INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Robust Convolutional Neural Network Model For Recognition of Fruits

Deepali M Bongulwar[1]*, **S N Talbar**[2]

1 Research Scholar, Department of Electronics & Communication Engineering, SSSUTMS, Sehore (M.P), India
2 Professor, Department of Electronics & Telecommunication Engineering, SGGSIE&T, SRTMU (M.S), Nanded, India

OPEN ACCESS

Received: 07.09.2021

Accepted: 16.12.2021

Published: 22.12.2021

Citation: Bongulwar DM, Talbar SN (2021) Robust Convolutional Neural Network Model For Recognition of Fruits. Indian Journal of Science and Technology 14(45): 3318-3334. https://doi.org/10.17485/IJST/v14i45.1493

* **Corresponding author**.

ddeepali2006@yahoo.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objectives: To develop a model for the automatic recognition of fruits utilizing deep learning techniques. **Methods:** We have designed a fruit classification and recognition Model using Convolutional Neural Networks (CNN). We have used excellent quality ImageNet dataset of fruit images for evaluation purpose. It contains 9,130 images of 11 different categories. The classification is challenging as the images comprise different fruits of the same color and shape, overlapped fruits, the background is not homogenous, and with different light effects etc. **Findings:** We have achieved a validation accuracy of 91.28 % and the testing accuracy of 100%. The same model is trained on the fruits-360 dataset with 92 categories of fruits with 47,526 images. The Model gives validation accuracy of 100% and testing accuracy of 100%. This study also compares the results obtained using transfer learning by training the EfficientNet-b0 architecture with the ImageNet and fruits-360 dataset. The validation accuracy is 96.77% and the testing accuracy is 100%. Again, the validation accuracy and testing accuracy for fruits-360 dataset is 100% and 99.9% respectively. **Applications:** Recognition of fruit is required in agricultural problems like robot harvesting and fruit counting and many more applications. Moreover, it can be used in the retail business, as a self-service system to recognize the fruits. It can be used in human-robot interactions. **Novelty:** The model once trained can achieve state-of-art accuracy for the recognition of any type of fruit with any background. It sometimes exceeds human-level performance. Hence, the Model is Robust enough to recognize the fruits.

Keywords: Deep learning; Convolutional Neural Networks; Fruit recognition; ImageNet; Transfer learning; Machine learning; fruits-360 dataset; EfficientNet-b0 model

1 Introduction

The mechanism of automatic classification of fruits gains importance due to labor cost increase and due to the lack of labor in rural production. The fruit recognition is required in agricultural applications to harvest fruits using robots. Therefore, the proposed system is a Deep Learning based fruit classification system that can be used in small farms. The area we are generally inspired by is making an independent robot that can carry out more intricate undertakings compared to a customary mechanical robot. It can be used in agricultural robotics to identify the fruits and help in packing bulk fruits for export.

Supermarkets place a high value on fruit quality assessment. A quality assurance check is an absolute necessity to guarantee that the stock is in acceptable condition and without any bad or blemished fruits. These types of quality inspections are typically performed by visually evaluating the fruits and rejecting the spoiled and faulty fruits from the excellent ones. In any case, the human examination isn't generally dependable because of stress, weariness, interruption and inability to concentrate. These inadequacies show the need of utilizing robotized image recognition techniques[1]. Since manual checkups by people can be mistake inclined and AI frameworks are getting increasingly productive, deep learning algorithms to characterize images are widely famous. Deep learning can be applied to develop a system to recognize the fruits using a large dataset of fruits. Fruit recognition is also used in the smart refrigerator. It can identify whether the fruits are fresh or not. It also identifies which fruit is left and which is to be purchased. It keeps the track of the stock and intimates accordingly[2]. Nowadays people are more conscious about health and healthy food. Supermarkets can have an autonomous fruit recognition system and help the consumer to select the fruit based on its nutrients. A supermarket can keep the track of sold and available fruits. The online system can use such a kind of automatic system[2]. Thus, fruit detection and recognition can be applied to a wide range of applications. We chose the task of identifying fruits as a starting point for the proposal for many reasons. One of the reasons is that fruits have certain categories that are difficult to separate. Consequently, we need to perceive how well AI would be able to finish the challenge of grouping them. Another factor is that fruits can always be found in stores, making them an excellent place to start.

Deep learning techniques have recently outperformed many algorithms in related fields, including image classification. The convolutional neural network is one of the most powerful deep learning approaches for image categorization. According to several studies, convolutional neural networks outperform regular image classification techniques like k-nearest neighbors, multilayer perception, and support vector machines. Convolutional Neural Networks (CNN) has become a hot research area in the field of image classification and recognition. When compared to traditional classification algorithms, CNN can extract features from the image. In CNN, the image can be directly used as input to the network, it does not require pre-processing and feature extraction[3]. In[4] the authors have developed a model to recognize the fruits from images. They have used the deep learning technique for the classification of fruits. They have used only 25 different categories of fruits from the fruits-360 dataset for evaluation purposes**.** Using the fruits-360 dataset, accuracy and loss were calculated for five different models using various combinations of neural network layers and epochs. They found the lowest loss with the presence of the dropout layer in the CNN algorithm as compared to without the dropout layer. The recognition rate has greatly improved with the rise in epochs throughout the experiment. The model obtained the best test accuracy of 100% best training accuracy of 99.79%. In[5] the researchers used a computer vision technique to identify papaya diseases. A total of 129 images were used in the creation of this piece. They used a feature collection to create a total of ten features. They separated the disease-affected area using the k-means clustering algorithm in order to extract the features. The classification challenge was completed by using three different classifiers. Support vector machines (SVMs) were discovered to have a 95.20% precision, beating out the other two classifiers. Although this work was expanded to include six more influential classifiers, SVM continues to outperform all others and k-NN, on the other hand, performs the poorest, with an accuracy of 71.11%. In[6] the authors have discovered that using data augmentation techniques, the accuracy rate of image-based classification of fruit can be improved. Their 13-layer CNN model achieved 95 percent accuracy. In[7], the researchers used a 9-layer convolutional neural network with data augmentation for the classification of fruits. It showed that with data augmentation they obtained an average precision of 96.34%.The architecture of a deep learning structure for rating bananas as healthy or defective is shown in the study[8]. The "ResNet-50" residual learning-based network was created to sort bananas. The system demonstrated a new deep learning technique called skip connections, which improved performance in a variety of tasks like classification and object detection. The system's design was applied in utilizing transfer learning, which resulted in a high accuracy of 99%. The researchers in[9] used the ImageNet dataset to pretrain eight CNN architectures, including VGG-16, VGG-19, Inception V3, ResNet-50, ResNet-101, ResNet-152, and AlexNet. According to the results, the best CNN for sorting Medjool dates was VGG-19 model which used the Adam optimizer and reported the best accuracy of 99.32%. In[10], a convolution neural network (CNN)-based fruit recognition model was proposed. Five West Africa indigenous fruits were selected. The study further presents transfer learning using VGG-16 and ResNet-50 models for result comparison. The baseline model produces the best accuracy of 96% as compared to VGG-16 and ResNet-50 as 91% and 84% respectively. The authors of[11] showed that fruit images of 22 categories from the ImageNet dataset and web crawling with complex backgrounds were classified with an accuracy of 95.60% using modified Darknet53. They replaced the batch normalization with the group normalization and also showed that accuracy does not depend on batchsize. A 13-layer deep convolutional neural network was used by the authors of[12] to create a fruit classification algorithm. They worked on 18 categories of fruits downloaded from Google and Baidu. The overall accuracy obtained was 94.94% over clean dataset but on complicated images like images with varied background the overall accuracy was 89.60%, over unfocused images was 91.03%, and over the occluded images was 92.55%.The authors of[13] proposed a CNN based technique for automatic fruit recognition and classification. They worked on a public dataset and self-made dataset with 7 categories. The fruit images in the public dataset were captured in a simple environment, whereas the fruit images in the self-created dataset were taken in a challenging way. On the public dataset, they had the greatest accuracy of classification of 99.8%. The categorization accuracy in the self-created dataset was 90.2%. Later on, they used appropriate data augmentation techniques to improve the classification accuracy of the self-created dataset to 98.9%.We have developed the CNN model to classify the images with complex backgrounds. The novelty of our research is that it can identify and recognize the fruit in half-cut, only seeds, and a small cut piece, kept in a dark place, enclosed in plastic bag or maybe under the water, just like the human being. It can also be used in humanoid robots. The authors of[14] proposed a realistic approach to the identification of fruits by using the two families of deep neural networks EfficientNet and MixNet that can accurately and quickly classify the fruits. The approach's output was tested on the fruits-360 dataset, considering a total of 48,905 images used for training and 16,421 images used for testing with 95 different fruit categories. The results demonstrate that for the most basic architectural designs, for example, EfficientNet-b0 and MixNet Small can predict with reasonable accuracy. The same EfficientNet-b0 is used in the paper to assess the efficiency of the proposed model. We proposed a Convolutional Neural Network (CNN) based fruit recognition technique to classify fruit images. Deep Learning algorithms include Convolutional Neural Networks (CNN). In Deep Learning, CNN is the most utilized Artificial Neural Network (ANN). CNN is utilized in many visual recognition systems, for example, video and image recognition, face recognition, manually written digit recognition, fruit recognition and so on. The fruit recognition using CNN have exceeded human-level performance. Thus, a fruit recognition system using CNN can be trained on the easily available dataset. Therefore, we have developed a fruit recognition system using CNN. We have used two different datasets. The input images of the ImageNet dataset were of size 224X224X3 pixels where 3 represents R, G, and B colors. To train our model we are using the ImageNet dataset, which contains 11 categories of fruits. We have chosen the fruits which we buy frequently from the market. The fruits are sometimes in a group with different light effects and the background is not homogeneous. Despite this, we have achieved the CNN Model with good accuracy. Thus, the Model can be used in human-robot interactions.

We have also trained the model on the publicly available fruits-360 dataset. We have achieved 100 percent accuracy. We have obtained state-of-the-art results. We have trained the same ImageNet dataset with transfer learning using the latest architecture EfficientNet-b0 available in MATLAB R2021a. The results obtained show how the model behaves with different datasets and proves the model to be best for recognizing the fruits. We have implemented the Model using MATLAB. To increase the efficiency and decrease the training time, efficient GPU like "NVIDIA GeForce RTX 2060 Super" was used in our experiment.

The entire paper is distributed into sections as follows. Section 1, explains the introduction and need for the classification of fruits and reviews the work related to the classification of fruits. In Section 2, we have presented the subtleties of different datasets like ImageNet and fruits-360 utilized in our system. Section 3 explains the methodology and structure of our proposed CNN model and the EfficientNet-b0 model. The findings of our experiments for both the datasets and the transfer learning are presented in Section 4. The discussion of the resulting output images is given in Section 5. Section 6 gives the conclusion.

2 Dataset

We have used two datasets to test the proposed model and compare its output for the recognition of fruits. The two datasets are the ImageNet dataset and the fruits-360 dataset.

Table 1. Dataset Properties

S.No.	Properties	ImageNet Dataset	Fruits-360 Dataset
1	Total Number of Images	9,130	47,526
2	Training set size	7,760	40,397
3	Validation Set Size	1,370	7,129
4	Test Set Size	1,870	15,938
5	Number of classes	11	92
6	Image Size	224X224X3 Pixels	100X100X3 Pixels
7	Background	Not Homogenous	Plain White
8	Each Image contains	Single/Group of Fruits	Single Fruit

2.1 ImageNet Dataset

All the images were downloaded from the ImageNet dataset. The Table 1 shows the details of ImageNet dataset. It contains 9,130 images of fruits of 11 different categories. The dataset consists of fruits like Banana, Blueberry, Custard Apple, Grapes, Guava, Jackfruit, Litchi, Mango, Pineapple, Pomegranate and Strawberry. The number of images of each category is 830 for training. All the images were initially resized to 224X224X3 pixels. Here, three represents the R, G, B colors. We have trained the Model using 85% of images and the remaining 15% were used for validation. The number of images in the training set was 7,760 and the remaining 1,370 images were used for validation. The total number of images in the test set is 1,870, with 170 in each category. Randomly selected 16 images are displayed in Figure 1. The images in the ImageNet dataset are more complex, as can be seen in the samples in Figure 1 since they often include fruits from different groups and the background is not uniform. The dataset was very challenging. The background was not homogeneous. Sometimes it was plain white, sometimes fruit was placed inside water, in a bowl, on a dish either single or in a group. They are rotten or fresh, cut into pieces, half cut, decorated in plates, alongwith seeds, close-up view, top view, side view, or a surface view showing its texture. Fruits were held at different angles, holding in hands, hanging on trees in their natural positions and so on. Fruits were enclosed in a plastic bag or covered by paper. Fruits were exposed to light, kept in dark, and some images were black and white.

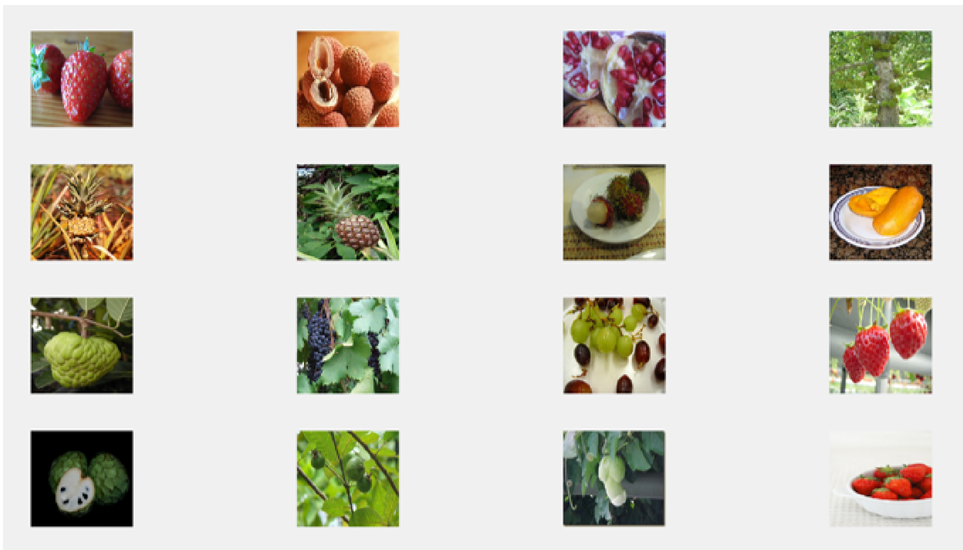

Fig 1. Sample Input Images of ImageNet dataset

2.2 Fruits-360 Dataset

It is a collection of images of fruits and vegetables. 2020.05.18.0 is the current version.

The dataset is available on GitHub [fruits_360_github] and Kaggle [fruits_360_kaggle].

We have used only fruits of 92 categories. The Table 1 above shows characteristics of the dataset.

The total number of images in the training and testing sets is 47,526 and 15,938 respectively. The size of each image is 100X100X3 pixels. Each image contains a single fruit. Different varieties of the same fruit (for example, apples) are classified into different groups. There are numerous apple varieties, each of which is treated as a different category. They are identified by digits like apple red 1, apple red 2 and so on. We have trained the model using 85% of images and the remaining 15% were used for validation. The number of images in the training set was 40,397 and the remaining 7,129 images were used for validation. The total number of images in the test dataset is 15,938. The images of the fruits-360 dataset in Figure 2 are simple and each image has a single fruit with white background. But, the fruits are difficult to recognize as they look similar, just like the apple red categories 1 and 2 belong to a different group, though they look similar. The data was randomly reflected in the horizontal direction, translated in the horizontal and vertical direction and also rotated. We have used data augmentation only for training images.

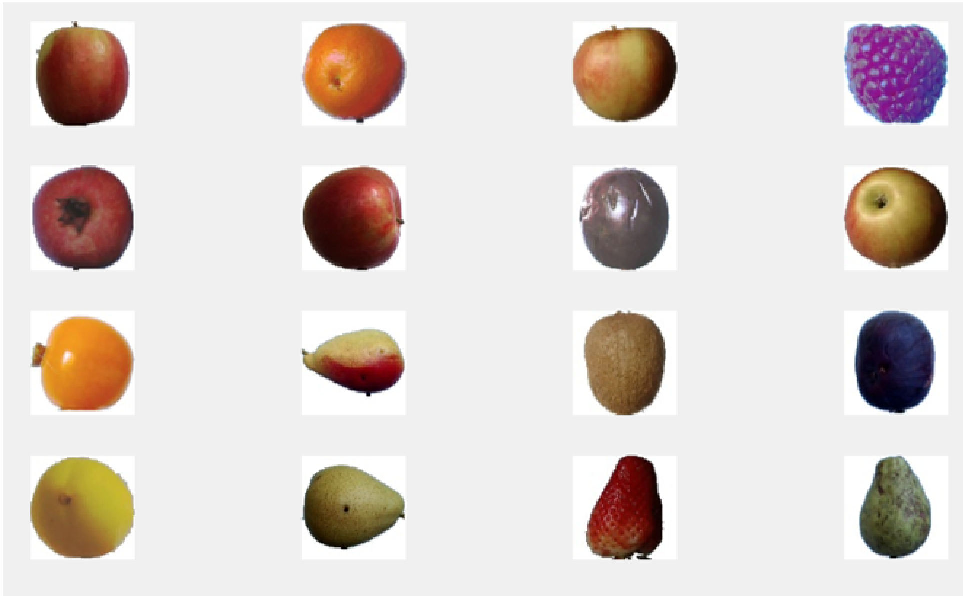

Fig 2. Sample Input Images of fruits-360 dataset

3 Methodology

3.1 The Proposed CNN Model and Its Structure

Convolutional neural networks are artificial neural networks with a deep feed-forward architecture. These are mostly used for 2D data i.e. images. The proposed CNN structure is divided into three stages: preprocessing, feature extraction, and classification. CNN's require a large set of the dataset. The images in the dataset are cropped and resized to 224X224X3 pixels for the ImageNet dataset and 100X100X3 for the fruits-360 dataset. Images are in the standard RGB format, where 3 refers to R, G, and B color. The next step is feature extraction. This is achieved by using convolution layers. The proposed CNN Model diagram consists of several blocks which in turn has sub-blocks. Each sub-block has a convolution layer followed by Batch Normalization Layer, ReLU and Max-Pooling layer. These blocks are directly attached to fully connected Layers and finally by a softmax layer to classify and recognize the input images. Figure 3 shows the block diagram of the proposed CNN Model. The convolutional layers use 7X7, 5X5 and 3X3 filters. They have the same depth as that of the input images. These slide across the images from left to right and top to bottom and produce the feature maps. A kernel, also known as a filter, is a small two-dimensional matrix that extracts particular features from the image. The output feature map is produced by convolving each kernel with the input feature map. At the end, all the results are added to produce a single output vector. The output feature vector f_{om} is generated by convolving the input feature vector f_{im} with the kernel k (x, y), i.e.

$$f_{om}(x, y) = (f_{im} * k)(x, y) = \sum_i \sum_j f_{im}(i, j).\, k(x - i, y - j) \quad (1)$$

Following them is the Batch Normalization layer. These layers normalize the preceding layers by deducting the mean of the batch and dividing by the batch standard deviation. The batchsize we have used is 64. It reduces overfitting as it has a regularization effect and increases the stability of the Model. They are followed by Rectified Linear Units (ReLU). These layers use

$$f(x) = \max(0, x) \quad (2)$$

as activation function.

These layers introduce a non-linearity function so that images are correctly classified but do not reduce the dimension of the images. After ReLU comes Max-Pooling layers. Pooling layers use kernels of size 2X2 and a stride of 2. These layers change the dimension of the images to one-fourth of their size and thus reduces the number of computations to be performed and so the time to process the model is also reduced. They are accompanied by fully connected layers (FC). Each neuron of the FC layer is connected to each neuron of the preceding layer. The FC layers are preceded by dropout layers. Again, these dropout layers regularize the model and decrease the number of computations and thus avoid overfitting of the model. Finally, the last layer uses a softmax activation function giving the output equal to the number of categories. It calculates the probabilities of each class. The highest probability class label is the output class of the given image.

The proposed CNN has 51 layers. There are a total of six blocks. All the blocks are connected through Max-Pooling layers. The Block_1 has three sub-block and each sub-block has a convolution layer and each convolution layer has 32 filters of size 7X7X3 and BiasLearnRate factor 5 as L1 regularization. Block_2 has two sub-blocks. Each sub-block has a convolution layer with 64 filters of size 5X5X3.Block_3 is the same as block_2 with two convolution layers and 128 filters of size 3X3X3.Block_4 has three sub-blocks. Each sub-block has a convolution layer with 256 filters of size 3X3X3.Block_5 has only one sub-block with one convolution layer and 384 filters of size 3X3X3. The last Block_6 has two sub-blocks with convolution layers having 512 filters of size 3X3X3. After this, there is a Max-Pooling layer which is then followed by four FC layers with 4096, 1024 and 120 neurons with 50%, 40%, 50% and 40% dropouts respectively. The final most layer is the softmax layer with 11 outputs for the ImageNet dataset and 92 for the fruits-360 dataset. The output with maximum probability is the output class.

We have used SGDM (Stochastic gradient descent with momentum) as an optimizer with piecewise learning. The optimizer is extremely important since it aids in lowering or raising the model's error function. The details of the hyperparameters are shown in Table 2. These parameters are obtained on the basis of trial and error. The model is trained for 110 epochs and the batchsize was 64. The size of the batchsize depends on the size of the dataset and the GPU. Here, 64 batchsize gives good results. The learning rate drop period was 50 and so the learning rate changed after every 50 epochs. The initial rate of learning was 0.01 as a smaller learning rate gives better results and the dropfactor was 0.2. L2 Regularization was added to reduce the overfitting of the model.

Table 2. Model Training Parameters

S.No.	Hyperparameters	Value
1	Optimizer	SGDM
2	Momentum	0.90 (default)
3	InitialLearnRate	0.01
4	Learning rate drop period	50
5	Learning rate drop factor	0.2
6	Batchsize	64
7	Epochs	110
8	L2 Regularization	0.0005

3.2 Transfer Learning

The performance of the model can be tested with a well-trained convolutional neural network model by using transfer learning. The process of taking a model that has been trained on a huge dataset and applying it to a specific dataset is known as transfer learning. We freeze the network's initial convolutional layers and train only the last few layers that render a prediction for recognition of an object with a CNN. The concept is that the convolutional layers obtain general, low-level features like edges, patterns, and gradients that are applicable across images, while the subsequent layers define unique objects within an image like eyes in face recognition example. We can illustrate the utility of transfer learning, by proving that the outcomes produced using the proposed model are superior to those obtained through transfer learning.

3.2.1 EfficientNet-b0 Architecture

We used a newly developed classifier, which has proven to be effective as well as efficient. We used EfficientNet[15] deep neural networks to create a framework that recognizes fruits. Over a million images from the ImageNet database were used to train the EfficientNet-b0 architecture. The images can be sorted into 1000 different categories, namely keyboards, mouse, pencils, and a variety of animals. As a consequence, the network has picked detailed feature representations for a variety of images. The EfficientNet family uses a compound scaling approach with fixed ratios in all three dimensions to maximize speed and precision. The results show that balancing network depth, width, and resolution can improve efficiency. Figure 4 shows the EfficientNet-b0 configuration, which is the smallest. There are a total 18 number of convolution layers, i.e., depth is 18, and every layer has a kernel of size 3X3 or 5X5. The size of the input image is 224X224X3 pixels, where 3 refers to R, G, and B color channels. The consecutive layers are reduced in resolution to minimize the size of the feature vector but increased in width to improve accuracy. If the second convolution layer has a width of 16 filters, then the consecutive convolution layer has a width of 24 filters. The last layer, the fully connected layer has the maximum number of filters as 1,280. Kernels of size 3X3, 5X5, or 7X7 are commonly used. The larger kernels may make the model better and increase the performance. Kernels that are large often aid in the capture of high-resolution patterns, whereas small kernels aid in the extraction of low-resolution patterns. As shown in

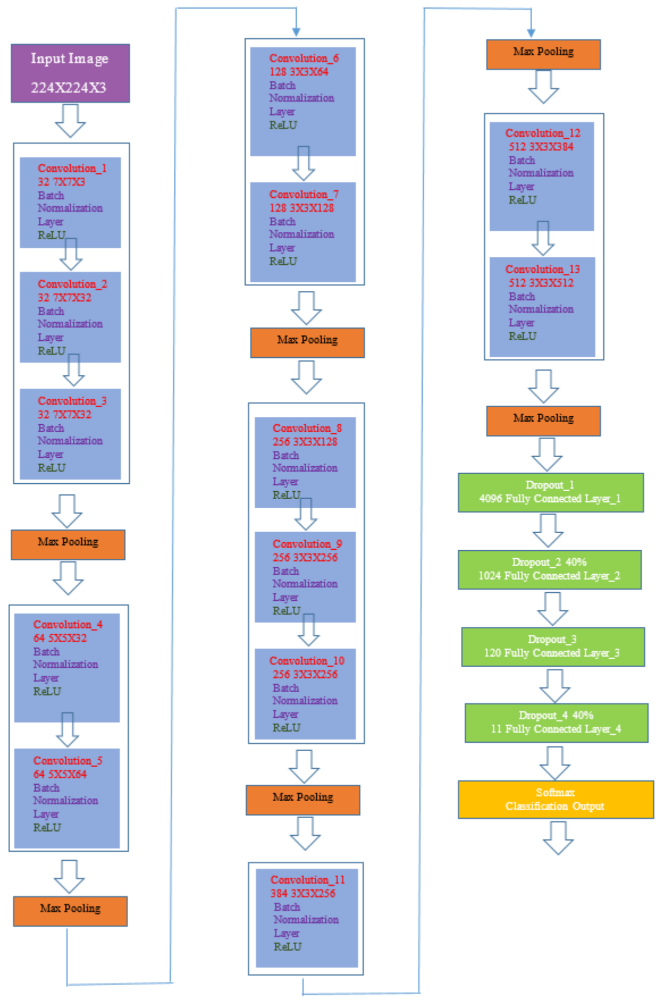

Fig 3. The Proposed CNN Model

Figure 4, we have trained and tested our dataset with this EfficientNet-b0 baseline model, which has 18 layers. The EfficientNet family have different configurations, like b1, b2 … & b7 and the base EfficientNet-b0. The smallest network is EfficientNet-b0.

3.3 Evaluation metrics

Accuracy: The ratio of accurately categorized fruits to the actual number of fruits in the validation dataset is used to calculate accuracy.

$$\text{Accuracy} = \frac{\sum_{p=1}^{n} TP_p}{\sum_{p=1}^{n}\sum_{q=1}^{n} N_{pq}} \qquad (3)$$

where Npq denotes the cumulative estimate of the classifier, and TP_p denotes true positive (the number of images in category p that are exactly labelled as category p).

3.4 Stochastic Gradient Descent with Momentum

It is among the most widely utilized deep learning optimization algorithms. The gradient descent algorithm changes weights and biases in deep neural networks to reduce the loss function by adding fewer steps in the direction of the loss function's negative gradient,

$$\theta\ell{+}1{=}\theta\ell{-}\alpha\nabla\mathrm{E}\,(\theta\ell) \qquad (4)$$

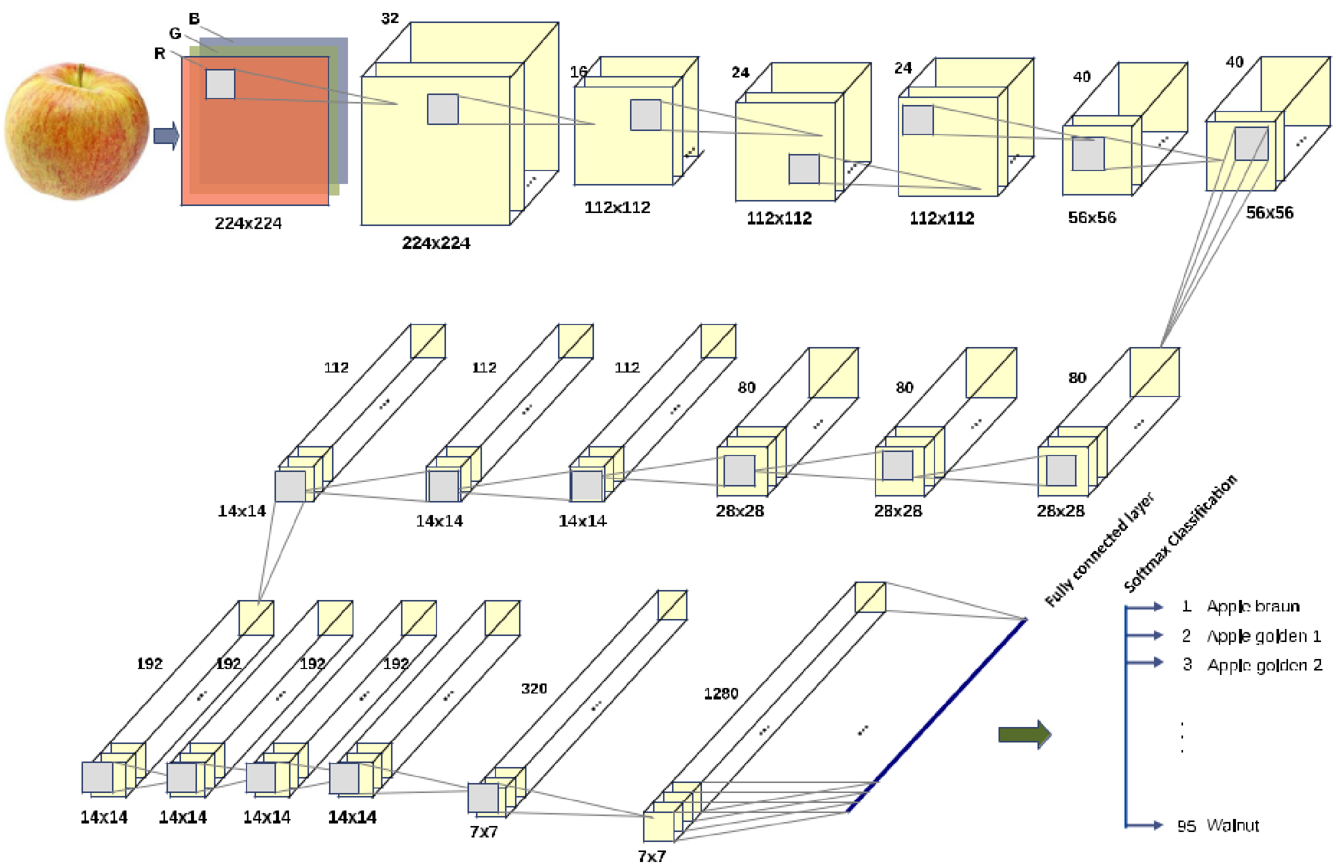

Fig 4. The EfficientNet-b0 Architecture [15]

here ℓ represents the number of iterations, α>0 indicates the rate of learning, θ is the parameter vector, and E (θ) is the loss function.

The entire training dataset is utilized to calculate the loss function's gradient, ∇E (θ). But, the stochastic gradient descent algorithm tests the gradient and updates a subset of the training set and not the entire dataset. This subset is named a minibatch. Iteration is the term used to describe the process of evaluating the gradient using the minibatch. With each iteration, the algorithm gets closer to minimizing the loss function.

An epoch can be defined as a complete pass of the training algorithm through the total number of samples using MiniBatches. The 'MiniBatchsize' and 'MaxEpochs' name-value pair arguments can be used to determine the size of MiniBatch and the total number of epochs, respectively. Here, the proposed model uses MiniBatchsize as 64 and MaxEpochs as 110. The stochastic gradient descent algorithm can move down the steepest descent path to the best solution. Another way to reduce the fluctuation is to use a momentum term.

$$\theta\ell+1=\theta\ell-\alpha\nabla E\,(\theta\ell)+\gamma\,(\theta\ell-\theta\ell-1) \qquad (5)$$

gives the increase in the stochastic gradient descent with momentum, here γ defines gradient step of the previous iteration to the present. The 'momentum' name-value pair argument may be used to define this value. Here, it is 0.90(default). Again, the SGDM is better optimizer as compared to another optimizer like 'Adam'. Although Adam converges faster, SGDM generalizes better and thus results in improved final performance.

In MATLAB, we define solver Name as 'sgdm' to utilize stochastic gradient descent with momentum for training a neural network. The starting value of the learning rate α can be defined as 'InitialLearnRate' name-value pair argument. The value of α is 0.01. We may even change the learning rates after some iterations. Here, we have used a piecewise learning schedule and so after a drop period of 50 epochs, the value of α changes by a drop factor of 0.2.

3.5 Hardware/Software

We have used Intel Core i7 Processor with 32 GB RAM and NVIDIA GeForce RTX 2060 Super GPU. We have implemented the model using MATLAB R2021a using Deep Learning Toolbox on Windows 10 Pro.

4 Experimental Results

We have tested and analyzed the proposed model with two different datasets for fruit classification.

4.1 Experiments of Proposed Model on ImageNet dataset

We have trained the CNN model for the classification and recognition of fruits. The ImageNet dataset was employed for the classification of 11 categories of fruits. We have trained the Model for 110 epochs with a minibatchSize of 64 and the learning rate was 0.01. We have optimized the dataset utilizing the stochastic gradient descent optimizer (SGDM) with a piecewise learning schedule. The data was shuffled after every epoch. The graphical representation of training and validation of the dataset for every epoch is shown in Figure 5. We have obtained 100% training accuracy and 91.28% validation accuracy. The total time taken to train the model is 731 minutes and 25 seconds. We have implemented the model using MATLAB software. We have used Neural Network Toolbox to find the accuracy of the model. Figure 6 shows the output labelled images. All are accurately classified. The validation accuracy for each category was also calculated using the confusion matrix. Figure 7 shows the confusion matrix giving the accuracy of classification and recognition of fruits. Here, the rows signify the predicted class i.e. output category of the fruit and the columns signify the true class of the fruit category. The diagonal cells represent the number of validation data samples to be correctly classified. Here, we have 124 samples of each class for validation. The cell in the base right shows the exact validation accuracy of the model. We have achieved 91.28% of the validation accuracy. Also, we tested the Model using test images. The test accuracy is 100%. The images can be tested within a fraction of a second.

4.2 Experiment of Proposed Model on fruits-360 dataset

Again, we tested the proposed model with the fruits-360 dataset. The image augmentation was applied similar to ImageNet dataset. We have obtained 100% training accuracy and 100% validation accuracy. The time required to train the model is 1001 minutes and 41 seconds. Figure 10and Figure 12show the training and validation accuracy behavior and the confusion matrix respectively. The confusion matrix shows that, not a single **s**ample is misclassified from the validation dataset. Figure 11 shows the output labelled images. All are accurately classified. The test accuracy is 100%.

4.3 Experiments of EfficientNet-b0 model on ImageNet dataset

Finally, we analyzed the EfficientNet-b0 model on the ImageNet dataset. In this case, we have fine-tuned the last layers of the model. We have used a small value of 3e-4 as an initial learning rate over 6 epochs and the minibatchsize of 10 images. We have optimized the dataset using a stochastic gradient descent optimizer with momentum (SGDM) with a constant learning schedule. The data was shuffled after every epoch. The time required to train the model is 55 minutes 01 second. The graphical representation of training and validation of the dataset for every epoch during fine-tuning the model is shown in Figure 15. The data was randomly reflected and translated in the horizontal as well as the vertical direction, and also scaled. We have used data augmentation only for training images. We have accomplished 100% training accuracy and 96.77% validation accuracy. The test accuracy is 100%. It gives results similar to the proposed Model on the same ImageNet dataset.

4.4 Experiments of EfficientNet-b0 model on fruits-360 dataset

Again, we tested the EfficientNet-b0 model with the fruits-360 dataset. The image augmentation was applied similar to ImageNet dataset. We have obtained 100% training accuracy and 100% validation accuracy. The time required to train the model is 350 minutes 40 seconds. Figures 20 and 22show the training and validation accuracy behavior and the confusion matrix respectively. Again, the confusion matrix shows that not a single sample is misclassified from the validation dataset. The test accuracy is 99.9%.

5 Results of Proposed Model

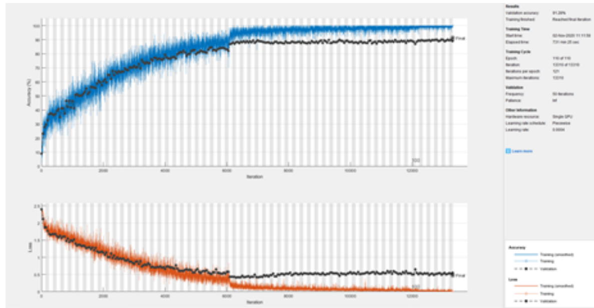

Fig 5. Training of the ImageNet dataset with the increase in Epochs

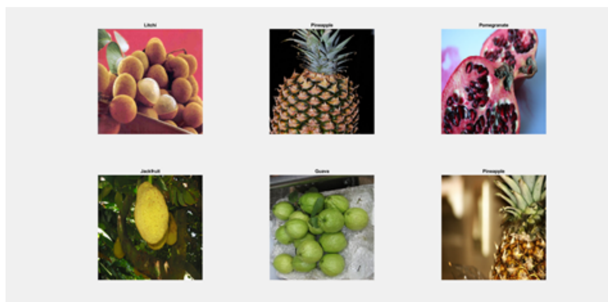

Fig 6. Sample output Images

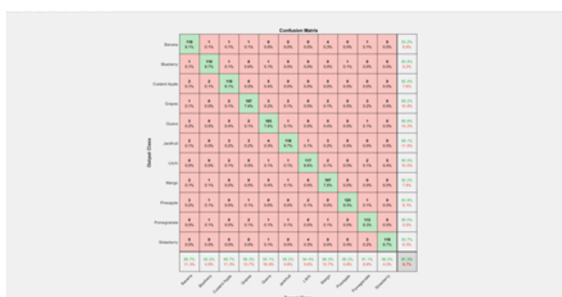

Fig 7. Confusion Matrix for ImageNet dataset

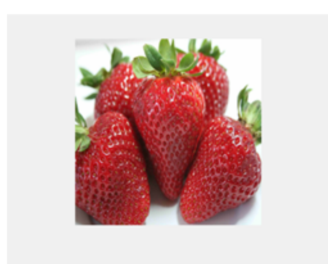

Fig 8. Test input image

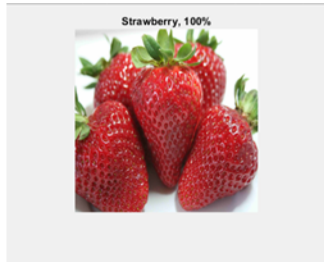

Fig 9. Result of testing

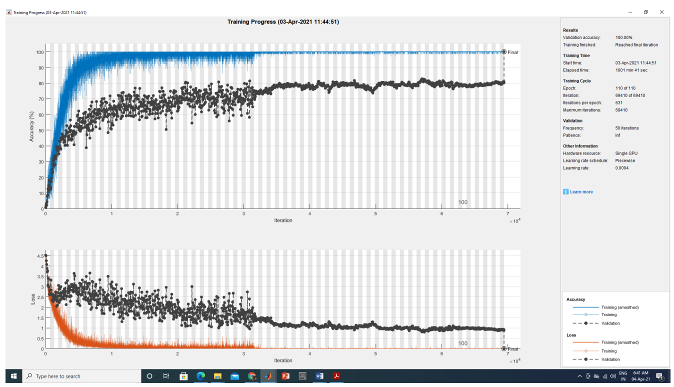

Fig 10. Training of the fruits-360 dataset with the increase in the Epochs (Crop the Taskbar)

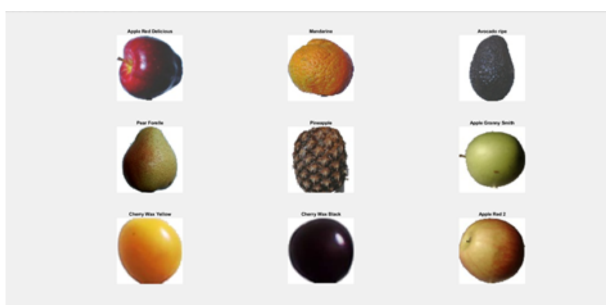

Fig 11. Sample output Images of Fruit-360 dataset

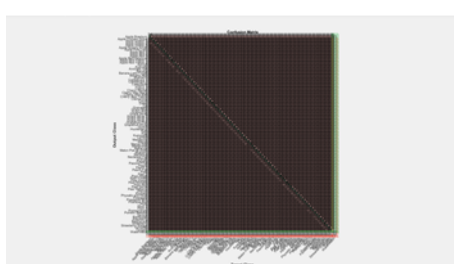

Fig 12. Confusion Matrix for fruit-360 dataset

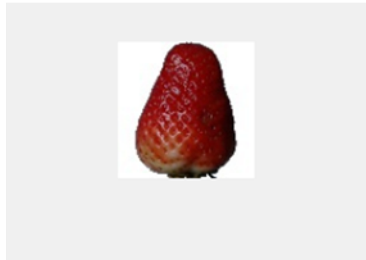

Fig 13. Test input

Fig 14. Result of testing

5.1 Results of EfficientNet-b0 Model for ImageNet dataset

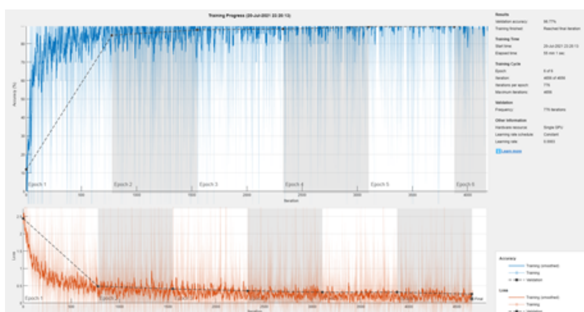

Fig 15. Training of the ImageNet dataset with increase in the Epochs

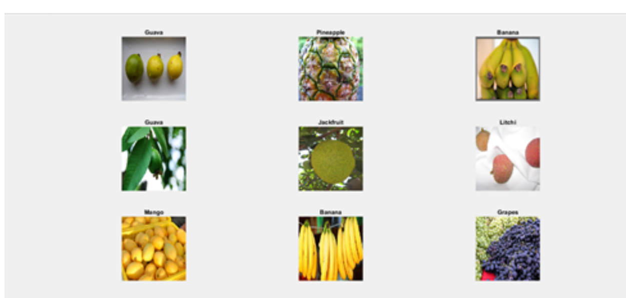

Fig 16. Sample output Images of ImageNet dataset

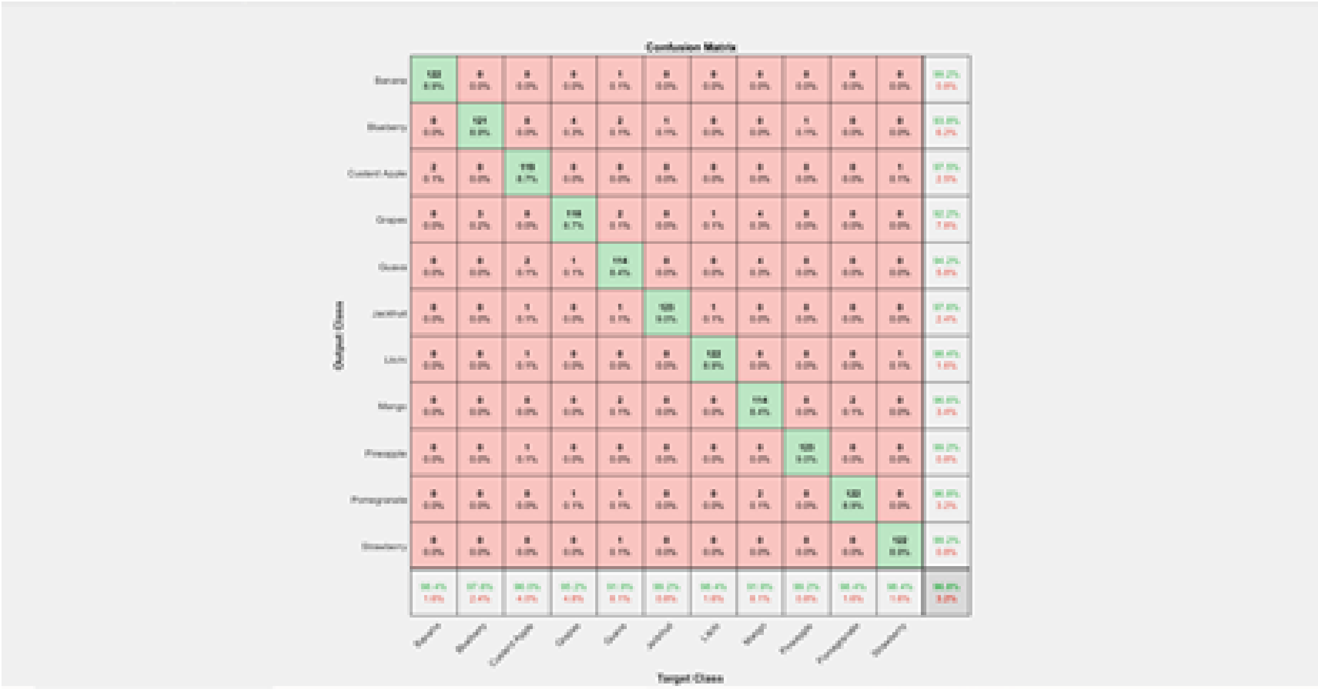

Fig 17. ConfusionMatrix for ImageNet dataset

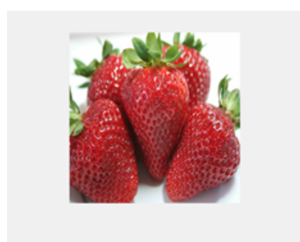

Fig 18. Test input

Fig 19. Result of testing

5.2 Results of EfficientNet-b0 Model for Fruit-360 dataset

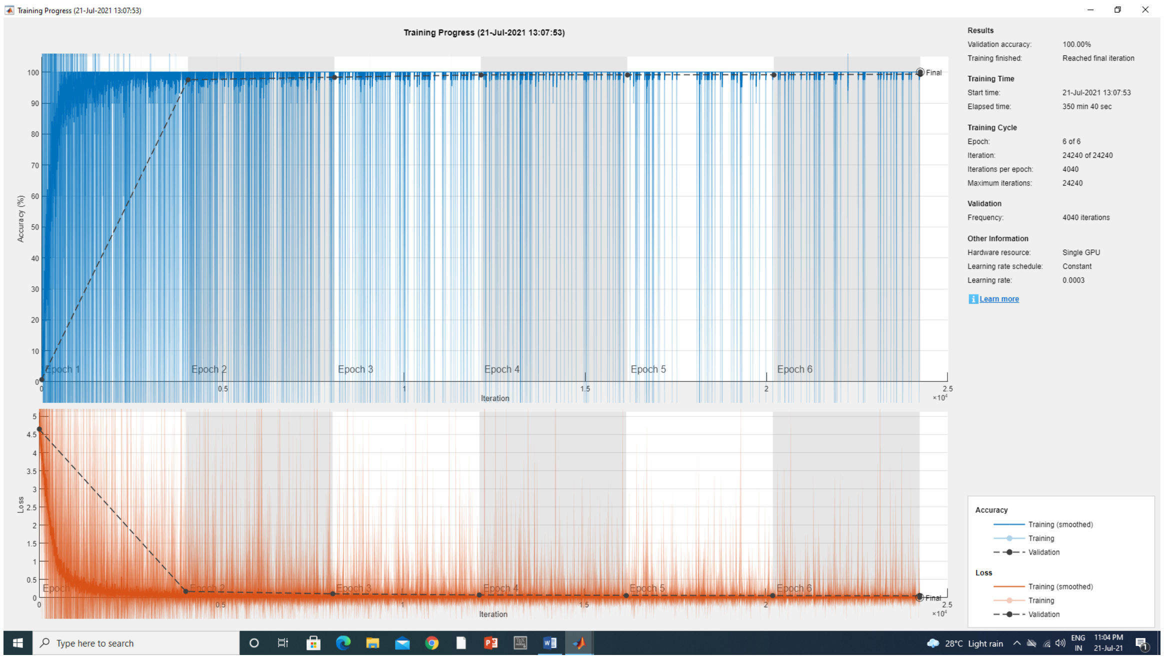

Fig 20. Training of the fruits-360 dataset with increase in the Epochs (Crop the Taskbar)

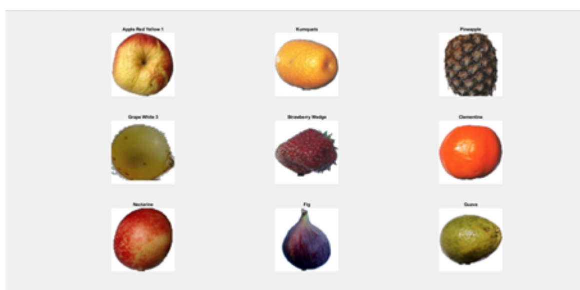

Fig 21. Sample output Images of fruits-360 dataset

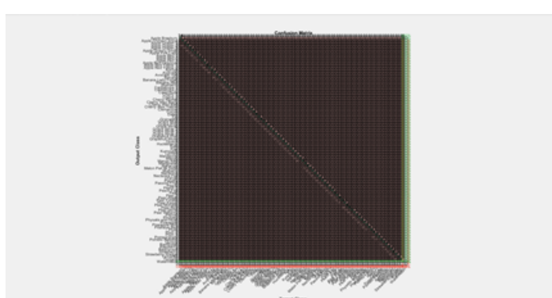

Fig 22. Confusion Matrix for fruits-360 Table 3 dataset

Table 3. Analysis of the Proposed Model Based on Previous Research

S. No.	Research Study	CNN Architecture/ Classifiers	Dataset	Number Of Images	Number Of Categories	Image Size	Classification Accuracy	Test Accuracy
1	The Current Study	The Proposed CNN Model	ImageNet	9,130	11	224X224X3	91.28%	100%
2		The Proposed CNN Model	fruits-360	47,526	92	100X100X3	100%	100%
3		EfficientNet-b0 Neural Network Model Using Transfer Learning	ImageNet	9,130	11	224X224X3	96.77%	100%
4		EfficientNet-b0 Neural Network Model Using Transfer Learning	fruits-360	47,526	92	100X100X3	100%	99.90%
5	Classification of 7 fruits using CNN [16]	CNN	ImageNet	10,578	7	256X256X3	91.66%	N/A
6	Fruit Classification using EfficientNet and MixNet [12]	EfficientNet-b0 Neural Network Model Using Transfer Learning	fruits-360	48,905	95	100X100X3	N/A	99.98%
7	Fruit Recognition Using EfficientNet-b0 algorithm [17]	EfficientNet-b0 Neural Network Model Using Transfer Learning	fruits-360	17624	25	100X100X3	95.67%	98%
8	Fruit Recognition using Deep Learning [18]	CNN	fruits-360	90,380	131	100X100X3	100%	98.66%
9	Fruit Image Classification using Pure CNN [3]	CNN	fruits-360	55,244	81	100X100X3	98.88%	97.87%
10	Fruit Classification using Six Layer CNN [19]	CNN	http://images.Google.com http://images.baidu.com	1800	09	256X256X3	91.44%	N/A

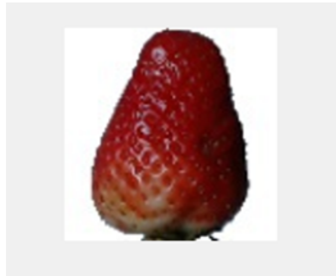

Fig 23. Test input

Fig 24. Result of testing

6 Discussion

The images of fruits-360 are clean with plain white background and so the validation and testing accuracy for both the proposed model and transfer learning is very high and promising. On the other hand, ImageNet dataset images are with complicated backgrounds and so could not reach 100% validation accuracy but could find testing accuracy as 100%. The effect of preprocessing and optimization was analyzed in both datasets. It was found that the factors that influence to have a better performance are initial learning rate, optimizer, and number of epochs, L2 regularization and the number of hidden layers. We have used the same optimizer for the proposed model and transfer learning but other parameters were different. We have chosen them based on the trial and error method and no specific algorithm was used to choose the initial learning rate and so on.

The results summarized in Table 3 for the proposed model show that all the images are accurately classified. The increase in number of epochs more than 110 does not significantly improve the accuracy. The accuracy obtained for the fruits-360 dataset is very good than that of the ImageNet dataset. The accuracy increases if the background of the images are plain or white and the dataset is simple with no complications. The fruits in the fruit-360 dataset have 92 categories with plain or white background. They are rotated. But still, they are correctly recognized. The time to train the model depends upon the hardware and the size of the dataset. Our model achieves the best accuracy of 91.28% and 100% for ImageNet and the fruits-360 model respectively. The accuracy with EfficientNet-b0 is 96.77% and 100% for ImageNet and fruits-360 dataset respectively. This shows that our model is well designed to recognize the fruits.

Once, the model is trained it can be used for testing. The testing accuracy is also 100%. It recognizes the test images within a fraction of a second and hence the model can be used for practical implementation.

Similarly, Table 3 shows a comparative analysis of our work with previous research carried out in fruit recognition using the CNN model. The authors of[18] and[3] used the same dataset fruits-360, but they could not achieve 100% validation and testing accuracy. Again, the same EfficientNet-b0 algorithm was used in[14] and[17] for the classification of fruits but could not achieve 100% validation accuracy and 99.90% testing accuracy as the proposed model has achieved. The classification accuracy of ImageNet dataset in[16] using CNN for 7 categories is 91.66%, while we achieved 91.28% for 11 categories with 100% testing accuracy. The comparison table shows that the proposed model achieves considerably more accuracy as compared to the previous research using CNN model. Even the EfficientNet-b0 results are more accurate as compared to the prior results. Therefore, the results of our model are excellent and can be applied to real world applications.

7 Conclusion

We have developed a new CNN model for the classification and recognition of fruits using MATLAB software. We have achieved very good accuracy on the used ImageNet and fruits-360 dataset. The results showed that the efficiency of the proposed model is similar to the transfer learning using the latest EfficientNet-b0 model. The testing accuracy is also very good with all the datasets and for both the architectures and so can be applied to the agricultural area. We have achieved state-of-the-art results and hence, we say the proposed model is a robust one.

Though, the proposed model gives 100% accuracy the number of computations is more and so needs more time to train the model.

As a future aspect, we can add more categories of fruits to the model. We can increase the training and validation sets to incorporate more things. We can test the model by changing the activation function and the optimization methods.

References

1) Ashraf S, Kadery I, Chowdhury MAA, Mahbub TZ, Rahman RM. Fruit Image Classification Using Convolutional Neural Networks. *International Journal of Software Innovation*. 2019;7(4):51–70. doi:10.4018/ijsi.2019100103.
2) Rafflesi DK, Rameswar. Multi class fruit classification using efficient object detection and recognition techniques. *Graphics and Signal Processing*. 2019;11:1–1. doi:10.5815/ijigsp.2019.08.01.
3) Asia K, Mohsin S, Jinhyuck P. Pure-CNN: A framework for fruit images classification. *International Conference on Computational Science and Computational Intelligence*. 2018;p. 404–408. doi:10.1109/CSCI46756.2018.00082.
4) Shadman S, Zahidun A, Siddique M, Bakr A. Implementation of fruits recognition classifier using convolutional neural network algorithm for observation of accuracies for various hidden layers. *ArXiv e-Journal*. 2019. Available from: https://arxiv.org/ftp/arxiv/papers/1904/1904.00783.pdf.
5) Habib MT, Majumder A, Jakaria AZM, Akter M, Uddin MS, Ahmed F. Machine vision based papaya disease recognition. *Journal of King Saud University - Computer and Information Sciences*. 2018;32(3):300–309. doi:10.1016/j.jksuci.2018.06.006.
6) Zhang YD, Dong Z, Chen X, Jia W, Du S, Muhammad K, et al. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*. 2017;78(3):3613–3632. doi:10.1007/s11042-017-5243-3.
7) Rodriguez M, Pastor F, Ugarte W, 28th Conference of Open Innovations Association (FRUCT). Classification of Fruit Ripeness Grades using a Convolutional Neural Network and Data Augmentation. 2021;p. 374–380. Available from: https://doi.org/10.23919/FRUCT50888.2021.9347597. doi:10.23919/FRUCT50888.2021.9347597.
8) Abdulkader H, Ma'aitah S, K M, Rahib A, Uzelaltinbulat H, Sonyel S, et al. Deep Learning Based on Residual Networks for Automatic Sorting of Bananas. *Journal of Food Quality*. 2021. doi:10.1155/2021/5516368.
9) Pérez-Pérez BD, Vázquez G, Salomón-Torres JP, R. Evaluation of convolutional neural networks' hyperparameters with transfer learning to determine sorting of ripe Medjool dates. *Agriculture*. 2021;11(2):115–115. doi:10.3390/agriculture11020115.
10) A MU, Aina S, Okegbile O, A RL, Oluwaranti A. *Recognition of West African Indigenous Fruits using a Convolutional Neural Network Model*;41:10–14. Available from: https://ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/1885.
11) Wang H, Zhang F, Wang L. Fruit classification model based on improved Darknet53 convolutional neural network. *In2020 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. 2020;p. 881–884.
12) Zhang YD, Dong Z, Chen X, Jia W, Du S, Muhammad K, et al. Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*. 2017;78(3):3613–3632. doi:10.1007/s11042-017-5243-3.
13) Wu L, Zhang H, Chen R, Yi J. Fruit Classification using Convolutional Neural Network via Adjust Parameter and Data Enhancement. *In2020 12th International Conference on Advanced Computational Intelligence (ICACI)*. 2020;p. 294–301.
14) Linh TD, Phuong N, Claudio TD, Davide D. Automated fruit recognition using EfficientNet and MixNet. *Computers and Electronics in Agriculture*. 2020;171:105326–105326. Available from: https://doi.org/10.1016/j.compag.2020.105326. doi:10.1016/j.compag.2020.105326.
15) Tan M, Le QV. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2019. Available from: https://arxiv.org/abs/1905.11946.
16) Yang M, In S, Cho. Fruit Classification using Convolutional Neural Network (CNN) Precision Agriculture Science and Technology. 2021. Available from: ComputerEngineering.
17) Thi CD, Van Tai, D. A fruits recognition system based on a modern deep learning technique. *Journal of physics: conference series*. 2019;1327:12050–12050. doi:10.1088/1742-6596/1327/1/012050.
18) Horea M, Mihai O. Fruit recognition from images using deep learning. *Acta Univ Sapientiae, Informatica*. 2018;10(1):1–38. Available from: https://arxiv.org/abs/1712.00580v10.
19) Lu S, Lu Z, Aok S, Graham L. Fruit Classification Based on Six Layer Convolutional Neural Network. *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. 2018;p. 1–5. doi:10.1109/icdsp.2018.8631562.