International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 3, June 2024, pp. 3084~3093
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i3.pp3084-3093

Coffee bean graded based on deep net models

Bipin Nair Balakrishnan Jayakumari[1], Abrav Nanda Koovamoola Mambilamthoda[1], Shalwin Ambalamoottil Stephen[1], Pranav Venkitesan[2], Venkatesh Raghavendra[1]
[1]Department of Computer Science, School of Computing, Amrita Vishwa Vidyapeetham, Mysuru, India
[2]Zenloop GMBH, Berlin, Germany

Article Info

Article history:

Received Jun 11, 2023
Revised Feb 12, 2024
Accepted Mar 3, 2024

Keywords:

Coffee bean
Deep learning
EfficientNet-B0
Grading
ResNet34
VGG-16

ABSTRACT

Coffee is a widely consumed beverage, and sorting coffee beans is a critical process that ensures high-quality graded coffee products. Coffee beans were graded into nine grades in robusta types. To automate the grading process, a deep learning-based approach was developed using a large dataset of high-resolution images and data augmentation techniques. In contrast to previous studies focusing on robusta type graded into six coffee bean grads, our research extends this framework by employing robusta type into nine grades with an outperformed accuracy. The proposed work uses four deep learning models, namely residual network 34(ResNet34), inception version 3 (Inception v3), efficient network Bayesian optimization (EfficientNet-B0), and visual geometry group-16 (VGG-16), where trained and evaluated for coffee bean classification into nine grades. The EfficientNet-B0 model exhibited outperformed accuracy, achieving 100% in distinguishing good and bad coffee beans, even in challenging lighting and background conditions.

Corresponding Author:

Bipin Nair Balakrishnan Jayakumari
Department of Computer Science, School of Computing, AmritaVishwa Vidyapeetham
Mysuru, 570026, India
Email: bipin.bj.nair@gmail.com

1. INTRODUCTION

Coffee is a globally popular beverage and an important economic crop cultivated in over 70 countries. The rise of specialty coffee has led to a greater emphasis on the quality of coffee, which is heavily influenced by the quality of green coffee beans. Manual sorting of coffee beans is labor-intensive and time-consuming, making it impractical for large-scale coffee production. Therefore, there is a need for automated methods to classify coffee beans based on their quality.

In recent years, image recognition technology has been applied in various fields, such as smart cities, medical care, and agriculture. Unal *et al.* [1] proposed various deep learning models to classify hazelnut kernels with a good accuracy of 99.28%. Musa *et al.* [2] experimented with a hybrid model of the GoogLeNet slap swarm on the dry beans dataset and classified it into 14 classes with a good accuracy of 91.43%. Kanae *et al.* [3] investigated a deep convolutional neural network (CNN) model on the peach fruit dataset and classified into seven traits with adequate accuracy. Mukesh *et al.* [4] used an ensemble classifier applied to the mango dataset for grading with adequate accuracy. Prabhu *et al.* [5] proposed a support vector regression with radial basis function kernel model on the Alphonso mango dataset for estimating weight with good accuracy. Seema *et al.* [6] in their study, a deep convolutional model was utilized to classify wheat seeds into four distinct categories using a dataset obtained from Kaggle. The model achieved a high accuracy rate of 98%. Qiang *et al.* [7] investigated a deep model you only look once X (YOLOX) applied on the shiitake mushrooms dataset for grading with the best accuracy. Knott *et al.* [8] experimented with a vision transformer on a fruit dataset and assessed the quality with 90% accuracy. Yuhang *et al*. [9] experimented with various deep-learning models on fruit datasets graded with average accuracy. Mukhriddin *et al*. [10] experimented on a deep model YOLO version 4 on fruit and vegetables classified with adequate accuracy. Bipin *et al*. [11] proposed a deep model Densenet121 model on 363 images of coffee beans to recognize the different grades based on their features and patterns with good accuracy 81.89%. Ahmad *et al*. [12] experimented with a CNN model on 3,360 ginger powder images graded as seven types with 99.70% accuracy. Limiao *et al*. [13] discussed a cross-domain denoising network on carrots graded with a high accuracy of 95.1%. Fu *et al*. [14] proposed a deep model GoogLeNet on fruits and vegetables to classify with a good accuracy of 98.82%. Basavaraj *et al*. [15] discussed a deep model on 30,000 paddy crops to classify with an accuracy of 92.89%. Meenaxi *et al*. [16] proposed deep learning models on okra lady's fingers classified into four categories with the best accuracy of 99%. Awad *et al*. [17] proposed a convolutional neural network on 2,000 lemons, classifying them into three classes with the best accuracy of 99.8%. Anuja *et al*. [18] experimented with a fruit weight measuring network on 328 pomegranates for weight estimation with adequate accuracy. Varsha *et al*. [19] proposed a deep model on the mango dataset, classifying it into two classes with a high accuracy of 93.33. Deborah *et al*. [20] experimented with an adaptive neural-fuzzy inference on a 200 cacao bean dataset to classify the beans into six quality levels with an accuracy of 99.71%. Chen *et al*. [21] proposed a deep model for the classification of tea categories based on tea images. Li *et al*. [22] investigated a deep model for identifying tomato diseases using hyperspectral images with high accuracy. Nen-Fu *et al*. [23] proposed a CNN model on coffee bean selection with a good accuracy of 93%. Wallelign *et al*. [24] created a CNN model using transfer learning to classify 2,109 coffee bean quality grades with 89.1% accuracy. Faridah *et al*. [25] proposes a hybrid model to grade robusta coffee beans into six grades based on various image parameters. The accuracy of the system is as follows: 100%, 80%, 60%, 40%, 100%, 40% and 100%, respectively.

2. METHOD

Figure 1 discusses the flow of the deep net models. In the first stage, the coffee bean image dataset is passed to four deep learning models, ResNet-34, VGG-16, Inception v3, and EfficientNet-B0, for predicting the nine grades. The model performs its own prediction on the coffee bean images and assigns them to one of the nine grades. The predictions generated by each model are evaluated based on their accuracy in classifying the coffee beans into the correct grades. The EfficientNet-B0 model particularly stands out in terms of accuracy, as it achieves 100% accuracy while predicting the coffee beans into nine grades. The EfficientNet-B0 model is fine-tuned as the proposed model to demonstrate robustness and reliability in predictions.

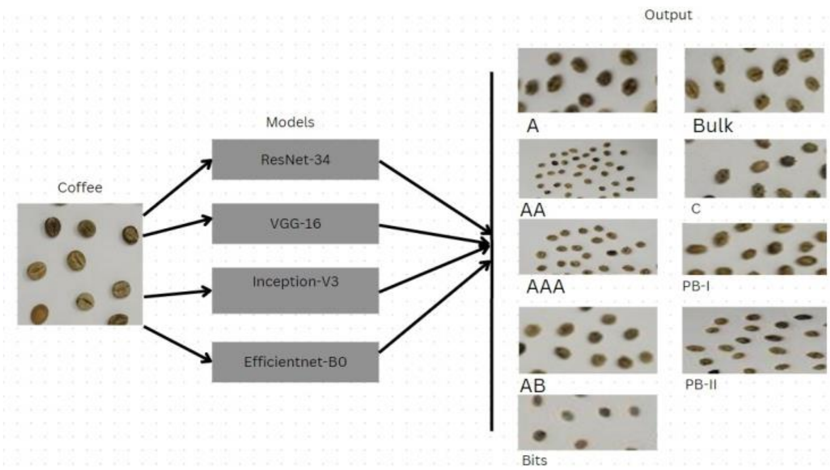


Figure 1. Flow diagram

2.1. Model 1-VGG-16

In Figure 2, the model comprises a total of 23 layers; 2 convolutional layers (convo) are present in block 1. 64 filters of 3×3 filter size (F-size) with an image input size (IMG) of (224,224,3) stride-1(S) function of activation (AF) MaxPooling layer (Max-PL) is inserted after each convolutional layer with a pool size of 2×2 and a stride of 2. Having two convolutional layers, block 2's picture input size is (112, 112, 64).

128 filters, each measuring 3 by 3 filters. Having three convolutional layers, block 3's picture input size is (112, 112, 64). 3×3 filter size filters totalling 256. Following the addition of each convolutional layer with a max-pooling layer and a pool size of 2×2, stride-1's activation function is ReLU, and stride-2 follows. The image input size for block 4's three convolutional layers is (28, 28, 256). 512 filters, each with a filter size of 3×3. The activation function for stride-1 is After every convolutional layer, a layer of max-pooling is added with a pool size of 2×2 and a stride of 2. The picture input size for block 5's three convolutional layers is (14, 14, 512). 512 filters, each with a filter size of 3×3. The activation function for stride-1 is After every convolutional layer, a layer of max-pooling is added with a pool size of 2×2 and a stride of 2. The neural network model includes a final block consisting of three fully connected layers, each having 256 units and using the rectified linear unit (ReLU) activation function.

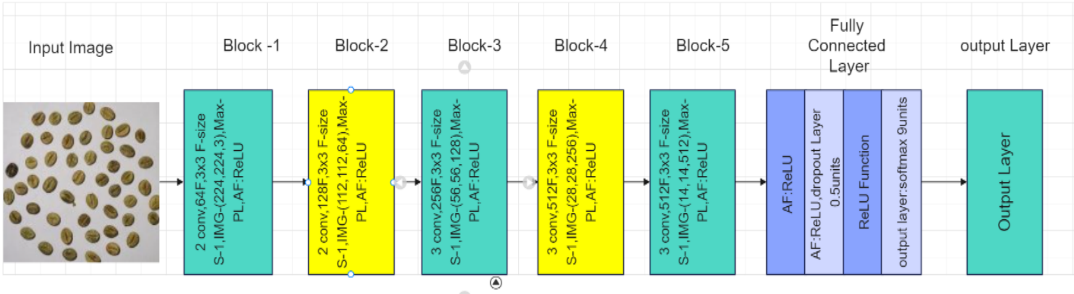
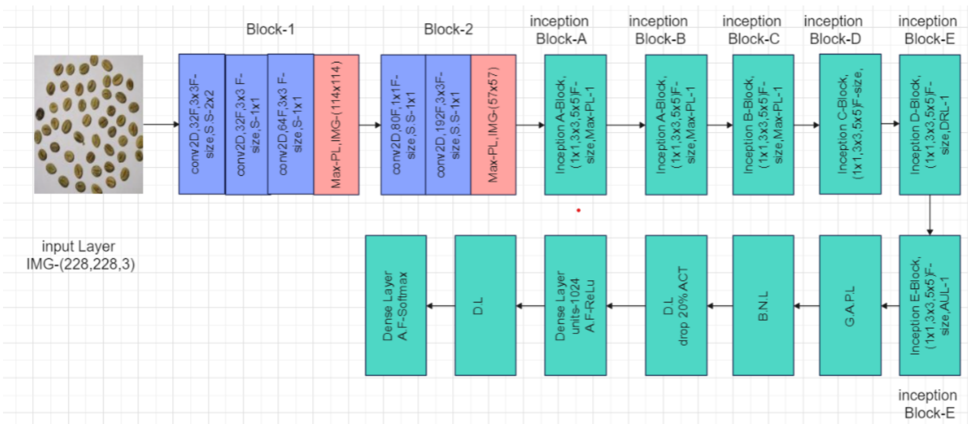


Figure 2. VGG-16 architecture

2.2. Model 2-inception v3

In Figure 3, the Conv2D layer with 32 filters (F) of size (3,3) and a stride (S) of (2,2), which cuts the image size in half, is the first layer in the pre-trained Inception v3 block. A Conv2D layer with 32 filters of size (3,3) and a stride of (1,1) also makes up the second layer. The third layer is a Conv2D layer with 64 filters of size (3,3) and a stride of (1,1). A MaxPooling2D layer (Max-PL) that cuts the image's size in half is added next. While the fifth layer has 192 filters of size (3,3) and a stride of (1,1), the fourth layer is another Conv2D layer with 80 filters of size (1,1) and a stride of (1,1). The following MaxPooling2D layer cuts the image in half. The Inception-A, Inception-B, Inception-C, Inception-D, and Inception-E blocks contain a MaxPooling2D layer to aid the model in learning features of varied sizes and complexity. A layer called batch normalization layer (B.N.L.) normalizes the layer's activations before it to make training easier. The GlobalAveragePooling2D layer (G.A.P.L.) lowers the spatial dimensions of the features to a single dimension. To avoid overfitting, a dropout layer (D.L) randomly removes 20% of the activations, followed by a dense layer (A.F) with 1024 units and ReLU activation. Following that is another dropout layer that again eliminates 20% of the activations.

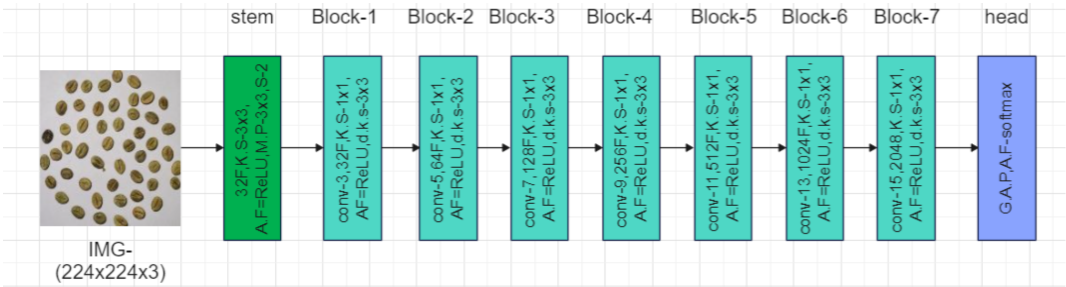


A.F=Activation Function, S.S=Stride size, F=Filter, S=Stride, Max-PL=Max Plooing layer, G.A.P.L=Global Average Pooling Layer, B.N.L=Batch Normalization Layer, D.L=Dropout Layer

Figure 3. Inception v3

2.3. Model 3-EfficientNet-B0

In Figure 4, the structure of EfficientNet-B0 consists of a stem, seven blocks, and a head. The stem, which is the model's input layer, is composed of batch normalization, a ReLU activation function (A.F), and a convolutional layer with 32 filters and a 3×3 kernel size (k.s). The next step is to apply a MaxPooling layer with a pool size of 3×3 and a stride (S) of 2. A ReLU A.F, a depthwise convolutional layer with a k.s of 3×3, batch normalization, and a pointwise convolutional layer with 32 filters and a k.s of 1x1 make up block 1's three layers. Block 2 contains five layers and is composed of a ReLU A.F, a depthwise convolutional layer with 64 filters and a k.s of 1x1, a batch normalization layer with a k.s of 3×3, and a pointwise convolutional layer with a k.s of 3×3. Block 3 comprises seven layers and is made up of a pointwise convolutional layer with 128 filters, and a k.s. of 1×1, a depthwise convolutional layer with a 3×3 k.s., batch normalization, and a ReLU A.F. Block 4 includes nine layers and is made up of a pointwise convolutional layer with 256 filters and a k.s of 1×1, a depthwise convolutional layer with a k.s of 3×3, batch normalization, a ReLU A.F, and a depthwise convolutional layer with a k.s of 3×3. Block 5 comprises 11 layers and is made up of a pointwise convolutional layer with 512 filters and a k.s. of 1×1, a depthwise convolutional layer with a 3×3 k.s., batch normalization, a ReLU A.F, and a layer with a ReLU A.F. Block 6 includes 13 layers and is made up of a pointwise convolutional layer with 1024 filters and a k.s of 1×1, a depthwise convolutional layer with a k.s of 3×3, batch normalization, a ReLU A.F, and a layer with a ReLU A.F. Block 7 includes 15 layers and is made up of a pointwise convolutional layer with 2048 filters and a k.s of 1×1, a depthwise convolutional layer with a 3×3 k.s, batch normalization, a ReLU A.F, and a layer with a ReLU A.F.

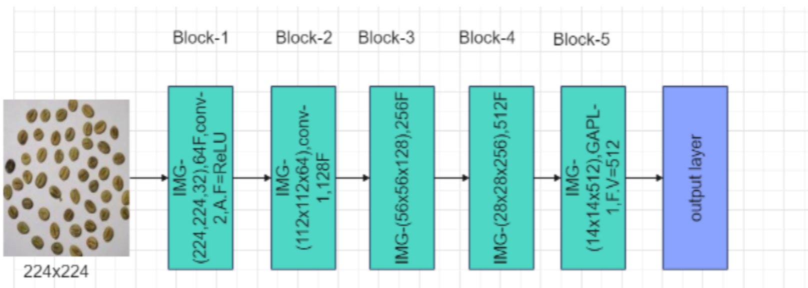


IMG-Image, A.F=Activation Function, d.k.s= depth-wise kernel size, F=Filter, K.S=Kernel Size, S=Stride

Figure 4. EfficientNet-B0

2.4. Model 4 - ResNet-34

In Figure 5, the model architecture comprises 5 blocks, each of which has a batch normalization function, a ReLU activation function, and several convolutional layers. The input image is subjected to 64 filters in the first block, resulting in an output with the dimensions 112×112×64. There are two convolutional layers with 64 filters in each sub-block of the first block, followed by batch normalization and ReLU activation function. The second, third, and fourth blocks all share the same structure, except the second block has 128 filters, the third has 256 filters, and the fourth block has 512 filters. These blocks' respective output sizes are 56×56×128, 28×28×256, and 14×14×512. The feature maps are condensed in the final block using a global average pooling layer to create a single vector of length 512, which is then input into a fully connected layer with 9 neurons to generate the model's output.



IMG-Image size, A.F=Activation Function, F=Filter, G.A.P.L=Gobal Average Pooling Layer, F.V=Feature Vector

Figure 5. ResNet-34

2.6. Proposed algorithm

The algorithm is a deep-learning model for image classification using four different pre-trained CNN architectures: ResNet34, Inception v3, EfficientNet, and VGG-16. The input to the model is an image, denoted as IImg. Each pre-trained CNN architecture extracts a feature from the input image, which is then fed into a set of fully connected layers to classify the image into one of the pre-defined classes. The output of each pre-trained CNN architecture is denoted as Res34Img, InceImg, EffImg, and Vgg16Img, respectively.

$$
\begin{gathered}
IIMG \;\rightarrow\; (Resnet\ 34, InceptionV3,\ Efficientnet - BO, VGG - 16) \\
\{ \\
Res34Img \;\rightarrow\; (con - 3, GAP - 1, B.N - 1, FC - 1) \\
InceImg \;\rightarrow\; (con - 4, IM - 7, MPL - 6, BN - 1, DL - 2, Dr - 2, Dr - L - 1) \\
EffImg \;\rightarrow\; (con - 58, APL - 1, MPL - 1GAPL - 1) \\
Vgg16Img \;\rightarrow\; (con - 13, MPL - 5, DL - 2, Dr - L - 1, FL - 1) \\
\}
\end{gathered}
$$

Res Img,**Ince**Img,**Eff**Img,**Vgg16**Img→Res 0Img,Ince 0Img,Eff 0Img,Vgg 0Img

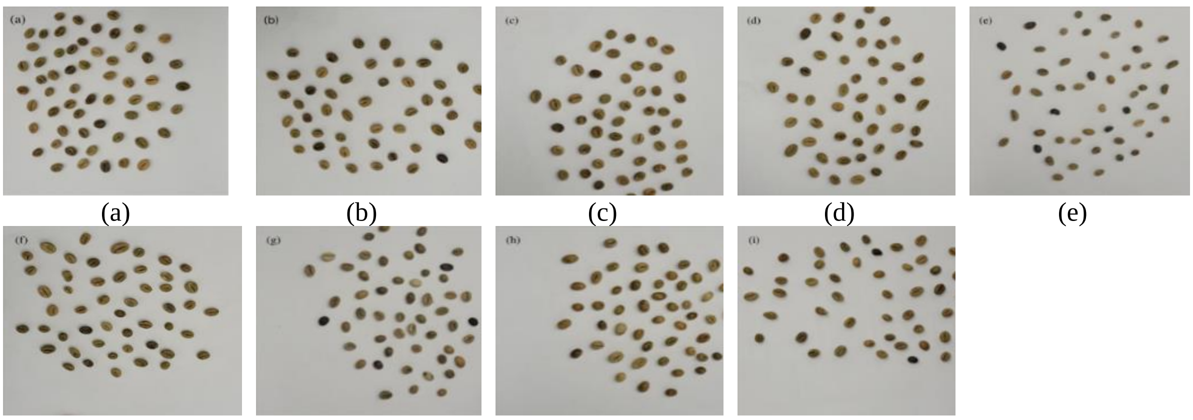
2.7. Data collection

Table 1 describes the dataset. The dataset consists of 50 images for each grade of coffee beans. These images were captured using a mobile phone under controlled illumination, and the beans used in the experiment are close to a year old from harvest. This dataset's potential processing includes using the images to classify the grade of the beans, predict the quality, and estimate the size and weight of the beans based on their appearance. This dataset was acquired from coffee estates in Wayanad, Kerala.

Table 1. Data Collection

Grade	Category -1	Category -2	Category -3	Category -4	Category -5	Category -6	Category -7	Category -8	Category -9
No of images	50	50	50	50	50	50	50	50	50
Source	Kerala, Wayanad								

A selection of example images demonstrating various coffee bean grades are shown in Figure 6. Among these grades is (a) grade A, which denotes superior beans that adhere to exacting standards for size, shape, color, and general appearance. In (b) grade AA designates exceptional quality with even more coveted qualities like homogeneity and shape. The highest quality, (c) grade AAA, includes remarkable qualities in terms of size, shape, color, and overall look. In (d) grade AB designates a mixture of beans with different qualities, sometimes including cosmetic flaws. Beans with a (e) grade C are of lesser quality and have noticeable flaws or irregularities. Beans that have been broken or dispersed make up (f) grade bits. Specific to particular coffee kinds, (g) grades PB-I and (h) PB-II denote particular size, shape, or quality attributes. Last-grade bulk refers to a generic category or mixture of beans that may not meet specific quality criteria for higher grades. Figure 6 describes the quality classifications of coffee beans across different grades from grade A, AA, AAA, AB, C, bits, PB-I, PB-II and bulk.



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Figure 6. Grades of coffee beans: Illustrating the quality classifications of coffee beans across different grades from: (a) grade A, (b) AA, (c) AAA, (d) AB, (e) C, (f) bits, (g) PB-I, (h) PB-II and (i) bulk

3. RESULTS AND DISCUSSION

In Table 2, we describe ResNet34. The architecture includes three convolutional layers, each followed by a batch normalization layer, a ReLU activation function, a global average pooling layer, and a fully connected layer. The skip connections are implemented using residual blocks, allowing for the gradients to flow directly from one layer to another. The model has achieved impressive results on various image classification tasks.

Table 2. ResNet34

Layer type	Convolutional Layers	Global Average Pooling Layers	Batch Normalization Layers	Fully Connected Layers
No. of layers	3	1	1	1

Table 3 describes Inception v3. The Inception v3 is a CNN architecture designed to balance depth and computational efficiency in deep networks. It includes four convolutional layers, six max-pooling layers, one batch normalization layer, seven inception modules, two dense layers with dropout, and a softmax activation function. Inception modules are multi-branch networks that allow different receptive field sizes to be learned simultaneously, enabling the Network to capture features at different scales and resolutions. The Inception v3 has achieved impressive accuracy on the ImageNet dataset and has been used in transfer learning for a variety of applications, including object detection and semantic segmentation.

Table 3. Inception v3

Layer type	Convolutional Layers	Inception Modules (Total)	Batch Normalization Layers	Dense Layers	Max-Pooling Layers	Dropout Layers
No. of layers	4	7	1	2	6	1

In Table 4, the model consists of 58 convolutional layers that use a variety of learned filters to extract characteristics at various spatial scales from the input image. One max-pooling layer follows the convolutional layers, and it downsamples the feature maps by taking the highest value found in each pooling zone. Additionally, the model has a single average pooling layer that takes the mean value across all pooling regions. A final layer computes the average value of each feature map across all spatial locations. This layer is called global average pooling. This results in a single feature vector representing the entire image and may be applied to downstream tasks like classification. The EfficientNet-B0 model is a deep convolutional neural network that, to extract features and provide a compact representation of the input image, combines convolutional layers, max-pooling layers, and pooling layers.

Table 4. EfficientNet-B0

Layer type	Convolutional Layers	Max-Pooling Layers	Average Pooling Layers	Global Average Pooling Layers
No. of layers	58	1	1	1

Table 5 describes VGG-16; the model consists of 13 convolutional layers that use a variety of learned filters to extract characteristics at various spatial scales from the input image. Max-pooling layers, which downsample the feature maps and aid in lowering the representation's dimensionality, are frequently added after these convolutional layers. Two dense layers follow the convolutional and max-pooling layers and collectively categorize the collected features into the required classes. The model also includes a single dropout layer that, while being trained, randomly eliminates part of the connections between the previous and subsequent layers. This enhances the model's generalization capabilities and prevents overfitting.

Table 5. VGG-16

Layer type	Convolutional Layers	Flatten Layers	Dense Layers	Max-Pooling Layers	Dropout Layers
No. of layers	13	1	2	5	1

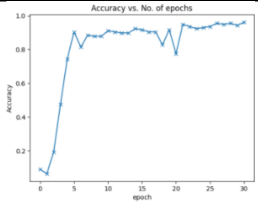
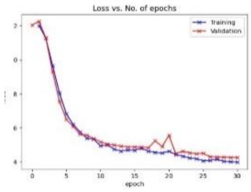
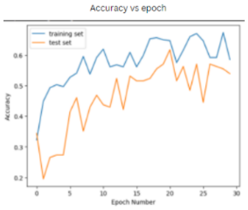
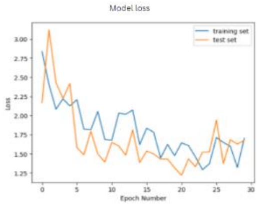
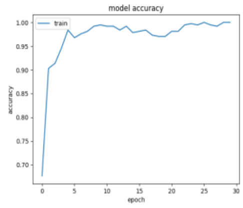
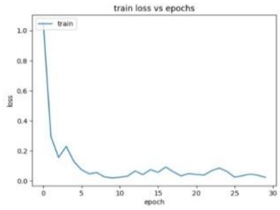
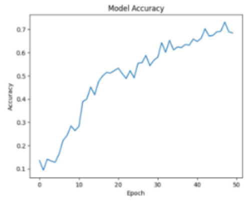
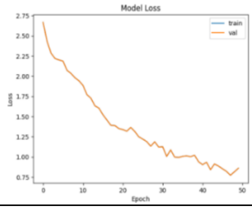
Table 6 presents the experimental results of different deep-learning models used for grading coffee beans. The table includes information on the model, train-test ratio, number of epochs, number of batches, accuracy, train loss, and validation loss. Here is a summary of the findings: ResNet34 consistently achieved high accuracy across various train-test ratios, epochs, and batches, ranging from 0.9615 to 0.9907. In contrast, Inception v3 achieved lower accuracy compared to ResNet34, ranging from 0.5782 to 0.7273. The accuracy varied across different configurations, and the train and validation losses showed some variability as well. EfficientNet-B0 performed similarly to ResNet34 in terms of accuracy, ranging from 0.9599 to 1.0000. On the other hand, VGG-16 displayed accuracy ranging from 0.5915 to 0.7543, which was comparatively lower than the other models. The train and validation losses for VGG-16 were relatively higher, suggesting some challenges in the model's performance.

Table 6. Experimental results

Id	Model	Train: Test ratio	No. of epochs	No. of batches	Accuracy	Train_loss	Validation_loss
1.0	ResNet34	70:30	30	13	0.9615	1.3981	1.4242
1.1	ResNet34	70:30	40	12	0.9653	1.4210	1.4184
1.2	ResNet34	70:30	50	12	0.9722	1.4161	1.4142
1.3	ResNet34	80:20	30	12	0.9630	1.4007	1.4050
1.4	ResNet34	80:20	40	12	0.9907	1.3876	1.3917
1.5	ResNet34	80:20	50	12	0.9907	1.3805	1.3944
1.6	ResNet34	90:10	30	12	0.9792	1.4019	1.4368
1.7	ResNet34	90:10	40	12	0.9333	1.3875	1.4531
1.8	ResNet34	90:10	50	12	0.9583	1.3863	1.4183
2.0	Inception v3	70:30	30	10	0.5850	1.7035	1.6752
2.1	Inception v3	70:30	40	10	0.5782	1.4938	1.7988
2.2	Inception v3	70:30	50	10	0.6361	1.3084	1.4761
2.3	Inception v3	80:20	30	10	0.6441	1.3451	1.0304
2.4	Inception v3	80:20	40	11	0.7088	1.0086	1.7242
2.5	Inception v3	80:20	50	11	0.7273	0.9647	1.0599
2.6	Inception v3	90:10	30	13	0.6269	1.6958	1.8214
2.7	Inception v3	90:10	40	13	0.6528	1.2830	2.0963
2.8	Inception v3	90:10	50	13	0.6658	1.3848	2.0412
3.0	Efficient net-b0	70:30	30	12	0.9599	0.1381	-
3.1	Efficient net-b0	70:30	40	11	0.9907	0.0388	-
3.2	Efficient net-b0	70:30	50	11	0.9846	0.0662	-
3.3	Efficient net-b0	80:20	30	12	1.0000	0.0077	-
3.4	Efficient net-b0	80:20	40	12	0.9946	0.0138	-
3.5	Efficient net-b0	80:20	50	12	0.9946	0.0164	-
3.6	Efficient net-b0	90:10	30	14	0.9856	0.0504	-
3.7	Efficient net-b0	90:10	40	12	0.9808	0.0552	-
3.8	Efficient net-b0	90:10	50	14	0.9880	0.0349	-
4.0	VGG-16	70:30	30	322	0.5915	1.0469	1.0748
4.1	VGG-16	70:30	40	322	0.6549	0.8873	0.8532
4.2	VGG-16	70:30	50	322	0.6729	0.7469	0.8029
4.3	VGG-16	80:20	30	372	0.6201	1.0355	0.8955
4.4	VGG-16	80:20	40	372	0.6667	0.9301	0.8516
4.5	VGG-16	80:20	50	372	0.7543	0.8600	0.8834
4.6	VGG-16	90:10	30	414	0.70	1.1127	0.8622
4.7	VGG-16	90:10	40	414	0.6800	0.8369	0.7087
4.8	VGG-16	90:10	50	414	0.7400	0.7815	0.5522

ResNet34 and EfficientNet-B0 consistently performed well across different train-test ratios, epochs, and batches, achieving high accuracy with low train losses. Inception v3 showed relatively lower accuracy, and VGG-16 had lower accuracy with higher train and validation losses. Overall, ResNet34 and EfficientNet-B0 models are recommended for grading coffee beans due to their higher accuracy and better performance. Table 7 provides an analysis of deep learning models, including ResNet34, Inception v3, EfficientNet-B0, and VGG16. The table includes columns for the model name, input image details, accuracy by epoch, loss by epoch, and predicted image. In the "Accuracy X Epoch" column, the x-axis represents the epoch number, indicating the different stages or iterations of the model training process. The y-axis represents the accuracy value achieved by the model at each corresponding epoch. In the "Loss X Epoch" column, the x-axis signifies the epoch number, representing the different stages or iterations of the model training process. The y-axis denotes the loss or error value experienced by the model at each corresponding epoch.

Table 7. Model Accuracy and predicted output.

Model	Input image	Accuracy x epoch	Loss x epoch	Predicated Image
ResNet34				
Inception v3				
EfficientNet-B0				
VGG-16				

4. CONCLUSION

A dataset of 464 images of nine different coffee bean grades was used in the study to test four deep-learning models. The findings revealed that ResNet34, EfficientNet-B0, VGG-16, and Inception v3 had the best accuracy of 99.07% and the lowest train and validation losses. There are various potential future paths for study based on the outcomes of the deep learning models on the grading of coffee beans. Expanding the dataset to boost the model accuracy is one possible direction for future research. Furthermore, applying transfer learning and fine-tuning methods may improve the performance of the models even further.

ACKNOWLEDGEMENTS

This research has been supported by the Amrita School of Computing, Amrita Vishwa Vidyapeetham, Mysuru campus.

REFERENCES

[1] Z. Ünal and H. Aktaş, "Classification of hazelnut kernels with deep learning," *Postharvest Biology and Technology*, vol. 197, Mar. 2023, doi: 10.1016/j.postharvbio.2022.112225.

[2] M. Dogan, Y. S. Taspinar, I. Cinar, R. Kursun, I. A. Ozkan, and M. Koklu, "Dry bean cultivars classification using deep cnn features and salp swarm algorithm based extreme learning machine," *Computers and Electronics in Agriculture*, vol. 204, Jan. 2023, doi: 10.1016/j.compag.2022.107575.

[3] K. Masuda *et al.*, "Application of deep learning diagnosis for multiple traits sorting in peach fruit," *Postharvest Biology and Technology*, vol. 201, Jul. 2023, doi: 10.1016/j.postharvbio.2023.112348.

[4] M. K. Tripathi and D. D. Maktedar, "Internal quality assessment of mango fruit: an automated grading system with ensemble classifier," *The Imaging Science Journal*, vol. 70, no. 4, pp. 253–272, May 2022, doi: 10.1080/13682199.2023.2166657.

[5] A. Prabhu, N. Shobha Rani, and H. T. Basavaraju, "An orientation independent vision based weight estimation model for Alphonso mangoes," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 5, pp. 7257–7275, May 2023, doi: 10.3233/jifs-223510.

[6] S. Shedole, S. B J, and N. A. V.P, "A convolution neural network-based wheat grain classification system," *Journal of Scientific Research*, vol. 66, no. 02, pp. 22–29, 2022, doi: 10.37398/jsr.2022.660204.

[7] Q. Liu, M. Fang, Y. Li, and M. Gao, "Deep learning based research on quality classification of shiitake mushrooms," *LWT*, vol. 168, Oct. 2022, doi: 10.1016/j.lwt.2022.113902.

[8] M. Knott, F. Perez-Cruz, and T. Defraeye, "Facilitated machine learning for image-based fruit quality assessment," *Journal of*

Food Engineering, vol. 345, May 2023, doi: 10.1016/j.jfoodeng.2022.111401.
[9] Y. Fu, M. Nguyen, and W. Q. Yan, "Grading methods for fruit freshness based on deep learning," *SN Computer Science*, vol. 3, no. 4, Jul. 2022, doi: 10.1007/s42979-022-01152-7.
[10] M. Mukhiddinov, A. Muminov, and J. Cho, "Improved classification approach for fruits and vegetables freshness based on deep learning," *Sensors*, vol. 22, no. 21, Oct. 2022, doi: 10.3390/s22218192.
[11] B. J. B. Nair, K. M. A. Nanda, A. S. Shalwin, M. G. L. Rao, and R. V, "Coffee Bean Grading Based on Weight Estimation Using Densenet121 Model," in *2023 7th International Conference On Computing, Communication, Control And Automation (ICCUBEA)*, Aug. 2023, pp. 1–6, doi: 10.1109/ICCUBEA58933.2023.10392243.
[12] A. Jahanbakhshi, Y. Abbaspour-Gilandeh, K. Heidarbeigi, and M. Momeny, "Detection of fraud in ginger powder using an automatic sorting system based on image processing technique and deep learning," *Computers in Biology and Medicine*, vol. 136, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104764.
[13] L. Deng, J. Li, and Z. Han, "Online defect detection and automatic grading of carrots using computer vision combined with deep learning methods," *LWT*, vol. 149, p. 111832, Sep. 2021, doi: 10.1016/j.lwt.2021.111832.
[14] F. Yuesheng *et al.*, "Circular fruit and vegetable classification based on optimized GoogLeNet," *IEEE Access*, vol. 9, pp. 113599–113611, 2021, doi: 10.1109/access.2021.3105112.
[15] B. S. Anami, N. N. Malvade, and S. Palaiah, "Deep learning approach for recognition and classification of yield affecting paddy crop stresses using field images," *Artificial Intelligence in Agriculture*, vol. 4, pp. 12–20, 2020, doi: 10.1016/j.aiia.2020.03.001.
[16] M. M. Raikar, M. S M, C. Kuchanur, S. Girraddi, and P. Benagi, "Classification and grading of okra-ladies finger using deep learning," *Procedia Computer Science*, vol. 171, pp. 2380–2389, 2020, doi: 10.1016/j.procs.2020.04.258.
[17] A. Saleh, R. Sukaik, and S. S. Abu-Naser, "Brain tumor classification using deep learning," in *2020 International Conference on Assistive and Rehabilitation Technologies (iCareTech)*, Aug. 2020, pp. 131–136, doi: 10.1109/iCareTech49914.2020.00032.
[18] Shivani and S. Singh, "Fruit disease detection using convolution neural network approach," *Asian Journal of Computer Science and Technology*, vol. 7, no. 2, pp. 62–65, Aug. 2018, doi: 10.51983/ajcst-2018.7.2.1871.
[19] V. Bhole and A. Kumar, "Mango quality grading using deep learning technique: Perspectives from agriculture and food industry," in *Proceedings of the 21st Annual Conference on Information Technology Education*, Oct. 2020, pp. 180–186, doi: 10.1145/3368308.3415370.
[20] D. G. Brosas, R. S. Villafuerte, and D. C. Obediencia, "Adaptive neuro-fuzzy approach for cacao bean grading classification process," in *2020 Third International Conference on Vocational Education and Electrical Engineering (ICVEE)*, Oct. 2020, pp. 1–5, doi: 10.1109/icvee50212.2020.9243281.
[21] C. Zhang *et al.*, "An instance-based deep transfer learning method for quality identification of Longjing tea from multiple geographical origins," *Complex & Intelligent Systems*, vol. 9, no. 3, pp. 3409–3428, Mar. 2023, doi: 10.1007/s40747-023-01024-4.
[22] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019, doi: 10.1109/tgrs.2019.2907932.
[23] N.-F. Huang, D.-L. Chou, and C.-A. Lee, "Real-time classification of green coffee beans by using a convolutional neural network," in *2019 3rd International Conference on Imaging, Signal Processing and Communication (ICISPC)*, Jul. 2019, pp. 107–111, doi: 10.1109/icispc.2019.8935644.
[24] S. Wallelign, M. Polceanu, T. Jemal, and C. Buche, "Coffee grading with convolutional neural networks using small datasets with high variance," *Journal of WSCG*, vol. 27, no. 2, pp. 113–120, 2019, doi: 10.24132/jwscg.2019.27.2.4.
[25] F. Faridah, G. O. F. Parikesit, and F. Ferdiansjah, "Coffee bean grade determination based on image parameter," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 9, no. 3, pp. 547–554, Dec. 2011, doi: 10.12928/telkomnika.v9i3.747.

BIOGRAPHIES OF AUTHORS

Bipin Nair Balakrishnan Jayakumari is currently an assistant professor at the Department of Computer Science, Amrita School of Computing, Amrita Vishwa Vidyapeetham, Mysuru Campus. His research interests include image analysis, machine learning and deep learning. He has published more than 90 research papers in reputable national and international journals. All those research papers are indexed in leading indexing databases such as Scopus and SCI. He can be contacted at bipin.bj.nair@gmail.com.

Abrav Nanda Koovamoola Mambilamthoda is a final year student of BCA pursuing a dissertation in the area of image classification at Amrita School of Computing, Amrita Vishwa Vidyapeetham, Mysuru Campus. His research interests include image analysis. He can be contacted at abravnanda97000@gmail.com.

Shalwin Ambalamoottil Stephen is a final year student of BCA pursuing a dissertation in the area of image classification at Amrita School of Computing, Amrita Vishwa Vidyapeetham, Mysuru Campus. His research interests include image analysis. He can be contacted at shalwinas@gmail.com.

Pranav Venkitesan is an accomplished senior software engineer at Zenloop GMBH in Berlin, Germany. His area of specialization is document image processing, medical image processing, image analysis, and bioinformatics. He can be contacted at pranavmechary@gmail.com.

Venkatesh Raghavendra is a final year student of BCA pursuing a dissertation in the area of document image classification and deep learning at Amrita School of Computing, Amrita Vishwa Vidyapeetham, Mysuru Campus. His research interests include image analysis. He can be contacted at raghavendrapraveen45@gmail.com.