Journal Homepage: - www.journalijar.com

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)

Article DOI: 10.21474/IJAR01/19402
DOI URL: http://dx.doi.org/10.21474/IJAR01/19402

*RESEARCH ARTICLE*

**ENHANCED CLASSIFICATION OF COFFEE BEAN SPECIES THROUGH COMPUTATIONAL MODELING AND IMAGE ANALYSIS**

**Nagashree B.[1], Rishika Mohan V.[1], Shreya Lal[1], Shweta Bathija[1], Dr. Shivandappa[2] and Dr. Ajeet Kumar S.[2]**

1. Undergraduate Student, Department of Biotechnology, R.V. College of Engineering, Bengaluru.
2. Assistant Professor, Department of Biotechnology, R.V. College of Engineering, Bengaluru.

**Manuscript Info**

***Manuscript History***
Received: 28 June 2024
Final Accepted: 30 July 2024
Published: August 2024

***Key words:-***
Image Classification, Confusion Matrix, Convolutional Neural Network (CNN), Datasets

**Abstract**

Maintaining quality standards and streamlining the value chain in the coffee business depend heavily on the accurate classification of coffee beans. Previously, this procedure was carried out by hand, which frequently resulted in errors that affected the beans' overall quality and market value. In recent times however, there have been chances to improve and automate coffee bean sorting accuracy owing to developments in deep learning, especially in the area of image classification. Fraud detection is very important to ensure that the consumers receive quality products. This work investigates the use of deep learning models to categorize coffee beans using image data. By using machine learning and computer vision technologies, we are able to analyze the data given to identify the irregularities found which may result in fraud or issues in the quality of the coffee beans thereby proving that using image classification algorithms can majorly reduce the errors formed with respect to manual sorting, which ultimately leads to improved control of quality and economic outcomes.



## Introduction:-

Coffee is one of the world's most popular beverages, ranking second only to water, and the sector has undergone considerable technological developments over time. There are several important types of coffee beans, each with distinct features that influence flavor profile, growth circumstances, and market demand. Three coffee bean kinds were chosen based on availability, demand, growth conditions, provenance, and regions of consumption. Arabica is the most popular coffee varietal, accounting for around 60-70% of global coffee production. It is often grown at higher altitudes, particularly in Ethiopia, Colombia, and Indonesia. Arabica beans are distinguished by their smooth and mild flavor, as well as a higher acidity level than other types. They are more sensitive and necessitate specific growing circumstances, such as a consistent environment with ample rainfall. While Arabica beans are valued for their excellent flavour, they are more susceptible to pests and diseases, making them more expensive to produce. Robusta beans, on the other hand, are often stronger, more bitter, and higher in caffeine than Arabica beans. They are generally grown at lower elevations in countries like Vietnam, Brazil, and sections of Africa. Robusta beans are more disease resistant and can handle tougher growing conditions, resulting in lower production costs. Because of their strong and bitter flavour, they are frequently used in espresso blends or instant coffee, and they are a popular choice for commercial coffee products due to their high yield and inexpensive manufacturing costs. The third type is Kaveri, which is a hybrid of Arabica and Robusta coffees. It is very disease-resistant, making it an excellent choice for farmers in places where coffee crops are threatened by pests and plant diseases. Kaveri beans offer a mix

**Corresponding Author:- Nagashree B.**
Address:- Undergraduate Student, Department of Biotechnology, R.V. College of Engineering, Bengaluru.

between the mildness of Arabica and the strength of Robusta, making them suitable for a variety of blends. This type is especially popular in parts of India, where it is produced for its durability and balanced flavor profile.

While the coffee business has achieved considerable technological advances, the issues of maintaining authenticity and quality remain. Many coffee companies continue to use manual sorting procedures, which are labor intensive, time consuming, and prone to human mistake. This can result in inconsistent quality and, more importantly, financial losses for manufacturers. To address these problems, automated classification systems have gained popularity as a means of improving accuracy and efficiency in coffee bean sortingImage classification is a machine learning technique that automates sorting and identifying objects by analyzing visual features. In coffee production, it helps in distinguishing between different coffee bean varieties and detecting defects, improving quality control, reducing human error, and lowering labor costs. Additionally, image classification is used in fraud detection by analyzing visual data from documents and financial records to identify anomalies and prevent fraud. This technology enhances accuracy and security across various industries.

**Image Classification**

Image classification is a machine learning technique used to categorize images into predefined classes based on their content. This process begins with feature extraction, which analyzes the attributes such as color, shape, and texture. During model training, these networks are fed with labeled datasets where they learn to recognize patterns and features associated with each class which can classify new, unseen images by predicting their class based on the learned features. Image classification techniques include transfer learning, which involves using pre-trained models on large datasets and fine-tuning them for specific tasks. This approach helps improve model performance and reduces the time required for training from scratch. Image classification plays a crucial role in fraud detection by analyzing visual data from documents and signatures to identify anomalies that may indicate fraudulent activity. By recognizing subtle patterns and discrepancies that are difficult for humans to detect, image classification can flag altered documents, fake signatures, or manipulated financial records. This enhances both the accuracy and speed of fraud detection, helping to prevent financial losses and improve security.

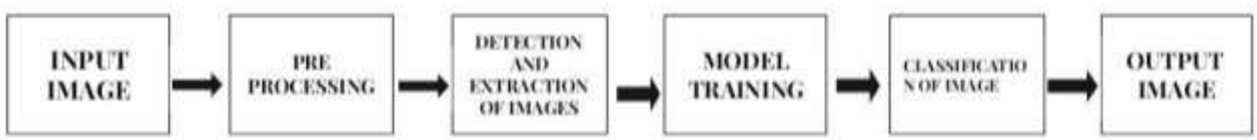



**Fig 1 (a):-** Process of image classification.

**Confusion Matrix**

A confusion matrix is a tool used to evaluate the performance of a classification model by comparing its predicted classifications to the actual outcomes. It is structured as a table with four main components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). In the matrix, rows represent actual classes, while columns represent predicted classes. The cells of the matrix show the counts of predictions for each combination of actual and predicted classes. The top-left cell shows TP, or correctly predicted positive cases, whereas the top-right cell shows FP, or cases incorrectly predicted as positive. Similarly, the bottom-left cell shows FN, or cases incorrectly predicted as negative, and the bottom-right cell shows TN, or correctly predicted negative cases. This matrix helps in calculating performance metrics such as accuracy, which is the proportion of correctly classified instances; precision, which measures the proportion of true positives among all positive predictions; recall, which indicates the proportion of true positives among all actual positives; and the F1 score, which balances precision and recall. Analyzing the confusion matrix allows for a comprehensive understanding of a model's performance and highlights areas for potential improvement.

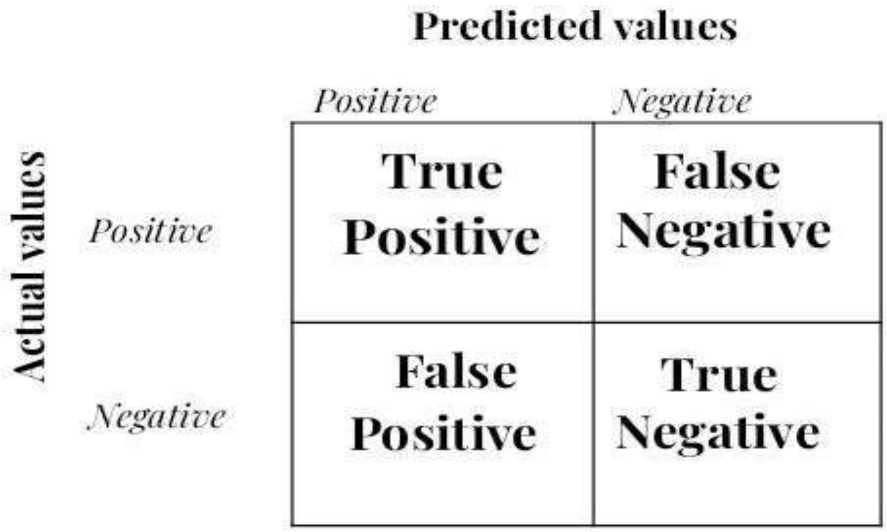



**Fig 1 (b):-** Confusion matrix.

**CNN Model**

Convolutional Neural Network (CNN) is a deep learning model engineered for the analysis of visual data. It administers through a set of layers which process and learn from image features in a tiered manner. Filters are applied to the input image by the convolution layers to identify features like the edges and textures. The filters generate feature maps which highlight various aspects of the images. Using functions like ReLU (Rectified Linear Unit), the activation layers introduce non-linearity, thereby permitting the model to capture even complex patterns. Next up are the pooling layers, which limit the spatial dimensions of the feature maps while sustaining important information, customarily using max pooling to select the maximum value in each region. The network includes fully connected layers which merge the learned features and produce the final predictions. Output layers provide the classification results by using activation functions like softmax for multi-class problems or sigmoid for binary ones. CNNs learn by adjusting the weights of the filters through backpropagation and optimization techniques, whilst training, thus allowing them to identify and sort the patterns in images. They are widely used in image classification, object detection, and facial recognition owing to their ability to understand and process visual data optimally.

## Methodology:-

**Preparing the Dataset**

Selecting the images of the coffee beans is the first step and involves compiling and sorting images of three coffee bean types-Kaveri, Arabica, and Robusta-into separate folders which should be labeled accordingly. This is followed by the loading and preprocessing operations wherein the images get loaded using the `PIL` library, and are cropped to the desired size (225x225 pixels). They are finally resized using Lanczos resampling, converted to numpy arrays and labeled.

**Splitting the Dataset**

The dataset is divided into two sections- training and testing- using an 80-20 split through `train_test_split`. This assures that the model can be evaluated on unseen data. Next, the pixel values of the images are normalized by dividing by 255.0 to scale them to a [0, 1] range, thus helping in faster convergence during training.

**Data Augmentation**

By implementing transformations such as rotation, zoom, and flips, `ImageDataGenerator` is used to create variations of the training data. This reduces overfitting and improves the model's generalization.

**Model Framework**

A sequential Convolutional Neural Network (CNN) model is built with multiple layers; (i) Three convolutional layers with increasing filters (32, 64, 128) and a (3x3) kernel size are used, each followed by a ReLU activation function, (ii) MaxPooling layers to reduce the spatial dimensions of the feature maps by half, (iii) Flattening the output from the final convolutional layer into a 1D vector, (iv) Dense Layer, each containing 128 neurons and a

ReLU activation function is added, followed by a dropout layer (0.5) to prevent overfitting, (v) A final dense output Layer with a softmax activation function outputs probabilities for each class (Kaveri, Arabica, Robusta).

**Training the Model**

The model is compiled using the Adam optimizer, sparse categorical cross-entropy loss function, and accuracy as the evaluation metric and is trained on the augmented training data for 15 epochs, with validation being performed on a subset of the training data.

**Testing**

After training, a confusion matrix is computed using 'sklearn.metrics.confusion_matrix' and the model's performance is evaluated on the test set. The confusion matrix helps visualize the classification accuracy of the model for each type of coffee bean and is displayed using `ConfusionMatrixDisplay` with the results visualized in a plot.

**Model Validation Using Sample Images**

In the step, a sample consisting of 16 random images are selected from the test set to verify the model's predictions and they are displayed along with their predicted and true labels, providing a clear visual representation of the model's classification performance.

**Analysis Of The Results**

The final step is to analyze the results, discussing the performance efficiency of the model based on accuracy, confusion matrix results, and visual validation.

**CODE**

```
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow import keras

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import random
import pathlib
import os
tf.random.set_seed(42)
train_data=keras.utils.image_dataset_from_directory('/content/coffee', validation_split=0.2, subset='training',
seed=1, shuffle=True, batch_size=32, image_size=(256,256) )
test_data=keras.utils.image_dataset_from_directory('/content/coffee', validation_split=0.2, subset='validation',
seed=1, shuffle=True, batch_size=32, image_size=(256,256))
filenames=pathlib.Path('/content/coffee')
for label in train_data.class_names:
  images=list(filenames.glob(f'{label}/*'))
  print(f'{label}: {len(images)}')
train_data.cardinality().numpy(), test_data.cardinality().numpy()
plt.figure(figsize=(8,5))
for images, labels in train_data.take(1):
  for i in range(15):
  index=random.randint(0,len(images)-1) # subtract 1 to ensure the index is within bounds
  ax=plt.subplot(3,5,i+1)
  plt.imshow(images[index].numpy().astype('uint8'))
  plt.title(train_data.class_names[labels[index]], color='blue', fontsize=12)
  plt.axis(False)
plt.show()
for images_batch, labels_batch in train_data:
```

```
  print(images_batch.shape)
  print(labels_batch.shape)
  break
from tensorflow.keras import layers
tf.random.set_seed(42)
cnn_1 = keras.Sequential([layers.Rescaling(1./255),
 layers.Conv2D(filters = 32, kernel_size = 3, activation = 'relu'),
layers.MaxPooling2D(pool_size = 2),
layers.Flatten(),
layers.Dense(500, activation = 'relu'),
layers.Dense(5, activation = 'sigmoid')])
tf.random.set_seed(42)
cnn_1 = keras.Sequential([
   layers.Rescaling(1./255),
   layers.Conv2D(filters = 32, kernel_size = 3, activation = 'relu'),
   layers.MaxPooling2D(pool_size = 2),
   layers.Flatten(),
   layers.Dense(500, activation = 'relu'),
   layers.Dense(5, activation = 'sigmoid')])
cnn_1.compile(
   loss = keras.losses.SparseCategoricalCrossentropy(),
   optimizer = keras.optimizers.Adam(),
   metrics = ['accuracy'] # Changed the string to a list
)
history_1 = cnn_1.fit(train_data, epochs = 5, validation_data = test_data)
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix
from tensorflow import keras
import tensorflow as tf
# Assuming your test_data is already preprocessed and available
# Get predictions from the model
predictions = cnn_1.predict(test_data)
predicted_labels = np.argmax(predictions, axis=1)
# Get true labels from the test data
true_labels = np.concatenate([y for x, y in test_data], axis=0)
# Compute confusion matrix
cm = confusion_matrix(true_labels, predicted_labels)
# Plot confusion matrix
plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=[f'Class {i}' for i in range(5)], yticklabels=[f'Class
{i}' for i in range(5)])
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.show()
```

## Results:-

Analysis of the results showed that the convolutional neural network (CNN) which was built to classify coffee bean types—Kaveri, Robusta, and Arabica—showed promising metrics during training as well as in the validation phases. The model then displayed rapid convergence over 15 epochs and attained near-perfect training accuracy at the end of the last epoch (99.94%), with the validation accuracy nearing a high 99.36%. This really low validation loss number of 0.0364 also ensures that the model is very generalizable and not overfitting, so it can be trusted to use in practical scenarios with real-life inputs. The confusion matrix provides more detail about how well the model did for each class; having very high precision over all classes. More precisely, it identifies 623 of the 628 Kaveri dataset images as true while misclassifying only 5 instances as Robusta. The model had a bit of a harder time with Robusta, differently classifying 10 out of 119 instances as Kaveri in the test set. The contrast with Arabica bean images however is stark, as the model was able to precisely classify all 237 instances perfectly along with their type, being correctly recognized. This shows that the model is able to classify Arabica beans with a high level of accuracy, paving the way for very good feature extraction capabilities in recognizing this particular kind. Nonetheless, the 10 cases where Robusta was misclassified as Kaveri suggest their feature spaces might overlap a little, making it harder to distinguish one from another under certain scenarios. This suggests an area for improvement on feature engineering, perhaps adding more layers or using an L2 regularization to help enforce locally changing neurons in the architecture. Additionally, this can be further improved by stronger data augmentation techniques to teach the model better how these visually similar classes differ. In conclusion, the CNN model built for coffee bean classification showed promising results overall, especially with Arabica species. The results obtained thus far show that CNNs are well-suited for the classification of individual coffee beans, directly serving both food quality and agricultural control sectors, providing great potential for enhancing process design and consequent final product assurance. In the future, attempts to alleviate these identified classification ambiguities should be made, with further improvements aimed at enhancing usability in realistic settings.

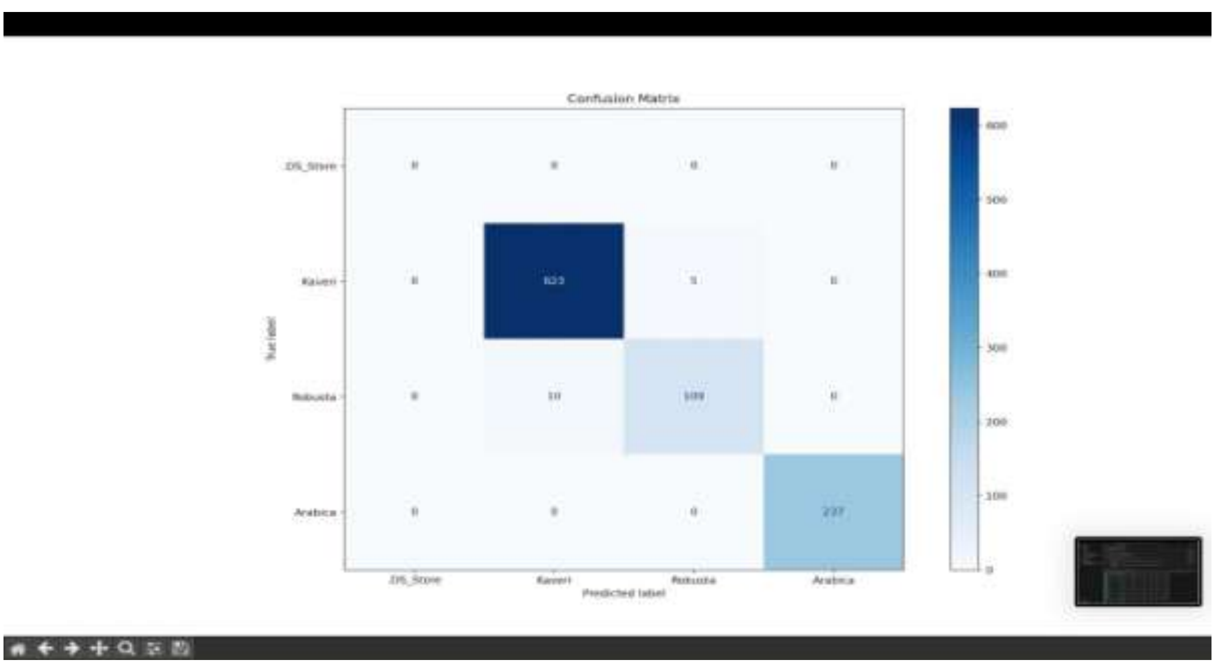
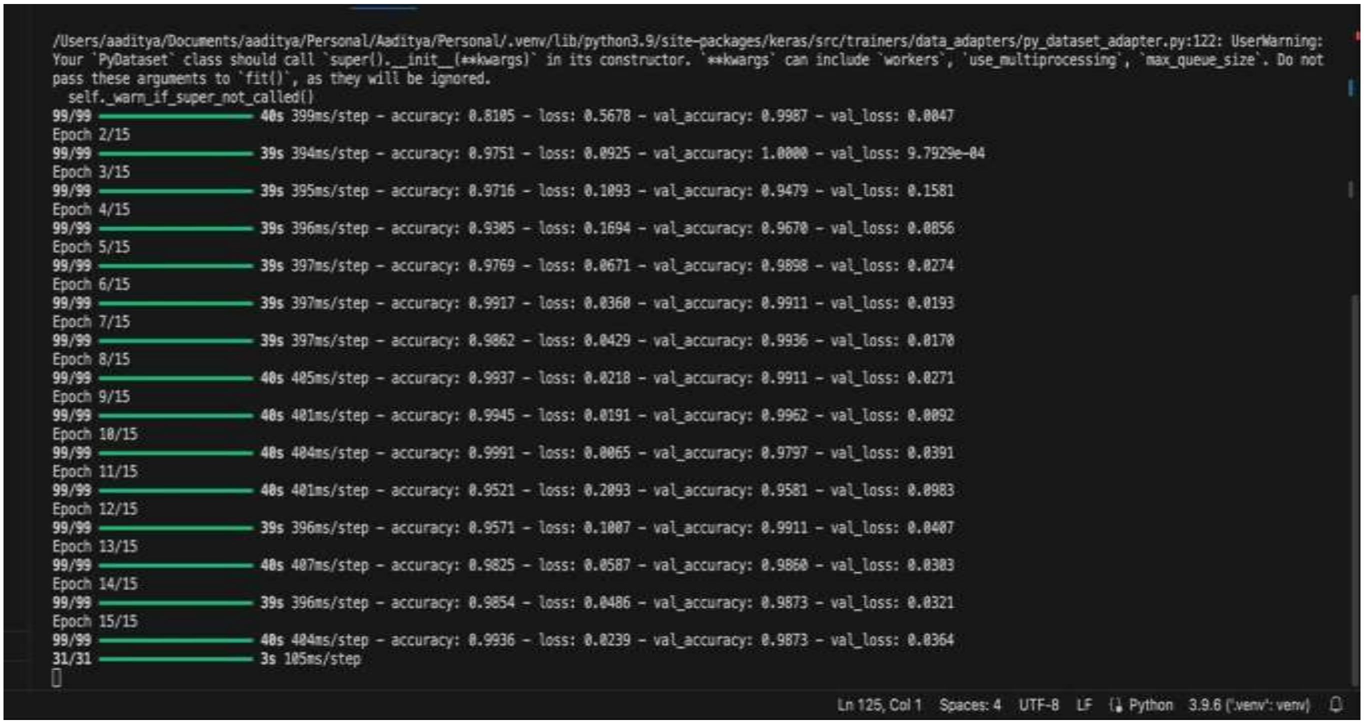




**Fig 3(b):-** Visualization of the accuracy of each type of coffee bean by using Confusion matrix.



```
/Users/aaditya/Documents/aaditya/Personal/Aaditya/Personal/.venv/lib/python3.9/site-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning:
Your `PyDataset` class should call `super().__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not
pass these arguments to `fit()`, as they will be ignored.
  self._warn_if_super_not_called()
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 399ms/step - accuracy: 0.8105 - loss: 0.5678 - val_accuracy: 0.9987 - val_loss: 0.0047
Epoch 2/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 394ms/step - accuracy: 0.9751 - loss: 0.0925 - val_accuracy: 1.0000 - val_loss: 9.7929e-04
Epoch 3/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 395ms/step - accuracy: 0.9716 - loss: 0.1093 - val_accuracy: 0.9479 - val_loss: 0.1581
Epoch 4/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 396ms/step - accuracy: 0.9305 - loss: 0.1694 - val_accuracy: 0.9670 - val_loss: 0.0856
Epoch 5/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 397ms/step - accuracy: 0.9769 - loss: 0.0671 - val_accuracy: 0.9898 - val_loss: 0.0274
Epoch 6/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 397ms/step - accuracy: 0.9917 - loss: 0.0360 - val_accuracy: 0.9911 - val_loss: 0.0193
Epoch 7/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 397ms/step - accuracy: 0.9862 - loss: 0.0429 - val_accuracy: 0.9936 - val_loss: 0.0170
Epoch 8/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 405ms/step - accuracy: 0.9937 - loss: 0.0218 - val_accuracy: 0.9911 - val_loss: 0.0271
Epoch 9/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 401ms/step - accuracy: 0.9945 - loss: 0.0191 - val_accuracy: 0.9962 - val_loss: 0.0092
Epoch 10/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 404ms/step - accuracy: 0.9991 - loss: 0.0065 - val_accuracy: 0.9797 - val_loss: 0.0391
Epoch 11/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 401ms/step - accuracy: 0.9521 - loss: 0.2093 - val_accuracy: 0.9581 - val_loss: 0.0983
Epoch 12/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 396ms/step - accuracy: 0.9571 - loss: 0.1007 - val_accuracy: 0.9911 - val_loss: 0.0407
Epoch 13/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 407ms/step - accuracy: 0.9825 - loss: 0.0587 - val_accuracy: 0.9860 - val_loss: 0.0303
Epoch 14/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 39s 396ms/step - accuracy: 0.9854 - loss: 0.0486 - val_accuracy: 0.9873 - val_loss: 0.0321
Epoch 15/15
99/99 ━━━━━━━━━━━━━━━━━━━━ 40s 404ms/step - accuracy: 0.9936 - loss: 0.0239 - val_accuracy: 0.9873 - val_loss: 0.0364
31/31 ━━━━━━━━━━━━━━━━━━━━ 3s 105ms/step
```



**Fig3 (c):-** Convolutional Neural Network Statistics.

**Applications**

Automated image classification of coffee beans can have a significant influence on different industry processes, improving not only operations but also the quality of products. Image classification automates the sorting process and makes sure that the coffee beans are sorted into various types such as Arabica, Robusta, and Kaveri and further graded based on their quality. This automation process allows any defective beans to be separated out that could have one of many problems (cracked, discolored, or funny shapes, etc.), and ensures the high standard is met for coffee production. For grading and sorting, image classification ensures things like beans are categorized correctly by size, color, or texture so that they all reach similar roast levels and flavors. Similarly, it means we can create blends with much higher levels of predictability for what a coffee will taste like on any given day and across large batches—this repeatability is key to success in specialty or commercial markets. Moreover, it is amending the manufacturing process by providing roast profiling options so one can pan out the most ideal roasting profile based on their coffee beans to ensure steady taste and quality. Moreover, it helps manage inventory while monitoring the flow of beans throughout the supply chain resulting in a more refined resource investment and operational cost reduction. Image classification has another significant advantage: cost reduction, as less manual sorting labor is required leading to lower labor costs along with increased throughput. It also cuts down on waste by ensuring that no broken beans make their way into the end product. That means the technology can confirm where beans were grown and what type of certification they have—something that could be important for high-end coffee brands, which often subscribe to certifications such as organic or fair trade. This guarantees that customers are authentic and ethically sourced products. Image Classification—Consumers will benefit when augmented with image classification to strengthen the overall coffee experience by maintaining product quality, consistency in blends, and aids in creating custom solutions based on individual tastes. A valuable tool for firms conducting R&D which deliver insights into how specific bean attributes affect flavor profiles driving new coffee varieties as well as the development of commercial cultivation and processing techniques. In addition, image classification in the coffee industry is good for sustainability including cutting down on energy and water usage as well during processing and roasting. These more efficient results will contribute to less waste and reduced environmental impact, which is increasingly important as the industry shifts towards becoming a more sustainable sector. This modernized the production of quality coffee and is yet another one of the advantages image classification has provided besides lifted, constant and a sustainable set of products. Its incorporation into a plethora of different coffee processing and producing stages illustrates the impact it has on high consumer demands as well as industry standards.

## Acknowledgments:-

The authors express their appreciation to the Department of Biotechnology, RVCE and RSST Trust Bangalore for their continuous support and encouragement.


## References:-


[1]Manoj Krishna, M., Neelima, M., Harshali, M., & Venu Gopala Rao, M. (2018). Image classification using deep learning. International Journal of Engineering & Technology, 7(2.7), 614. https://doi.org/10.14419/ijet.v7i2.7.10892

[2]Arboleda, E. R., Fajardo, A. C., & Medina, R. P. (2018). Classification of coffee bean species using image processing, artificial neural network and K nearest neighbors. In 2018 IEEE International Conference on Innovative Research and Development (ICIRD) (pp. 1-5). IEEE. https://doi.org/10.1109/ICIRD.2018.8376326

[3]Hassan, E. (2024). Enhancing coffee bean classification: A comparative analysis of pre-trained deep learning models. Neural Computing and Applications, 36, 9023–9052. https://doi.org/10.1007/s00521-024-09623-z

[4]Amanin, N. N., & Saraswati, G. W. (2022). Classification of Arabica coffee green beans using digital image processing using the K-nearest neighbor method. Journal of Applied Intelligent System, 7(2), 111-119. https://doi.org/10.33633/jais.v7i2.6449

[5]El Jireh, P. B., Dionson, M. G. D., & Javier, R. K. T. (2024). Convolutional neural network architectures for classification of coffee varieties in the Philippines. In 2024 IEEE Symposium on Wireless Technology & Applications (ISWTA) (pp. 140-145). IEEE. https://doi.org/10.14445/22315381/IJETT-V72I6P128

[6]Luis, V. A. M., Quiñones, M. V. T., & Yumang, A. N. (2022). Classification of defects in Robusta green coffee beans using YOLO. In 2022 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET) (pp. 1-6). IEEE. https://doi.org/10.1109/IICAIET55139.2022.9936831

[7]Izza, M. N., & Kusuma, G. P. (2024). Image classification of green Arabica coffee using transformer-based architecture. International Journal of Engineering Trends and Technology, 72(6), 304–314. https://doi.org/10.14445/22315381/ijett-v72i6p128

[8]Adhitya, Y., Prakosa, S. W., Köppen, M., & Leu, J. S. (2020). Feature extraction for cocoa bean digital image classification prediction for smart farming application. Agronomy, 10(11), 1642. https://doi.org/10.3390/agronomy10111642

[9]Janandi, R., & Cenggoro, T. W. (2020). An implementation of convolutional neural network for coffee beans quality classification in a mobile information system. IEEE. https://doi.org/10.1109/icimtech50083.2020.9211257

[10]Huang, N., Chou, D., & Lee, C. (2019). Real-time classification of green coffee beans by using a convolutional neural network. IEEE. https://doi.org/10.1109/icispc.2019.8935644

[11]Lee, J., & Jeong, Y. (2022). Prediction of defect coffee beans using CNN. IEEE. https://doi.org/10.1109/bigcomp54360.2022.00046

[12]Mihailova, A., Liebisch, B., Islam, M. D., Carstensen, J. M., Cannavan, A., & Kelly, S. D. (2022). The use of multispectral imaging for the discrimination of Arabica and Robusta coffee beans. Food Chemistry X, 14, 100325. https://doi.org/10.1016/j.fochx.2022.100325

[13]Buonocore, D., Carratù, M., & Lamberti, M. (2023). Classification of coffee bean varieties based on a deep learning approach. In 2022 IMEKO TC10 International Conference on Measurement for Diagnostics, Optimization and Control. https://doi.org/10.21014/tc10-2022.002

[14]kaggle datasets download -d michals22/coffee-dataset.