AI Based Experimental Analysis for Discrimination of Cocoa Beans Using Structural Image Features

THESIS

Submitted by

SAMIKSHAN DAS

[Roll No. - 10060920015]

[Registration No. – 203000460910016]

Supervised by

Dr. Amitava Akuli

Joint Director

Centre for Development of Advanced Computing (C-DAC), Kolkata
West Bengal 700091, India

Guided by

Prof. Aniruddha Dey

In partial fulfillment for the award of the degree

Of

MASTER OF TECHNOLOGY

IN

Information Technology - Artificial Intelligence
2022

MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY
West Bengal, India

BONAFIDE CERTIFICATE

Certified that this project report **"AI BASED EXPERIMENTAL ANALYSIS FOR DISCRIMINATION OF COCOA BEANS USING STRUCTURAL IMAGE FEATURES"** is the bonafide work of **SAMIKSHAN DAS** who carried out the project work under my supervision.

Dr. Somdatta Chakravortty
HEAD OF THE DEPARTMENT
Dept. of Information Technology
MAKAUT, West Bengal, India

Dr. Amitava Akuli
SUPERVISOR
Joint Director
C-DAC, Kolkata, India

Prof. Aniruddha Dey
INTERNAL GUIDE
Dept. of Information Technology
MAKAUT, West Bengal, Indi

June, 2022
MAKAUT, West Bengal

External Examiner

ACKNOWLEDGEMENT

I would like to take this opportunity to thank everyone whose cooperation and encouragement throughout the ongoing course of this project remains invaluable to me.

I am sincerely grateful to my guide professor Aniruddha Dey, Assistant Professor, Maulana Abul Kalam Azad University of Technology and Dr. Amitava Akuli, Joint Director, C-DAC Kolkata, for their wisdom, guidance and inspiration that helped me to go through this project and take it to where it stands now.

I would also like to express my sincere gratitude to Prof. Somdatta Chakrabarty, HOD, Information Technology, Maulana Abul Kalam Azad University of Technology, West Bengal, for her encouragement.

Last but not the least, I would like to extend my warm regards to my family and peers who have kept supporting me and always had faith on me.

Samikshan Das
M.Tech in Artificial Intelligence
Reg. No.: 203000460910016
MAKAUT, West Bengal

ABSTRACT

This paper proposes a machine vision solution for classification of cocoa beans using structural characteristics like shape and size of the cocoa beans extracted from images using machine learning and deep learning techniques. For machine learning models the features are extracted from images through a chain of image processing techniques and traditional machine learning techniques (KNN, SVM, Decision Tree and Random Forest) and Convolutional Neural Network are employed to classify the cocoa beans in four classes i.e. large, medium, small and rejected. Comparative studies among different techniques are also performed. Prior to train the machine learning models, extracted features are optimized by two optimization techniques- Univariate Selection and Feature Importance. The CNN model is optimized with Adam optimizer. Trained models are evaluated using K-fold cross validation and finally mean cross validation scores are calculated for performance analysis. The experimental results show that among the Machine learning algorithms the Random Forest Classifier provides highest mean accuracy score of 0.75 while overall the CNN model predicts with maximum mean accuracy score 0.83.

Table of Contents

List of Figures

List of Tables

INTRODUCTION

Cocoa is the key ingredient in chocolate and chocolate confections. Chocolate is incredibly popular, and it is one of the common foods in the world. About 300 to 600 cocoa beans are required to make 1 kg of chocolate, but this may vary depending on the cocoa content in the beans. The cocoa nibs, cocoa paste (mass or liquor), butter, powder and couverture etc. are the byproducts of the cocoa beans. These are the main ingredients to make chocolate and other food products. The beans are also used to make cosmetics and soaps. About 70 percent of the world's cocoa beans come from West African countries like Ghana, Nigeria Ivory Coast, Indonesia and Cameroon.

	Country	**Production 2013**	**% of World Total**
1	Cote d Ivoire	1,448,992 m/t	33.8%
2	Ghana	835,466 m/t	15.4%
3	Indonesia	777,500 m/t	15.2%
4	Nigeria	367,00 m/t	8.7%
5	Cameroon	275,000 m/t	5.9%

Table-1: The Top 5 cocoa bean producing countries

After plucking of cocoa beans from the tree, it undergoes through a post-harvesting treatment. The first chemical process is fermentation [1]. It is one of the most important operations because the final quality is improved assuring the development of the cocoa flavor with sweet aroma. The cocoa quality attributes are strongly correlated with the degree of fermentation [2] as it involves processes such as reducing sugars, free amino acids and bean PH. A good fermentation also contributes to the reduction of astringency and bitterness of cocoa.

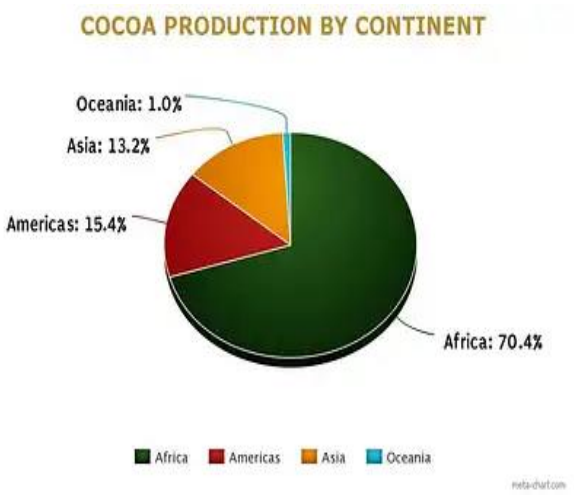

Fig.1: Cocoa production by continent

The next stage in the process is drying, which diminishes acidity levels in the cocoa beans. Beans are then roasted. Roasting helps to control the final flavor of the beans. So, the possible quality parameters are there like imagery evaluation and sensory evaluation. In our scope we have done the Imagery Evaluation [3].

The quality of cocoa beans is determined by its size, shape and texture. The beans are categorized as large beans, medium beans, small beans and rejected beans.

Quality examination of cocoa beans is generally done using manual procedures; i.e., using the visual inspection. Accuracy of human inspection armed with experience depends on individual adaption, choice, mental state etc. Manual inspection of individual cocoa beans is qualitative, tedious and subjective. Hence, manual analysis may not be suitable for routine checks on the quality of commercial cocoa beans. Therefore, a quick and reliable method is desired to classify cocoa beans for quality control. In recent years Machine vision techniques have been emerged into the assessment of the quality control of agricultural and food products. Automation technology backed up by artificial intelligence can be gainfully used to eliminate the limitations of manual inspection. Computer vision [4] which combines image analysis with machine learning [5] and deep learning techniques are implemented to provide automated inspection. Here, images can be analyzed and processed to get useful information to the user. Structural features like size, shape, texture features are extracted from the image. In order to remove the redundant and unimportant features, features are optimized using two feature optimization techniques i.e. Univariate Selection and Feature Importance. The key objective of this study is to determine the possibility of using size, shape and texture features to determine cocoa bean quality. For classification four machine learning algorithms and one deep learning technique are adopted to determine the cocoa bean quality. And a comparison study has been made after testing on the cocoa bean test dataset on the basis of the performances of these four ML algorithms ie. KNN [6], Support Vector Machine [7], Decision tree [8], Random Forest [9] and a deep learning technique CNN. Experimental results are presented accordingly.

The rest part of the manuscript is planned as given below. ‘Proposed Method’ explains the proposed technique. The tentative outcomes on cocoa beans databases are available in ‘Results and Discussion’. Finally, concludes with a ‘Conclusion’.

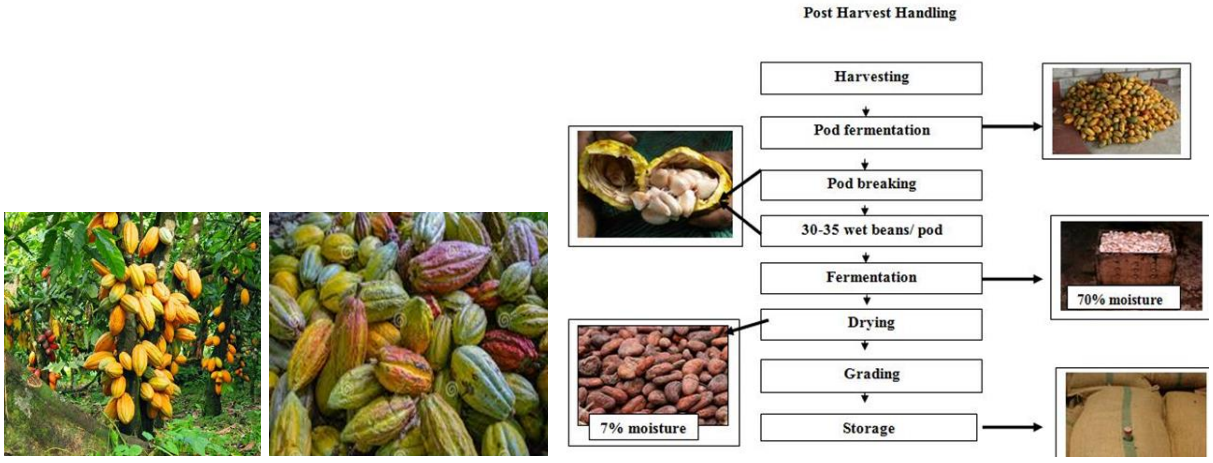

Fig 2 (a) & (b): Cocoa beans process flow for harvesting and post-harvesting

LITERATURE REVIEW

Adhitya, Yudhi, Prakosa, Setya, Köppen, Mario, Leu, nd Jenq-Shiou [] proposed to use the co-occurrence matrix feature of GLCM to extract the features from cocoa bean images and the classification accuracy result shows that it is more reliable than extracting features by CNN. K. Sánchez; J. Bacca; L. Arévalo-Sánchez; H. Arguello; S. Castillo [] have proposed a non-invasive system in their study to obtain hyperspectral images of cocoa beans and the data gathered by processing the images is used to classify each bean based on its fermentation level. The results show that the fermentation level of dried cocoa beans can be estimated by non-invasive hyperspectral image acquisition and image processing techniques. Lestari, U, Kumalasanti, R and Wulandari, Erwin have used 256 x 256-pixel images of cocoa beans is to train an Artificial neural network for classification of cocoa bean quality. The study shows that using Backpropagation method the ANN model predicts with 70% accuracy rate and an error rate of 30%.

PROPOSED METHOD

In this manuscript, our algorithm is based on discrimination of cocoa beans using structural image features. Indian samples are collected from the market and the data collection is done in the form of digital data or images of cocoa beans. Before taking the images, beans are placed on white background. 25 beans per image are preferred. Images of the cocoa beans are captured with the help of a digital image capturing setup. The image capturing setup comprises of a digital colour camera and a controlled illumination system placed inside an enclosed cabinet. The e-COCOA Vision system comprises of image acquisition from an input device to analyse and finally grading cocoa sample based on predefined criteria. Fig.3 describes the system setup for cocoa beans image capturing. A portable image capturing setup has been made using 20 LEDs fitted throughout the roof of the cabinet equidistant from each other. A Logitech C920 webcam is there to capture the image. Colour of the cabinet is made of aluminium sheet and painted with black colour to avoid the unnecessary reflection.

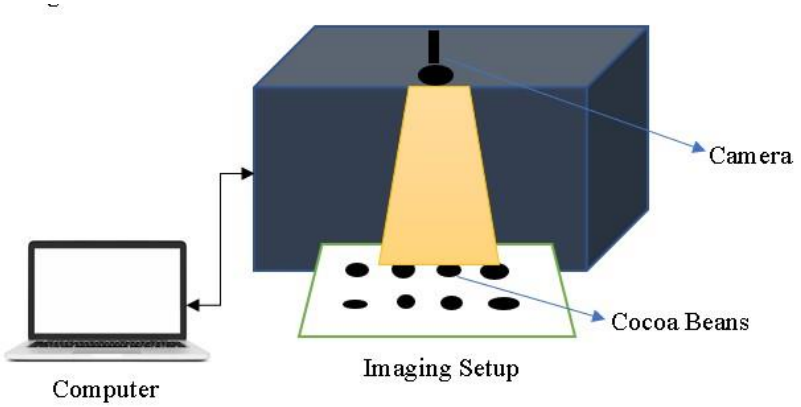

Fig 3: System setup for image capturing

The digital images of cocoa beans comprise of 4 classes of cocoa beans (Fig.4) where 3 classes are of whole beans and are categorized as (1) large bean (2) medium bean (3) small bean and the rest are categorized as (4) rejected beans which are fragmented. Images of 220 beans were taken for experimentation. Among 220 images, 70% were taken for model training and 30% were used for testing. The workflow diagram while working with machine learning models and the CNN model of the system is shown in Fig.5 and Fig.6 respectively.

Fig. 4 cocoa beans on white paper

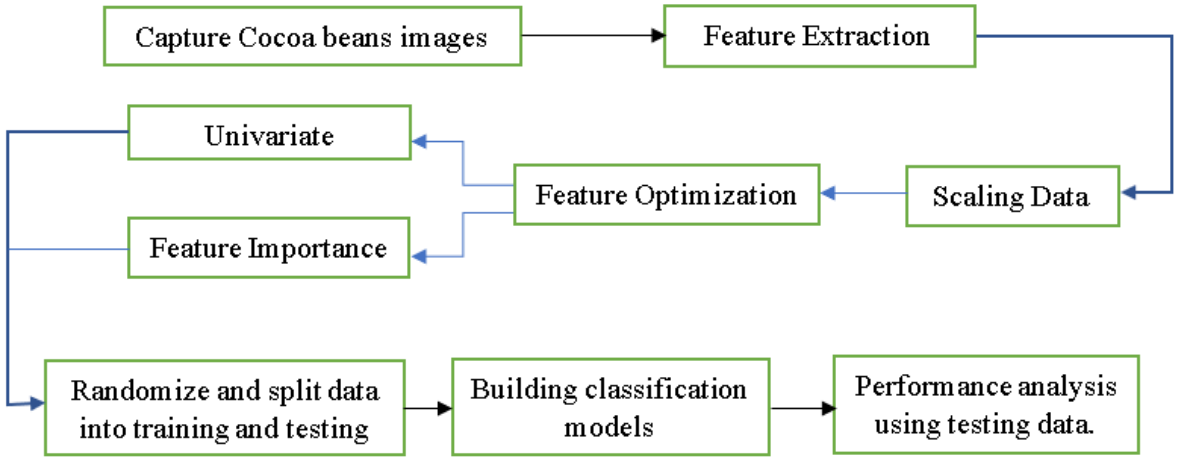

Fig.5 Workflow diagram of the proposed method using ML models

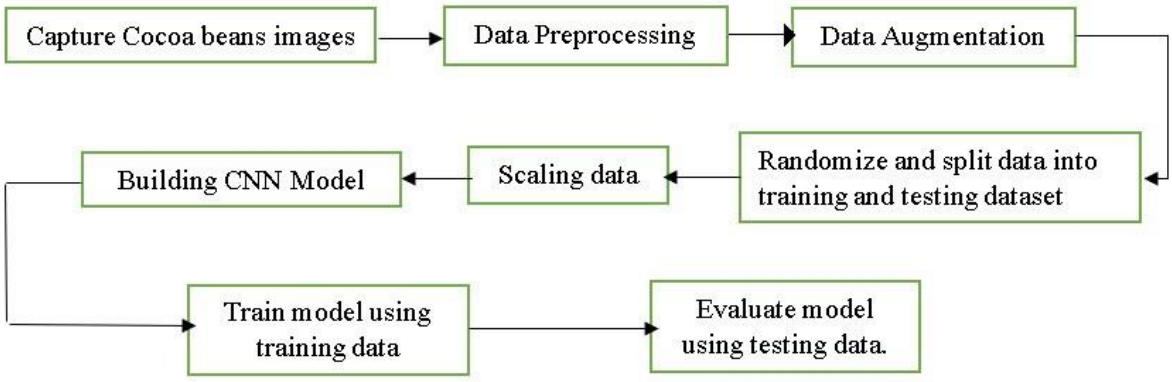

Fig.6 Workflow diagram of the proposed method using CNN model

Data Pre-processing

For Ml models after image data collection data needs to be processed beforehand to extract features by enhance the quality of the image and eliminating the background. For this purpose, the following steps are followed.

- *Gray image conversion:* The RGB images are converted to 8-bit gray scale images. Image analysis using gray scale helps us to eliminate white background.
- *Image Segmentation:* A global thresholding technique using OTSU has been used for image thresholding. Output image after applying the thresholding technique yields with a binary image.
- *Smoothing with Gaussian filter:* Smoothing technique has been applied using a Gaussian smoothing filter with kernel size=3. This helps to eliminate the high frequency noise in the image.
- *Object Identification:* Erosion technique is applied for identifying and removing small particles which are adjacent to the image boundaries. Finally, objects are identified based on the area of the particles in the image.

For the neural network model, the image dataset is prepared by separating the images of different classes in different folders with their respective class names. To enhance the number of images data augmentation is done using 'ImageDataGenerator' class available is Keras library. It also minimizes the chances of overfitting. After augmentation total 900 images comprises of all four classes were generated with rotation range=45, width shift range=0.2, height shift range=0.2 and enabled horizontal flip.

Feature Extraction

To train the ML models a set of 23 image features are extracted from the images which are Perimeter, Convex Hull Perimeter, Max Feret Diameter, Equivalent Ellipse Major Axis, Equivalent Ellipse Minor axis, Equivalent Rectangle Long Side, Equivalent Rectangle Short side, Equivalent Rectangle Diagonal, Hydraulic Radius, area, Convex Hull area, Ratio of Equivalent Ellipse Axis, Ratio of Equivalent Rectangle sides, Elongation Factor, Compactness factor, Heywood circularity factor, and 7 HU Moment features. For CNN model feature extraction is done by the convolutional layers itself.

Feature Optimization

Generally, for a machine learning model, all the independent features in the dataset do not impact the dependent feature in same measure. Some features may have very less impact. Feature optimization [10] is done to eliminate the redundant features to improve the machine learning models. It reduces the training time and complexity of the models without compromising the accuracy.

In this study, two feature optimization techniques have been used Univariate analysis and feature iportance.

For Univariate selection [11], Scikitlearn provides the SelectKBest class that works with a suite of different statistical tests and based on the scores of these tests the correlation between each feature with the target label is measured. The statistical tests available in this class for classification are 'f_classif' for ANOVA F-value [12] between features, 'mutual_info_classif' for mutual information [13] for a discrete target label and 'chi2' for Chi-squared stats [14] of non-negative features. Keeping in mind that the independent features here are continuous and dependent features are categorical the mutual information test is selected. Fig.8 shows each feature and it its corresponding univariate selection score.

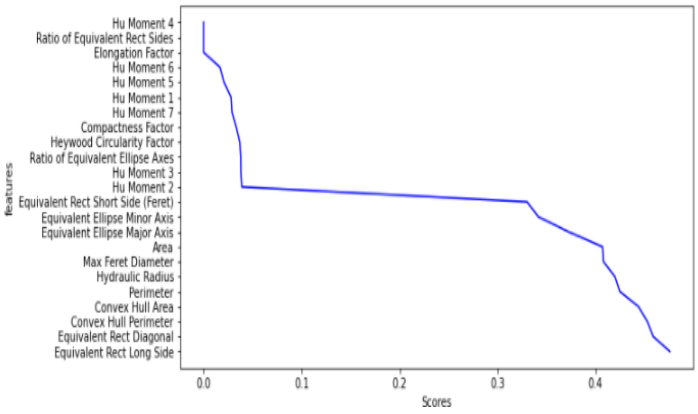

Fig. 7 Line chart for univariate feature selection

	features	Score
0	Perimeter	0.424857
1	Convex Hull Perimeter	0.452859
2	Max Feret Diameter	0.408114
3	Equivalent Ellipse Major Axis	0.372409
4	Equivalent Ellipse Minor Axis	0.341688
5	Equivalent Rect Long Side	0.475832
6	Equivalent Rect Short Side (Feret)	0.330108
7	Equivalent Rect Diagonal	0.458822
8	Hydraulic Radius	0.419752
9	Area	0.407097
10	Convex Hull Area	0.443806
11	Ratio of Equivalent Ellipse Axes	0.037955
12	Ratio of Equivalent Rect Sides	0.000000
13	Elongation Factor	0.000000
14	Compactness Factor	0.033345
15	Heywood Circularity Factor	0.036971
16	Hu Moment 1	0.028022
17	Hu Moment 2	0.039065
18	Hu Moment 3	0.038141
19	Hu Moment 4	0.000000
20	Hu Moment 5	0.020831
21	Hu Moment 6	0.016707
22	Hu Moment 7	0.028995

Fig. 8 Feature scores for univariate selection

Feature importance [15] refers to a class of methods that provides scores for each independent feature in a prediction model based on the importance of the feature in making accurate predictions. The higher the score the more relevant the feature towards the target label. In this study 'ExtraTreeClassifier' method available in scikitlearn library is used for measuring feature importance. It implements a

number of randomized decision trees as estimators using different subsets of main dataset to calculate the importance of each feature and select the top relevant features. Fig.10 shows each feature and it its corresponding feature importance score.

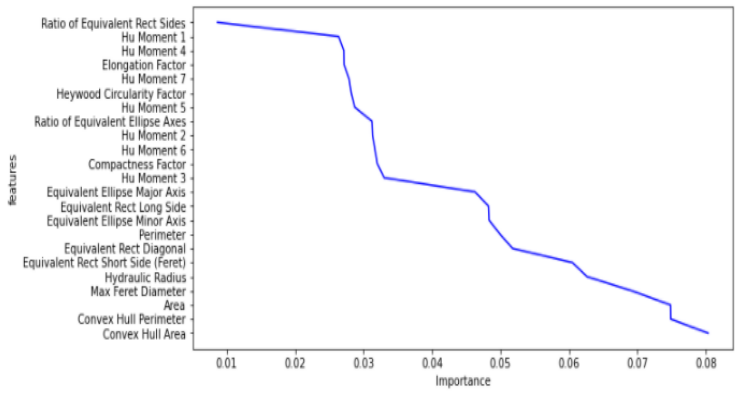

Fig. 9 Line chart of Feature Importance

	Features	Importance
0	Perimeter	0.049933
1	Convex Hull Perimeter	0.074856
2	Max Feret Diameter	0.069188
3	Equivalent Ellipse Major Axis	0.046265
4	Equivalent Ellipse Minor Axis	0.048323
5	Equivalent Rect Long Side	0.048236
6	Equivalent Rect Short Side (Feret)	0.060495
7	Equivalent Rect Diagonal	0.051756
8	Hydraulic Radius	0.062639
9	Area	0.074827
10	Convex Hull Area	0.080332
11	Ratio of Equivalent Ellipse Axes	0.031214
12	Ratio of Equivalent Rect Sides	0.008642
13	Elongation Factor	0.027147
14	Compactness Factor	0.031998
15	Heywood Circularity Factor	0.028187
16	Hu Moment 1	0.026344
17	Hu Moment 2	0.031306
18	Hu Moment 3	0.032991
19	Hu Moment 4	0.027105
20	Hu Moment 5	0.028685
21	Hu Moment 6	0.031656
22	Hu Moment 7	0.027874

Fig. 10 Feature importance score

After applying univariate selection and feature importance optimization techniques the relation between each of the individual features and the target dependent feature can be visualized by the line charts in Fig. 7 and Fig. 9. In both cases after the first eleven features with largest scores there is a sharp change in the curve that suggests that these eleven features have the most impact and the rest of the features are comparatively less relevant in terms of predicting the target class labels. Therefore, these eleven most relevant features are selected to build the classification models.

Feature Scaling

Algorithms like KNN and SVM are affected by the range of features because they use the distance between the samples to determine the similarity between them. For neural network like CNN the gradient descent converge much faster with scaled data. For these reasons the independent feature set is rescaled ranging them

in between 0 and 1 by Min-Max Normalization before training KNN and SVM models. Here, the formula denoted as follows:

$$X' = (X - Xmin) / (Xmax - Xmin) \quad (1)$$

For the Convolutional Neural Network, the RGB channel values, representing each image, are divided by 255 to scale them in between 0 and 1.
On the other hand, tree based ML classifiers are not sensitive towards the scale of the features, so this classification models can perform well without rescaling the feature set.

Data Analysis

Based on the classification problem in terms of algorithms three different types of supervised [16] approach for classification has been taken- (1) Distance based classifier, (2) Tree based classifiers, (3) Neural-net-based classifier. The programming language used to develop the classification model is python 3.9.7 and the necessary python libraries which were imported are matplotlib, pandas, numpy, seaborn, sklearn, tensorflow, OpenCV, PIL, os, pathlib, pydotplus and six.

RESULTS AND DISCUSSION

The performance of the proposed method has been tested on the cocoa beans testing dataset. The evaluation metrics used for evaluating the models is Accuracy score. It is the sum of True Negative and True Positive divided by the sum of True Negative, True Positive, False Positive and False Negative. Here, formula defined as follows:

$$Accuracy\ score = (TP + TN) / (TP + TN + FP + FN) \quad (2)$$

$$F\beta\ score = (1 + \beta\text{^}2) \times (Precision \times Recall) / (\beta\text{^}2 \times Precision + Recall) \quad (3)$$

Precision is the ratio of correctly predicted positive observation to the total predicted positive observation. Recall is the ratio of correctly predicted positive observation to the all observations in actual positive class. F1 score is the Fβ score where β=1.

$$Precision = TP / (TP + FP) \quad (4)$$

$$Recall = TP / (TP + FN) \quad (5)$$

Using the ‘accuracy_score’ and ‘f1_score’ method available in Scikitlearn library the accuracy score and F1 score is obtained for all the four classification models.

For Distance based algorithms, two traditional classification algorithm K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) are used. KNN classifies cases based on the similarity and this similarity is measured by a distance matrix such as Euclidean Distance [17], Manhattan Distance [18], Minkowski Distance or Hamming Distance [19]. Cases those are near to each other are said to be 'Neighbours'. While predicting classes for unknown data point the most popular class label or class label with the majority value from its neighbours is considered as the class label for the unknown data point. On the other hand, SVM is efficient in handling the non-linearity of dataset by transforming the data to a higher dimensional space and then classification is performed by finding the best hyperplane that differentiates the classes very well. Although SVM is also memory efficient as it uses a subset of the training data in the decision function but the training time is higher than KNN as KNN does not derive any discriminative function from the training data, it stores the training dataset and learns from it only while making real time predictions but SVM learns during training period.

The KNN classification model is trained with the training dataset having K value in a range from 1 to 20. And using the testing dataset the best minimum value for K is determined to be 10 for which the model predicts with maximum accuracy score 0.76 and F1 score 0.71. Fig.11 plots the K-values and corresponding accuracy score using K value for KNN model. In Table-2 the accuracy scores for each K value from1 to 20 are shown.

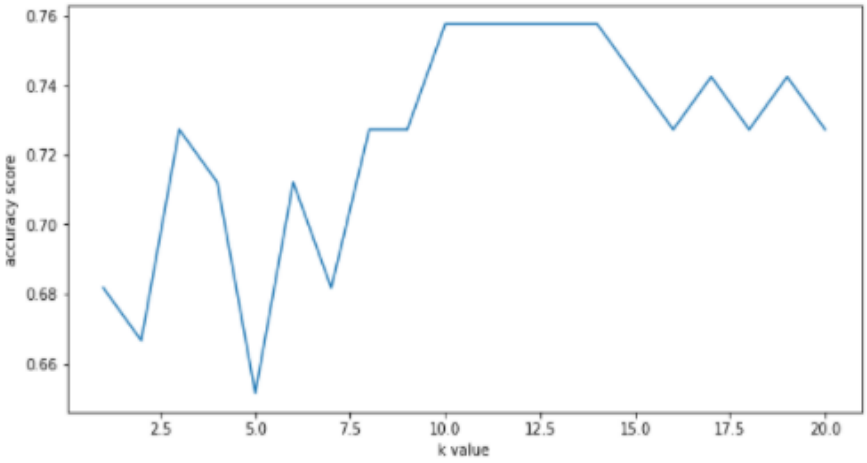

Fig. 11 Value of K Vs Accuracy Score

K Value	Accuracy Score
1	0.681818
2	0.666667
3	0.727273
4	0.712121
5	0.651515
6	0.712121
7	0.681818
8	0.727273
9	0.727273
10	0.757576
11	0.757576
12	0.757576
13	0.757576
14	0.757576
15	0.742424
16	0.727273
17	0.742424
18	0.727273
19	0.742424
20	0.727273

Table-2: Value of K and corresponding accuracy scores

While Using the Polynomial Kernel function the SVM model gives maximum accuracy score 0.73 and F1 score 0.71. Accuracy score and F1 Score for different kernel functions are mentioned in Table 3.

SVM Kernel Function	**Accuracy Score**	**F1 Score**
Linear	0.68	0.61
RBF	0.73	0.68
Sigmoid	0.53	0.38
Polynomial	0.72	0.71

Table-3: Performance evaluation of different kernel function.

For Tree based algorithms, two popular algorithms Decision Tree Classifier and Random Forest classifier are used. Decision Tree classifier is a tree structured classifier where the branches represent the decision rules, the internal nodes represent the features of the sample dataset and the leaf nodes represents the final output or class labels. It uses Recursive Partitioning [20] to split the training records into segments by minimizing the impurity at each step. But whenever decision tree is built to its complete depth it comes with low bias that means the model gets overfitted to the training dataset and high variance which suggests that the model is prone to give large amount of errors while working with new test data. In Random Forest Classifier instead of using a single decision tree multiple decision trees with high variance generated from the subsets of the main dataset are considered and by combining the trees with respect to a majority vote the high variance gets converted into low variance. One more thing is if we change some data or add some new data to our model it would not affect much because the changes will be distributed to all the decision trees while we are doing random sampling of the rows and columns.

For both of these Tree Based algorithms the criterion function selected is Entropy [21] for selecting the root node or internal nodes at different level of the decision trees. The goal is to find the tree with smallest entropy in its nodes. So for Decision Tree model criterion function selected for splitting is ‘entropy’ with ‘best’ splitter strategy and maximum depth for the decision tree is determined to be 4 after trying a range of values from 1 to 10 to achieve the maximum accuracy score 0.74 and F1 score 0.73. The decision tree obtained from Decision Tree Classifier is shown in Fig.12. While the random forest classifier is trained with same criterion function for 150 decision trees with maximum depth 4 to achieve the maximum accuracy score 0.74 and F1 score 0.71.

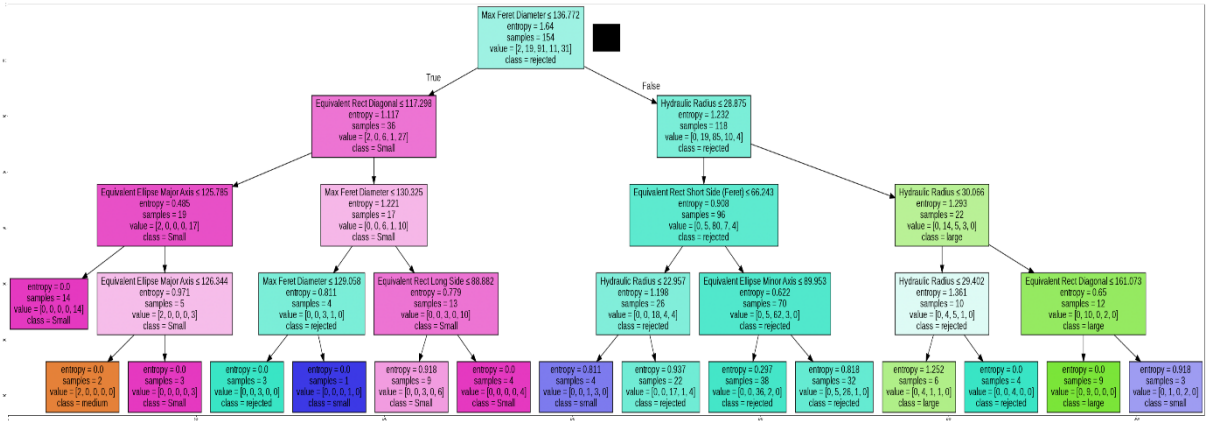

Fig. 12 Decision tree obtained from Decision Tree Classifier

For neural-net-based algorithms whenever the inputs are of image format or video format Convolutional Neural Network (CNN) [22] is the most popular deep learning algorithm which consists of a number of convolutional layers. In each layer a linear operation known as convolution is performed between the image matrix and filter metrices or kernel such as vertical edge filter, horizontal edge filter etc. These filters are responsible for extracting information from images. After each convolution operation the image size is generally reduced. To avoid this loss of data padding with zeros evenly to the left/right or up/down of the input is done with strides 1 to get the output as same size of the input after each convolutional layer. Then each convolutional layer is followed by a max-pooling [23] layer consist of 2D filters to extract higher intensity pixels lying within the region covered by the filter. As the convolutional layers are stacked sequentially followed by a max pooling layer the output of a max-pooling layer becomes a feature map containing the most prominent features from the previous feature map obtained from previous layer. This enables the model to extract features from cocoa beans like pixel patterns for fragmented beans. In this study the 'Sequential' class available in Keras library is used to build the CNN model with five convolutional layers consisting increasing number of filters of size (3,3) as the model goes deeper. Number of filters in these five consecutive layers are 16, 32, 64, 128 and 256 respectively. Activation function used at each convolutional layer is ReLU [24] to prevent the exponential growth in the computation required to operate the neural network**.** It also prevents the chance of vanishing gradient or exploding gradient that lies while using sigmoid activation function. After that a flatten layer is placed to convert the two-dimensional resultant metrices from pooled feature map to a single continuous one-dimensional vector for transition from the convolutional layer to fully connected layer. After that two fully connected dense layers are placed. The first one consists of 512 neurons along with ReLU as activation function and the second dense layer which is also the output layer consists of 4 neurons representing the four output classes for our model. Activation function used at the output layer is softmax [25] for multinomial probability distribution

of the output for four classes. Two gradient descent-based optimization algorithms- Adaptive Momentum (Adam) [26] and Root Mean Square Propagation (RMS Prop) are tested for optimizing the model where each time with different hyperparameters Adam optimizer results with maximum accuracy. The training images and class labels are fitted to the model with epoch value set to 50. These hyperparameters are measured after training the model with different sets of hyperparameters and evaluating the model each time using Accuracy metrics and Sparse Categorical Cross Entropy loss function. The maximum accuracy score achieved by the CNN model is 0.85 while the loss is 0.36. Fig. 13 describes the model summary. How the layers are sequentially stacked along with the size of the output metrices after each layer is shown in Fig. 14.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 200, 200, 16)      448
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 100, 100, 16)      0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 100, 100, 32)      4640
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 50, 50, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 50, 50, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 25, 25, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 25, 25, 128)       73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 12, 12, 128)       0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 12, 12, 256)       295168
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 6, 6, 256)         0
_________________________________________________________________
flatten (Flatten)            (None, 9216)              0
_________________________________________________________________
dense (Dense)                (None, 512)               4719104
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 2052
=================================================================
Total params: 5,113,764
Trainable params: 5,113,764
Non-trainable params: 0
_________________________________________________________________
```

Fig. 13 CNN model summary

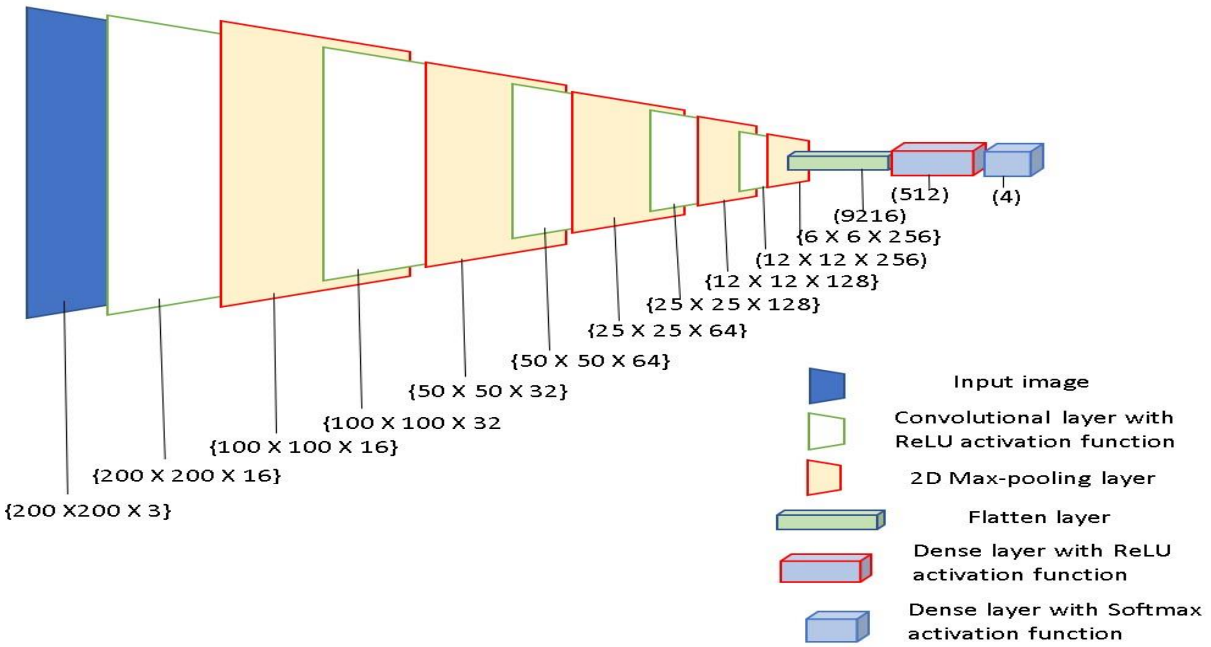

Fig. 14 CNN architecture

Classification Reports

The classification reports generated for the five classification models are displayed in Fig. 15.

KNN Classification report:

	precision	recall	f1-score	support
large	0.60	0.67	0.63	9
medium	0.75	0.94	0.84	35
rejected	0.00	0.00	0.00	7
small	0.92	0.73	0.81	15
accuracy			0.76	66
macro avg	0.57	0.59	0.57	66
weighted avg	0.69	0.76	0.71	66

SVM Classification report:

	precision	recall	f1-score	support
large	0.71	0.56	0.63	9
medium	0.71	0.86	0.78	35
rejected	1.00	0.29	0.44	7
small	0.73	0.73	0.73	15
accuracy			0.73	66
macro avg	0.79	0.61	0.65	66
weighted avg	0.75	0.73	0.71	66

Decision tree Classification report:

	precision	recall	f1-score	support
large	1.00	0.56	0.71	9
medium	0.73	0.94	0.83	35
rejected	0.40	0.29	0.33	7
small	0.82	0.60	0.69	15
accuracy			0.74	66
macro avg	0.74	0.60	0.64	66
weighted avg	0.75	0.74	0.73	66

Random Forest Classification report:

	precision	recall	f1-score	support
large	0.67	0.67	0.67	9
medium	0.71	0.97	0.82	35
rejected	1.00	0.14	0.25	7
small	1.00	0.53	0.70	15
accuracy			0.74	66
macro avg	0.84	0.58	0.61	66
weighted avg	0.80	0.74	0.71	66

CNN Classification report:

	precision	recall	f1-score	support
0	0.93	0.85	0.89	78
1	0.75	0.82	0.78	73
2	0.76	0.81	0.78	74
3	0.97	0.91	0.94	75
accuracy			0.85	300
macro avg	0.85	0.85	0.85	300
weighted avg	0.85	0.85	0.85	300

Fig. 15 Classification reports

As the dataset is not perfectly balanced, at the end the classification models are evaluated using stratified K-fold cross validation [27] with 10 folds. It ensures the proportion of target features of different classes is same across the original data, training data and testing data. The performance evaluation of the five algorithms using Stratified K-fold cross validation is mentioned in Table 4.

Fold	**KNN** Cross Validation Scores	**SVM** Cross Validation Scores	**DT** Cross Validation Scores	**RF** Cross Validation Scores	**CNN** Cross Validation Scores
1	0.7272	0.7272	0.6818	0.7727	0.7083
2	0.5909	0.6818	0.7272	0.6818	0.9916
3	0.7272	0.7727	0.7272	0.7727	0.8833
4	0.6818	0.7727	0.6818	0.7272	0.7833
5	0.6363	0.5909	0.6363	0.6818	0.7249
6	0.7727	0.7727	0.7727	0.8181	0.7583
7	0.7727	0.7727	0.7272	0.7727	0.7666
8	0.8636	0.8636	0.8181	0.8181	0.7916
9	0.5909	0.5909	0.6818	0.6363	0.9666
10	0.7272	0.7272	0.7727	0.7727	0.9166
Max	0.86	0.86	0.82	0.82	0.97
Min	0.59	0.59	0.64	0.64	0.71
Mean	0.71	0.73	0.72	0.75	0.83

Table-4: K fold cross validation result with 10 folds

CONCLUSION

By training and testing the classification models using structural feature set with four traditional classification algorithms KNN, SVM, Decision Tree and Random Forest and one deep learning algorithm Convolutional Neural Network it can be concluded that the resultant accuracy scores and F1 scores achieved by the four ML models are in range between 0.71 to 0.75 and 0.68 to 0.73 respectively. While the accuracy range for CNN is slightly better that is 0.78 to 0.86 having loss value in a range between 0.59 to 0.88. The K-fold cross validation results show that among the ML algorithms the Random Forest Classifier provides highest mean accuracy score of 0.75 while overall the CNN model predicts with highest mean accuracy score 0.83.

REFERENCES

[1] L. De Vuyst,S. Weckx.(2016). Review Article: The cocoa bean fermentation process: from ecosystem analysis to starter culture development, *Journal of Applied Microbiology ISSN 1364-5072*,

[2] K. Sánchez; J. Bacca; L. Arévalo-Sánchez; H. Arguello; S. Castillo, "Classification of Cocoa Beans Based on their Level of Fermentation using Spectral Information", *TecnoLógicas*, vol. 24, nro. 50, e1654, 2021.

[3] C. Relf, (2003). Image Acquisition and Processing with LabVIEW. 10.1201/9780203487303.

[4] M. M. Oliveira, B. V. Cerqueira, S. B. Douglas, F. Barbin, Classification of fermented cocoa beans (cut test) using computer vision, Journal of Food Composition and Analysis, Volume 97,2021, 103771.

[5] A. I. Khan, S. Al-Habsi, Machine Learning in Computer Vision, Procedia Computer Science,Volume 167,2020,Pages 1444-1451.

[6] G. Guo, H. Wang, D. Bell, Y. Bi,. (2004). KNN Model-Based Approach in Classification.

[7] T. Evgeniou, M. Pontil, (2001). Support Vector Machines: Theory and Applications. 2049. 249-257.

[8] H. Patel, P. Prajapati, (2018). Study and Analysis of Decision Tree Based Classification Algorithms. International Journal of Computer Sciences and Engineering. 6. 74-78.

[9] J. Ali, R. Khan, N. Ahmad, I. Maqsood, (2012). Random Forests and Decision Trees. International Journal of Computer Science Issues(IJCSI). 9.

[10] L. Abualigah, A. Aldulaimi, M. Al Shinwan, A. Khasawneh, H. Alabool, M. Diabat, M Shehab, (2020). Optimization Algorithms to Solve Feature Selection Problem: A Review. International Journal of Science and Applied Information Technology. 8. 10.30534/.

[11] R. Subho, Md. Chowdhury, D. Chaki, S. Islam, Md. Rahman, (2019). A Univariate Feature Selection Approach for Finding Key Factors of Restaurant Business. 605-610.

[12] E. Ostertagova, O. Ostertag, (2013). Methodology and Application of One-way ANOVA. American Journal of Mechanical Engineering. 1. 256-261. 10.12691/ajme-1-7-21.

[13] N. Hoque, D.K. Bhattacharyya, J.K. Kalita,MIFS-ND: A mutual information-based feature selection method,Expert Systems with Applications,Volume 41, Issue 14, 2014,Pages 6371-6385.

[14] R. Singhal, R. Richa, R. Rakesh. (2015). Chi-square test and its application in hypothesis testing. Journal of the Practice of Cardiovascular Sciences. 1. 10.4103/2395-5414.157577.

[15] Saarela, M., Jauhiainen, S. Comparison of feature importance measures as explanations for classification models. *SN Appl. Sci.* **3,** 272 (2021).

[16] J E T. Akinsola, (2017). Supervised Machine Learning Algorithms: Classification and Comparison. International Journal of Computer Trends and Technology (IJCTT). 48. 128 - 138.

[17] I. Dokmanic, R. Parhizkar, J. Ranieri and M. Vetterli, "Euclidean Distance Matrices: Essential theory, algorithms, and applications," in IEEE Signal Processing Magazine, vol. 32, no. 6, pp. 12-30, Nov. 2015.

[18] M. D. Malkauthekar, "Analysis of euclidean distance and Manhattan Distance measure in face recognition," Third International Conference on Computational Intelligence and Information Technology (CIIT 2013), 2013, pp. 503-507.

[19] B.Abraham, K Vladimir, R. Timo. (2002). Generalized Hamming Distance. Information Retrieval. 5. 10.

[20] C. Strobl, J. Malley, G. Tutz, (2009). An Introduction to Recursive Partitioning: Rationale, Application, and Characteristics of Classification and Regression Trees, Bagging, and Random Forests. Psychological methods. 14. 323-48. 10.1037/a0016973.

[21] Du, Ming & Wang, Shu & Gong, Gu. (2011). Research on Decision Tree Algorithm Based on Information Entropy. Advanced Materials Research. 267. 732-737. 10.4028/www.scientific.net/AMR.267.732.

[22] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

[23] ***Naila Murray, Florent Perronnin***; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 2473-2480

[24] Agarap, Abien Fred. (2018). Deep Learning using Rectified Linear Units (ReLU).

[25] I. Kouretas and V. Paliouras, "Simplified Hardware Implementation of the Softmax Activation Function," 2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST), 2019, pp. 1-4, doi: 10.1109/MOCAST.2019.8741677.

[26] Z. Zhang, "Improved Adam Optimizer for Deep Neural Networks," 2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS), 2018, pp. 1-2, doi: 10.1109/IWQoS.2018.8624183.

[27] Berrar, Daniel. (2018). Cross-Validation. 10.1016/B978-0-12-809633-8.20349-X.