Cocoa-Net: Performance Analysis on Classification of Cocoa Beans Using Structural Image Feature

[1]Chandrajit Pal, [2]Samikshan Das, [3]Amitava Akuli, [4]Sudip Kumar Adhikari, [5]Aniruddha Dey*
[1]Department of Electrical Engineering, IIT, Hyderabad, India
[2]Capgemini, India
[3]Centre for Development of Advanced Computing, Kolkata, India.
[4]Department of Computer Science & Engineering, CGEC, Cooch Behar, India
[5]Department of Computer Science & Engineering, MSIT, Kolkata, India.
E-mail: palchandrajit@gmail.com, samikshandas5@gmail.com, amitava.akuli@gmail.com, sudipadhikari@ieee.org, anidey007@gmail.com
*Corresponding author

Keywords: KNN, decision tree, SVM, random Forest, Cocoa-Net, cocoa beans.

Received: July 29, 2024

Abstract. The process of cocoa hybridization produces new types that have unique chemical properties impacting the manufacturing of chocolate yet are resistant to a number of plant illnesses. Image analysis is a valuable tool for visually differentiating cocoa beans where deep neural networks (DNN) play a pivotal role in implementing them. In this manuscript, we compare machine learning and deep learning models because it takes into consideration multiple images covering a wide range of agricultural products. Specifically, we extract features from images using a series of image processing techniques, following which we use both traditional machine learning methods (KNN, Decision tree, SVM, and Random Forest) and Convolutional Neural Networks (proposed Cocoa-Net and RESNET 50) to classify the cocoa beans into four categories: large, medium, small, and rejected. Since each methodology offers strong classification performance and has potential for use in the classification of food, they were all chosen. To test these methods, a dataset including 200 samples of fragmented images was utilized. Studies that compare various similar approaches are also carried out. Two optimization techniques: Univariate Selection and Feature Importance have been leveraged to optimize the retrieved features before the learning models are trained. The Adam optimizer is used to optimize the proposed Cocoa-Net model. K-fold cross validation is utilized to assess trained models, and mean cross validation scores are then computed for performance analysis. The empirical result shows that, the proposed Cocoa-Net model predicts with the highest classification accuracy score of 0.85.

Povzetek: V raziskavi so razvili Cocoa-Net model za klasifikacijo kakavovih zrn, ki je dosegel najvišjo klasifikacijsko točnost 0,85. Uvedene tehnike optimizacije značilk so dodatno izboljšale zmogljivost modela v primerjavi s tradicionalnimi metodami strojnega učenja.

1 Introduction

The cocoa bean seeds are found in the fruit pods of the Theobroma cocoa tree. Almost two thirds of the cocoa produced worldwide is grown in West Africa [1]. Ghana produces more than twenty percent of the world's total, making it the second largest producer in the world. To obtain the unique flavour and aroma of cocoa, raw cocoa must be fermented, dried, and roasted because it has an unpleasant, astringent flavour [2]. Following the picking of the cocoa beans, the cocoa pods are opened, allowing a variety of bacteria to naturally colonise the pulp surrounding the beans. These microorganisms convert the pulp's sugars into lactic acid and ethanol, which are then used to produce chocolate. A portion of the ethanol is converted into acetic acid by the acetic acid-producing bacteria through an exothermic process [3]. The mass is heated to about 50°C by the ethanol and acetic acid that are introduced into the test. The bean's germs and cell walls are destroyed by this heat. Start the procedures that yield beans with a high degree of fermentation [4]. Ghana is divided into two agro-ecological zones: the southern forest and the northern savannah. The savannah zone includes Sudan, Guinea, and the coastal areas. The semi deciduous, transitional, and rainforest zones are all found in the southern forest region [5]. The models were created using two regression techniques: partial least square regression (PLSR) and principal component regression (PCR) [6]. Ghosh et al proposed entropy-based feature extraction technique which can preserve the core data while reducing the volume of data being processed [7]. Feature extraction has also been applied in the field of smart farming application for Cocoa bean digital image classification prediction [8]. Nazir et al. investigates the procedure of the deep convolutional neural network for mispronunciation finding of Arabic phonemes [9]. In image processing, textural pattern is a crucial component. In order to incorporate the co-

occurrence matrix in texture analysis employed the sum and difference histogram technique to manipulate histograms for texture classification [10]. The distribution of the grayscale's spatial value is one of the factors that define a texture, so one of the techniques recommended in the numerous machine vision studies is the application of the Gaussian function [11]. By directly extracting significant features from the data in a multi-level abstract, deep learning (DL) increases the predictive power of machine learning [12]. A variety of agricultural products may find promising solutions acknowledgments to computer vision and machine learning (ML)'s predictive capabilities [13]. Pereira et al predicting the ripening of papaya fruit using digital imaging and random forests [14]. Tian et al. discussed the use of computer vision technology in agricultural automation [15]. In order to boost agricultural productivity, particularly in terms of quality and competitiveness, technology utilisation is required especially in terms of superiority and effectiveness [16]. The accessibility of technological advancements like deep learning and machine learning is also essential for enhancing farmer welfare and piquing the interest of the younger generation in developing diverse derivative business opportunities [17–19]. An example of information technology progress is smart farming which gives farmers the ability to exercise more reliable control. Srikanth *et al.* implemented ANN with 35 input nodes for the purpose of classifying four different classes of cocoa beans: i.e. whole, broken, fractions, and skin-damaged beans [19]. Numerous investigators have carried out the process of crop classification and grading systems identification [20, 21]. In order to meet quality control requirements, Dey *et al.* used image processing technology and SVM classifier [23, 24]. In supermarkets, fruit was categorized based on species class and price using deep learning techniques. In order to classify fruit in supermarkets according to species class and price applied deep learning techniques [25]. Furthermore, automation in agriculture has also made use of traditional machine learning techniques [26]. To help farmers measure the rate of fermentation and guarantee the quality of their cocoa beans, a study on the subject was carried out by Tan et al [27]. In order to provide the farmer with a higher-quality grade of beans and a more suitable payment, the processing factory can distinguish between good and regular beans during the classification process [28]. Real-time image processing can quickly reduce the amount of time it takes to fused image, and it can be processed for additional analysis [29]. Image fusion has a broad coverage area in conjunction with the adoption of machine learning to combine hybrid data [30]. Summarization of very recent machine learning based cocoa classification techniques has been presented in Table I.

Table 1: Recent study (2021-2022) of summarized methods

Authors, Year, Reference	Summery
Das et al., 2022 [31]	Machine vision approach for morphologically categorizing cocoa beans
Tercan and Meisen, 2022 [32]	A comprehensive review of predictive quality in manufacturing using machine learning and deep learning
Kim et al., 2021 [33]	Kim et al. present a deep learning-based framework for product quality inspection.
Lopes et al., 2021 [34]	To classify cocoa beans into different varieties, Lopes et al. compare two computer vision systems: a traditional Computer Vision System (CVS) and a Deep Computer Vision System (DCVS).
Anggraini et al. 2021 [35]	To differentiate between fermented and unfermented cocoa, machine learning model to make this determination.
Abu et al. 2021 [36]	Identifying cocoa plantations in Ghana and Cote d'Ivoire and the effects they have on protected areas
Oliveira et al. 2021 [37]	Quick and accurate classification of cocoa beans into four fermentation categories was achieved through the use of computer vision and Random Forest.

A high-quality control measure is the effectiveness attained when applying computer vision techniques to automation [38]. To train machine learning models for classification based on D-S theory, features are extracted as part of image processing [39, 40].

Images can be processed and evaluated using the proposed methods to provide the user with helpful information. The image is processed to extract structural features, such as size, shape, and texture. Features are optimized using two feature optimization techniques, namely Univariate Selection and Feature Importance, to eliminate the unnecessary and redundant features.

The main contributions of the proposed work are:

- Database Creation of Cocoa Beans image.
- This study's primary goal is to ascertain whether characteristics of cocoa beans' size, shape, and texture can be used to assess their quality.
- Four machine learning algorithms are used for classification in order to assess the quality of the cocoa bean. Based on the results of testing four

traditional classifiers (KNN, Decision Tree, SVM, Random Forest) and Convolutional Neural Networks (proposed Cocoa-Net and RESNET 50) on the cocoa bean test dataset, a comparative analysis has been conducted. The suggested Cocoa-Net model and ResNet50 calculates with the overall mean accuracy score of 0.85, 0.84 respectively

The remaining portion of the manuscript is organised as follows. Section II delineates the materials and methods. Proposed CocoaNet architecture describes in Section III. The tentative results for cocoa bean databases can be found in Section IV. Finally, conclusion is included in Section V.

2 Materials and methods

Our algorithm in this manuscript uses structural image features to discriminate between cocoa beans. The market is the source of Indian samples, and digital data or photos of cocoa beans are used for data collection. Beans are positioned on a white background prior to photo capture. For best results, use 25 beans per image [31]. Using a digital image capture setup, pictures of the cocoa beans are taken. A digital colour camera and a controlled lighting system are housed inside a closed cabinet to form the image capture setup. The e-Cocoa Vision system is designed to grade cocoa samples according to predetermined criteria after acquiring images from an input device for analysis [31].

The system configuration for taking pictures of cocoa beans is shown in Figure 1. A transportable system for taking pictures has been created by placing twenty LEDs evenly spaced across the cabinet's roof. There is a Logitech C920 webcam to take the picture [31]. The cabinet is painted black to prevent unwanted reflections and is constructed of aluminium sheets.

For the study, digital photos of the cocoa beans were acquired from South Sulawesi, Indonesia. Based on [36], the sampling procedure was used. The following categories applied to these samples of cocoa beans: (i) Whole beans are defined as cocoa beans that have a whole seed skin covering every part of the bean and do not show any fractures; (ii) broken beans are defined as cocoa beans that have a missing part that is half (1/2) or less than the full bean; (iii) beans fractions are defined as cocoa beans that are less than half (1/2) of the full bean; (iv) skin-damaged beans are defined as beans that have a missing bean shell that is half or lower size than the full bean; (v) fermented beans, a type of cocoa bean that is the end result of the curing process and is cleaned or left unwashed before being dried; (vi) unfermented beans, a type of cocoa bean in which half or more of the sliced greyish chips' surface is visible, while the surface is dirty white; (vii) moldy beans, a type of cocoa bean that has mound inside of it, and when the bean is split exposed, the fungus can visible with the naked eye.

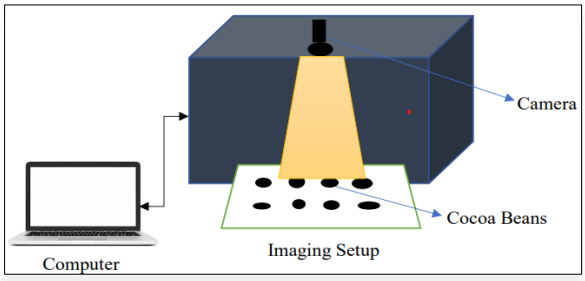

Figure 1: System structure for Cocoa Beans image capturing [31]

The ultimate objective of gathering these digital photos of cocoa beans at the factory was to lessen the amount of classification work that needs to be done there. The acquisition of the cocoa bean image was made possible by a small digital camera [31], as represented in Figure 2.

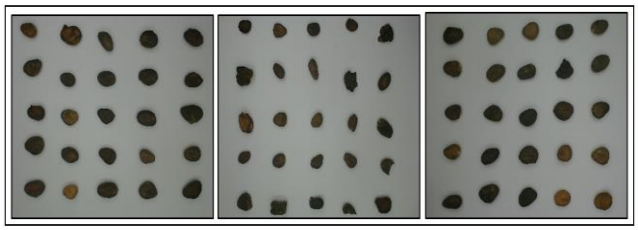

Figure 2: Cocoa beans on white paper [31]

Four classes of cocoa beans are included in the digital images; three classes consist of whole beans and are classified as (i) large bean, (ii) medium bean, and (iii) small bean. The remaining beans are classified as (iv) rejected beans that are fragmented. For the experiment, pictures of 220 beans were captured of the 220 images, 30% were used for testing and 70% were taken for model training [31]. Fig 3 displays the workflow diagram for the system using proposed Cocoa-Net model and machine learning models.

2.1 Data Pre-processing

After data collection, data processing is required to improve image quality by removing background. The actions listed below are taken.

- *Gray image conversion:* We are utilising an RGB image with 24 bits. The RGB image has been converted to an 8-bit grayscale image. Grayscale image analysis aids in the removal of the white background [31].
- *Image Segmentation:* For image thresholding, a global thresholding method utilising OTSU has been applied. After using the thresholding technique, the output image produces a binary image [31].
- *Smoothing with Gaussian filter:* A Gaussian smoothing filter with a kernel size of three has been used to apply the smoothing technique. This aids in removing the image's high frequency noise [31].
- *Object Identification:* The *erosion* method is used to locate and eliminate tiny particles that are close to the

boundaries of the image. Lastly, the area of the particles in the image is used to identify the objects [31].

- *Particle Analysis:* Finally, a set of 23 pertinent image features are extracted from the images using particle analysis by selecting all pixel measurements [31].

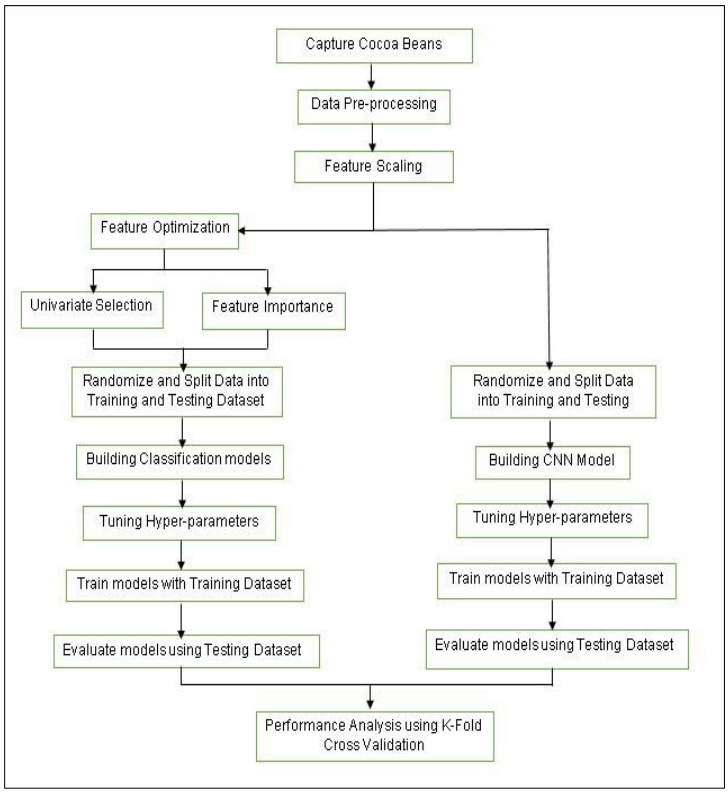

Figure 3: Workflow diagram of the proposed method

For the convolution 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 'Image Data Generator' class available is Keras library. It also minimizes the chances of over fitting. 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.

2.2 Feature extraction

A set of twenty-three image features—including area, convex hull area, Max Feret diameter, equivalent ellipse major axis, equivalent ellipse minor axis, equivalent rectangle long side, equivalent rectangle short side, equivalent rectangle diagonal, hydraulic radius, Elongation factor, compactness factor, Heywood circularity factor, and seven HU moment features—are extracted from the images in order to train the proposed models [31].

2.3 Feature optimization

In most cases, not every independent variable in the dataset has the same amount of influence on the dependent feature when it comes to machine learning models [31]. Certain aspects could not have much of an effect. In order to enhance machine learning models, superfluous features are removed using feature optimisation [7]. It keeps accuracy intact while cutting down on model complexity and training time. Univariate analysis and feature importance are the two feature optimisation strategies used in this study.

2.4 Feature scaling

In our approach we have implemented the feature scaling algorithm over the cocoa bean's images [31]. Let min be the minimum pixel value of the image matrixI_{min} Now, subtract I_{min} from each pixel of the I. Similarly, let *max* is the maximum pixel value of the I_{max} image matrix. We define an $ImageFactor(I_f)$ as follows:

$$ImageFactor(I_f) = \frac{(I - I_{min})}{(I_{max} - I_{min})} \quad (1)$$

Now, each pixel value of the I_S image matrix is updated by dividing with the $ImageFactor(I_f)$. The new I_S values are defined as follows:

$$ImageFactor(I|S) = I_S / I_f \quad (2)$$

3 Details of proposed Cocoa-Net

In this section, we'll go over the details of the proposed Cocoa-Net model, a Deep Learning-based technology with the potential for excellent accuracy in the field of Cocoa Beans recognition.

Cocoa-Net utilizes a series of convolutional layers followed by pooling layers, flattens layers, fully connected layers, and an output layer. The first convolution layer uses 6 filters, and subsequent layers use 16, 64, and more filters with strides and batch normalization. The architecture employs Max Pooling, ReLU activation functions, and a final SoftMax activation function for classification. Whereas Alexnet Consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully connected layers at the end. The first convolutional layer uses 96 filters, with subsequent layers increasing the number of filters up to 384 and 256. Cocoa-Net uses varying kernel sizes and filters specifically tuned for detecting features relevant to cocoa beans, such as edge detection and curved features.

Each of CNN's layers are responsible for a distinct function. The proposed Cocoa-Net structure contains convolution layer, pooling layer, flatten layer, fully connected layer and Output layer. Figure 4 displays the proposed Cocoa-Net structure with input and output image shape representations.

a. Convolutional layer: This layer performs a dot $\odot$ product between the weights and small patches of the input data to produce a feature map. The layer is called a convolutional layer because it performs a convolution operation on the input data.

For example, the first convolution layer uses a 5×5 kernel with 6 filters, while subsequent layers adjust the kernel and filter sizes to optimize feature extraction. Cocoa-Net is tailored for cocoa bean recognition with specific architectural choices, filter configurations, and optimization techniques, whereas AlexNet is a general-purpose CNN model known for its broader application in image recognition tasks.

In the subsequent second convolution layer, a 3×3 kernel, 16 filters in total, and one stride are utilised. We resemble to the maximum pooling layer with a 2×2 kernel size. In the final convolution layer, a 3×3 kernel, 64 kernels in total, one stride, batch normalisation, and dropout are utilised.

b. Pooling layer: This layer down samples the feature map by taking the maximum or average value of a set of adjacent values. Pooling helps to reduce the spatial size of the data, which reduce the computational cost and helps to reduce over fitting.

Following the Convolutional layer, the Pooling Layer's operation is performed. Pooling decreases the quantity of information in each feature retrieved from the layer above while preserving the most essential data. Pooling layer reduces the dimensionality. Here, Max Pooling was utilised. Here, we have taken a 2×2 filter and a stride of length 2 is used.

c. ReLU layer: This layer applies the Rectified Linear Unit (*ReLU*) activation function to the output of the previous layer. The *ReLU* function replaces all negative values in the output with zeros, allowing the network to model non-linear relationships between the input and output. The ReLU activation function is used to introduce non-linearity to a DNN model. In neural networks, particularly convolutional neural networks (CNNs) and multilayer perceptions, it is the most widely employed activation function.

d. Fully Connected layer: This layer is used to perform classification or regression tasks. The fully connected layer takes the output of the previous layer, flattens it into a vector, and then multiplies it by a weight matrix to produce the final output.

The experiment employed a batch size of eight, a learning rate of 0.001, a training epoch of fifty, and a cross entropy loss function. After each convolution and pooling layer, the ReLU activation function is utilised, whereas the SoftMax activation function is used at the output layer. The image is resized as 48 pixels by 48 pixels. The layers are sequentially stacked along with the size of the output metrics after each layer is shown in Figure 4. Cocoa-Net case study utilising loss function Categorical Sparse Cross Entropy illustrated in Table II.

Cocoa-Net is designed with an optimized number of layers and filters, allowing it to extract significant features from the dataset more efficiently compared to ResNet-50's deeper architecture. Cooa-Net is specifically tailored for the cocoa bean classification task. Its layers and filters are optimized to capture the nuances of cocoa bean images, which might not be as efficiently captured by the more general-purpose ResNet-50. The training techniques used for Cocoa-Net, including the choice of optimizers like Adam, hyper parameters, and data augmentation strategies, contribute to its superior performance. The use of stratified K-fold cross-validation ensures that the model is well-validated and reduces the risk of over fitting. These factors collectively enable Cooa-Net to outperform ResNet-50 in the specific task of cocoa bean classification, despite its relatively simpler architecture. Descriptions of proposed Cocoa-Net parameters are given in Table III.

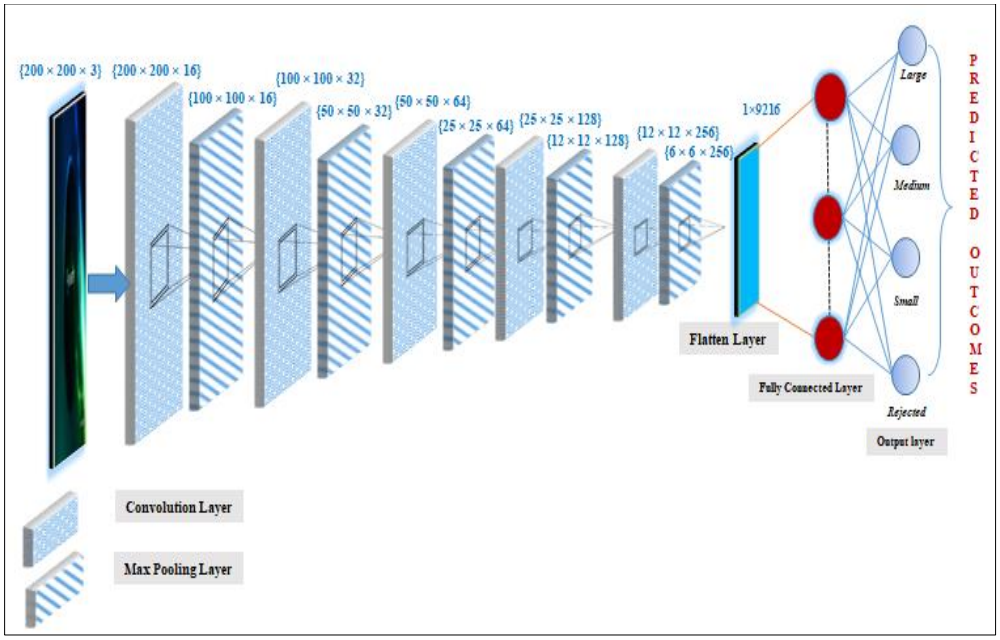

Figure 4: Proposed Cocoa-Net architecture

Table 2: Case study Cocoa-Net with loss function sparse categorical cross entropy

No. of Conv Layer	Optimizer	Filter Distribution (No. Of filters in each layer)	Epoch	Test Accu.	Test Loss
4	RMSprop	(16,32,64,128,256)	50	0.76	1.4
4	adam	(16,32,64,128,256)	30	0.84	0.5
4	adam	(16,32,64,128,256)	50	0.82	0.98
4	adadelta	(16,32,64,128,256)	30	0.38	1.35
4	adam	(16,32,64,128,512)	30	0.81	0.68
4	adam	(16,32,64,128,512)	40	0.83	0.88
5	adam	(16,32,64,128,256,512)	30	0.86	0.44
5	adam	(16,32,64,128,256,512)	40	0.84	0.44
5	adam	(16,32,64,128,256,512)	50	0.86	0.63
5	adagrad	(16,32,64,128,256,512)	30	0.28	1.44
5	adamax	(16,32,64,128,256,512)	30	0.84	0.48
5	adamax	(16,32,64,128,256,512)	40	0.86	0.50
5	adamax	(16,32,64,128,256,512)	50	0.86	0.42
5	Nadam	(16,32,64,128,256,512)	30	0.81	0.78
5	Nadam	(16,32,64,128,256,512)	40	0.7	0.92
5	Nadam	(16,32,64,128,256,512)	50	0.76	0.89

Justification for kernel size:

The initial layer is designed to capture low-level features such as edges and textures. Using a larger 5×5 kernel helps in detecting these fundamental features over a slightly broader area of the image, which is crucial for effective feature extraction from the outset. The smaller 3×3 kernels are used in subsequent layers to capture more detailed and localized features. This size is commonly used in CNNs because it allows the network to learn intricate patterns and hierarchical features while keeping the computational load manageable. Smaller kernels are effective in maintaining spatial resolution and are sufficient to detect complex features when combined in deeper layers.

Justification for number of filters:

Starting with a small number of filters helps in learning basic features without overwhelming the model with parameters. This approach ensures the model captures essential features first before diving into more complex patterns. The exponential increase in the number of filters as the network goes deeper (16 in the second layer to 64 in the third) is a strategic design choice. This progression allows the network to gradually learn more complex and higher-level features from the

input data. Each subsequent layer can detect more sophisticated patterns and combinations of features detected by the previous layers, enhancing the model's ability to capture fine-grained details necessary for accurate classification.

In this study the 'Sequential' class available in Keras library is used to build the proposed Cocoa-Net 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 is 16, 32, 64, 128 and 256 respectively. Activation function used at each convolutional layer is ReLU 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. Different optimization algorithms like- Adaptive Momentum (Adam), Root Mean Square Propagation (RMS Prop), Adaptive Gradiant optimizer (Adagrad), Adamax optimizer and Nadam optimizer are tested for optimizing the model where each time with different hyper parameters Adam optimizer results with maximum accuracy. The training images and class labels are fitted to the model with epoch value set to 30. These hyper parameters are measured after training the model with different sets of hyper parameters 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 0.86 while the loss is 0.44.

For this experiment, the default optimizer is Adam. Adam is the finest alternative for the first training of deep learning networks. The subsequent layer is the flatten layer, which takes the output of the preceding layers and flattens it into a single vector that may be used as input for the subsequent stage. The objective of the Flatten layer is to reduce the matrix to a vector with a single dimension. Fully connected layer provides the final probability associated with each class. The 1D data used as the input to the neurons of this layer, which execute a dot product of this input data and the neuron weights to generate a single probability value per neuron. The likelihood of each emotion is estimated after applying the softmax function. When all probability values are compared, the one with the highest probability is considered as the final emotion for the supplied input. We used dropout and regularization strategies in our experiment to deal with the limited size of the datasets. Figure5 describes the model summary.

Table 3: Description of proposed Cocoa-Net architecture.

Layer Name	Network parameter	Training parameter
Input layer: Image	----------	Loss function: categorical cross entropy, Optimizer: Adam, Number of Epochs:50, Learning rate: 0.001
Convolution	Kernel size= 5×5, Number of kernels=6, Stride=1×1, Padding=Same, Activation function= ReLU	
Pooling	Pool size =2×2, Stride= 1×1, Padding=Same, Pool technique= Max pool	
Convolution	Kernel size= 5×5, Number of kernels =16, Stride=1×1, Padding=Same, Activation function=ReLU	
Pooling	Pool size =2×2, Stride= 1×1, Padding=Same, Pool technique=Max pool	
Convolution	Kernel size= 3×3, Number of kernels= 64, Stride=1×1, Padding=Same, Activation function=ReLU	
Pooling	Pool size =2×2, Stride= 1×1, Padding= Same, Pool technique= Max pool	
Flatten	----------	
Fully Connected layer	Number of neurons=128	
Output layer	Number of neurons=7, Activation function=SoftMax	

4 Simulation 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 $Accuracyscore$. 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:

$$Accuracyscore = (T_P + T_N)/(T_P + T_N + F_P + F_N) \quad (7)$$

$$F\beta score = (1 + \beta^2) \times (Precision \times Recall)/(\beta^2 \times Precision + Recall) \quad (8)$$

The ratio of correctly predicted positive observations to all predicted positive observations is known as precision. The ratio of all observations in the actual positive class to the correctly predicted positive observation is known as recall. F1 score is equal to Fβ score for β = 1.

$$Precision = T_P/(T_P + F_P) \quad (9)$$

$$Recall = T_P/(T_P + F_N) \quad (10)$$

Two classification algorithms used for distance-based algorithms are K-Nearest Neighbour (KNN) and Support Vector Machine (SVM) [23, 24]. Cases are categorized by KNN according to their similarity, which is determined by a distance matrix. "Neighbours" are cases that are close to one another. The most widely used class label or the class label with the majority value from its neighbours is taken into consideration when predicting classes for unknown data points. However, SVM is effective at managing the non-linearity of the dataset by converting the data into a higher dimensional space.

Two well-liked algorithms for tree-based algorithms are Random Forest classifier and Decision Tree classifier. The decision rules are represented by the branches of the decision tree classifier, the sample dataset's features are represented by the internal nodes, and the final output, or class labels are represented by the leaf nodes. By reducing the impurity at each stage, it divides the training records into segments using recursive partitioning. However, when a decision tree is fully developed, it has low bias, indicating that the model is overfit to the training dataset, and high variance, indicating that the model is likely to produce a high number of errors when working with fresh test data. Instead of using a single decision tree, the Random Forest Classifier considers multiple high variance decision trees that are generated from subsets of the main dataset. The high variance is then converted to low variance by combining the trees according to a majority vote. Furthermore, since we are randomly sampling the rows and columns, any changes we make to the model will affect all of the decision trees equally, so adding new or changing existing data won't have a significant impact. Entropy is the criterion function chosen for both of these tree-based algorithms in order to choose the internal or root nodes at various decision tree levels.

```
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
_________________________________________________________________
```

Figure 5: Our proposed Cocoa-Net model summary

Finding the tree whose nodes have the least amount of entropy is the aim. The maximum depth of the decision tree is therefore determined to be 4 after attempting a range of values from 1 to 10 in order to achieve the maximum accuracy score of 0.74 and F1 score of 0.73. The criterion function chosen for splitting in the Decision Tree model is "entropy" with the "best" splitter strategy. To obtain the maximum accuracy score of 0.74 and the F1 score of 0.71, the random forest classifier is trained using the same criterion function for 150 decision trees with a maximum depth of 4.

Table IV describes the stratified K-fold cross validation performance evaluation of the KNN, SVM, Decision Tree, Random Forest, proposed Cocoa-Net, and ResNet50 algorithms. Table V defines performance evaluation of the KNN, SVM, Decision Tree, Random Forest, proposed Cocoa-Net, and ResNet50 algorithms. The training dataset for the KNN classification model has a K value between 1 and 20. Additionally, based on the testing dataset, it is found that 10 is the optimal minimum value for K, for which the model predicts with a maximum accuracy score of 0.76 and an F1 score of 0.71.

Cocoa-Net outperforms for following reason:

- Cocoa-Net employs a series of convolutional and pooling layers to effectively extract features from the input images. Each convolutional layer in Cocoa-Net applies filters to detect different features, such as edges, textures, and shapes, which are crucial for accurate classification of cocoa beans. The network's structure allows it to capture fine-grained details and hierarchical features, enhancing its ability to differentiate between different types of cocoa beans.
- The first convolutional layer uses a 5×5 kernel with six filters, which helps in capturing basic features like edges. This larger kernel size at the beginning allows for a broader receptive field, enabling the network to gather more contextual information from the initial layers.
- Following layers use 3×3 kernels, which are standard in many state-of-the-art CNN architectures, as they balance the trade-off between computational efficiency and the ability to capture fine details. The increase in the number of filters from 16 in the second convolutional layer to 64 in the third convolutional layer is exponential, allowing the network to learn more complex features as the depth increases.
- The use of pooling layers with a stride of 2 helps in down-sampling the feature maps, reducing the spatial dimensions while preserving the most significant features. This process reduces the computational load and helps in mitigating overfitting by providing a form of translational invariance.

By analyzing the design choices and the performance metrics of Cocoa-Net, it is clear that the model's architecture is tailored to efficiently extract relevant features with fewer layers, resulting in superior performance compared to deeper networks like ResNet-50. The careful selection of kernel sizes, the exponential increase in the number of filters, and the strategic use of pooling layers contribute to the model's ability to outperform other models while maintaining computational efficiency.

Figure 6 describes (proposed Cocoa-Net and ResNet50) training and validation performance in terms of iteration on the cocoa beans datasets, respectively. Figure 7 shows the classification reports that were produced for each of the five classification models. Since the dataset is not perfectly balanced, stratified K-fold cross validation with 10 folds is ultimately used to evaluate the classification models. It guarantees that the original data, training data, and testing data all have the same percentage of target features for the various classes

Cocoa-Net is crucial for understanding the down-sampling process:

- In the absence of specific information, it is generally assumed that the stride for convolutional layers is set to 1. This is a common practice in many CNN architectures where detailed stride information is not specified. The assumption helps maintain the resolution of feature maps until pooling layers are applied for down-sampling.
- Stride lengths in convolutional layers directly affect the spatial dimensions of the output feature maps. A stride of 1 ensures that the feature maps retain their spatial dimensions, while a larger stride reduces the dimensions more aggressively. This control over dimensions can be crucial for capturing fine details in the earlier layers and gradually abstracting information in deeper layers.
- Using a stride of 1 in convolutional layers helps balance the computational load by ensuring that down-sampling is primarily handled by pooling layers. This strategy allows the network to learn detailed spatial hierarchies before significant dimensionality reduction occurs, potentially improving the feature learning process.
- Maintaining a stride of 1 in convolutional layers before applying pooling operations with a larger stride (e.g., stride of 2) helps in retaining more detailed features. This approach is beneficial for models like Cooa-Net, which aims to outperform deeper networks like ResNet-50 with a more efficient architecture.

Table 4: Performance evaluation of K folds cross validation of KNN, SVM, Decision Tree, Random Forest, proposed Cocoa-Net, and ResNet50.

Fold	KNN	SVM	Decision Tree	Random Forest	Proposed Cocoa-Net	ResNet50
1	0.727	0.727	0.682	0.773	0.708	0.705
2	0.591	0.682	0.727	0.682	0.971	0.931
3	0.727	0.773	0.727	0.773	0.883	0.863
4	0.682	0.773	0.682	0.727	0.783	0.783
5	0.636	0.591	0.636	0.682	0.723	0.723
6	0.773	0.773	0.773	0.818	0.758	0.748
7	0.773	0.773	0.727	0.773	0.767	0.767
8	0.864	0.864	0.818	0.818	0.792	0.792
9	0.591	0.591	0.682	0.636	0.967	0.947
10	0.727	0.727	0.773	0.772	0.917	0.907
Max	0.861	0.861	0.821	0.821	**0.971**	**0.947**
Min	0.591	0.591	0.640	0.640	**0.708**	**0.705**
Mean	0.713	0.731	0.721	0.751	**0.832**	**0.812**

Table 5: Performance evaluation of KNN, SVM, Decision Tree, Random Forest, proposed Cocoa-Net and ResNet50.

Method	Machine Learning Techniques				Deep Learing	
	KNN	SVM	Decision Tree	Random Forest	Proposed Cocoa-Net	ResNet50
Accuracy	0.760	0.730	0.740	0.740	**0.850**	0.840
Precision	0.568	0.788	0.740	0.850	**0.852**	0.841
Recall	0.585	0.610	0.600	0.580	**0.850**	0.841
F Score	0.570	0.645	0.640	0.610	**0.850**	0.835

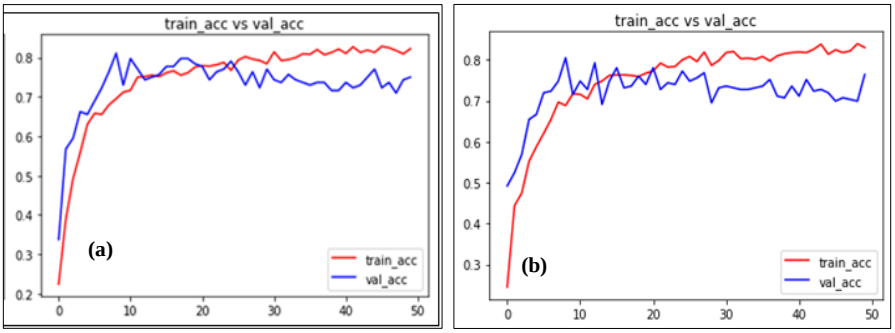

Figure 6: Training and validation performances analysis using cocoa bean dataset a) Proposed Cocoa-Net, b) ResNet50 images.

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

Figure**7:** Classification accuracy Score of (KNN, SVM, Decision Tree, Random Forest [31]) Proposed Cocoa-Net

5 Conclusion

Pattern recognition is a complex task due to the numerous aspects of the image that must be examined in order to achieve precise results. The four conventional machine learning algorithms KNN, SVM, Decision Tree, Random Forest as well as our proposed Cocoa-Net algorithm and ResNet50 for cocoa bean classification were compared in this study. Through the visualization of key image regions, the significance of the extracted features was examined in order to offer insights. The accuracy range for proposed Cocoa-Net proved to be better, ranging from 0.72 to 0.97 with a loss value in the range of 0.59 to 0.88. The resulting accuracy and F1 scores obtained by the four ML models are in ranges between 0.71 to 0.75 and 0.68 to 0.73, respectively. The Random Forest Classifier has the highest mean accuracy score of 0.75 according to the K-fold cross validation results. The proposed Cocoa-Net model and ResNet50 predicts with the overall mean accuracy score of 0.85, 0.84 respectively. Proposed Cocoa-Net contributes to the development of solutions through the visualization of techniques, by offering pertinent information for future research based on comprehensive learning (applied to the food industry) algorithms. As a result, the Cocoa-Net approach may be applied as a quick and impartial way to distinguish among various types of cocoa beans in the food business. Additionally, the food industry can enhance supply chain product tracking by utilizing visualization techniques.

Conflict of interest

The authors declare that they have no conflict of interest.

Data availability

All data analysed are included in this paper.

References

[1] R. Essah, D. Anand, and S. Singh (2022) An intelligent cocoa quality testing framework based on deep learning techniques. *Measurement: Sensors* 24: 100466. https://doi.org/10.1016/j.measen.2022.100466

[2] R. Hayati, Z. Zulfahrizal, and A. A. Munawar (2021) Robust prediction performance of inner quality attributes in intact cocoa beans using near infrared spectroscopy and multivariate analysis. *Heliyon* 7: (2): 1-7. https://doi.org/10.1016/j.heliyon.2021.e06286

[3] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem (2019) A survey on the role of IoT in agriculture for the implementation of smart farming. *IEEE Access* 7: 156237–156271. https://doi.org/ 10.1109/ACCESS.2019.2949703

[4] C. Yoon, M. Huh, S. G. Kang, J. Park, and C. Lee (2018) Implement smart farm with IoT technology. *in: 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 749–752. https://doi.org/ 10.23919/ICACT.2018.8323908

[5] I. Abdulai, P. Vaast, M. P. Hoffmann *et al.* (2018) Cocoa agroforestry is less resilient to sub-optimal and extreme climate than cocoa in full sun. *Global Change Biol.* 24 (1): 273–286. https://doi.org/ 10.1111/gcb.13885

[6] D. N. de Oliveira, A. C. B. Camargo, C. F. O. R. Melo *et al.* (2018) A fast semiquantitative screening for cocoa content in chocolates using MALDI-MSI. *Food Res.* Int. 103: 8- 11. https://doi.org/10.1016/j.foodres.2017.10.035.

[7] M. Ghosh, and A. Dey (2023) Fractional-weighted Entropy-based Fuzzy G-2DLDA Algorithm: A New Facial Feature Extraction method. *Mutimedia Tools and Applications*, 82 (2): 2689–2707. https://doi.org/10.1007/s11042-022-13328-7.

[8] Y. Adhitya, S. W. Prakosa, M. Köppen *et al.* (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] F. Nazir, M. N. Majeed, M. A. Ghazanfar *et al.* (2019) Mispronunciation detection using deep convolutional neural network features and transfer learning—based model for Arabic phonemes. *IEEE Access* 7, 52589- 52608. https://doi.org/ 10.1109/ACCESS.2019.2912648

[10] M. Mukhopadhyay, A. Dey, A. Ghosh *et al.* (2022) Facial emotion recognition based on Textural pattern and Histogram of Oriented Gradient. *Proceeding of the ICACIS 2022*, pp- 111-119. https://doi.org/10.1007/978-3-031-25088-0_9.

[11] J. K. Sing, A. Dey, M. Ghosh (2019) Confidence Factor Weighted Gaussian Function Induced Parallel Fuzzy Rank Level Fusion for Inference and its Application to Face Recognition. *Information Fusion* 47: 60- 71. https://doi.org/10.1016/j.inffus.2018.07.005

[12] A. Z. da Costa, H. E. Figueroa, and J. A. Fracarolli (2020) Computer vision-based detection of external defects on tomatoes using deep learning. *Biosyst Eng.*, 190: 131–144. https://doi.org/10.1016/j.biosystemseng.2019.12.003

[13] A. Bhargava and A. Bansal (2020) Quality evaluation of mono & bi-colored apples with computer vision and multispectral imaging. *Multimedia Tools and Applications*, 79: 7857–7874, https://doi.org/10.1007/s11042-019-08564-3

[14] L. F. S. Pereira, S. Barbon, N A. Valous *et al.* (2018) Predicting the ripening of papaya fruit with digital imaging and random forests. *Comput Electron Agric* 145: 76–82. https://doi.org/10.1016/j.compag.2017.12.029

[15] H. Tian, T. Wang, Y. Liu *et al.* (2020) Computer vision technology in agricultural automation—a review. *Inf Process Agric* 7(1):1–19, https://doi.org/10.1016/j.inpa.2019.09.006.

[16] S. Navulur, A.S.C.S Sastry and M N G. Prasad (2017) Agricultural management through wireless sensors and Internet of Things. *Int. J. Electr. Comput. Eng.* 7: 3492–3499. http://doi.org/10.11591/ijece.v7i6.pp3492-3499

[17] S. K. Behera, A. K. Rath, A. Mahapatra *et al.* (2020) Identification, classification & grading of fruits using machine learning & computer intelligence: A review. *J. Ambient Intell. Human. Comput.*, 11: 1- 11. https://doi.org/10.1007/s12652-020-01865-8

[18] K. G. Liakos, P. Busato, D. Moshou *et al.* (2018) Machine Learning in Agriculture: A Review. *Sensors*, 18(8), 2674. https://doi.org/10.3390/s18082674

[19] V. Srikanth, G. K. Rajesh, A. Kothakota *et al.* (2020) Modeling and optimization of developed cocoa beans extractor parameters using box behnken design and artificial neural network. *Computers and Electronics in Agriculture*, 177: 105715. https://doi.org/10.1016/j.compag.2020.105715

[20] O. Saha Mandal, A. Dey, A. Ghosh, and R. N. Shaw (2022) Fruit-Net: Fruits recognition system using Convolution Neural Network. *Proceeding of the ICACIS 2022*, pp- 120-133. https://doi.org/10.1007/978-3-031-25088-0_10

[21] H. S. Gill and B S. Khehra (2021) Hybrid classifier model for fruit classification. *Multimed Tools Appl* 80: 27495–27530. https://doi.org/10.1007/s11042-021-10772-9

[22] G. Ashiagbor, O. A. Asare-Ansah, E. Boakye Amoah *et al.* (2023) Assessment of machine learning

classifiers in mapping the cocoa-forest mosaic landscape of Ghana. *Scientific African*, 20, e01718. https://doi.org/10.1016/j.sciaf.2023.e01718

[23] A. Dey, S. Chowdhury, and M. Ghosh (2017) Face Recognition using Ensemble Support Vector Machine. *Proceeding of the ICRCICN 2017*, pp. 46-50. https://doi.org/10.1109/ICRCICN.2017.8234479

[24] A. Dey, and S. Chowdhury (2020) Probabilistic Weighted induced Multi-Class Support Vector Machines for Face Recognition. *Informatica Si*, 44 (4): 345- 353. https://doi.org/10.31449/inf.v44i4.3142

[25] M. S Hossain, Al-Hammadi M, and Muhammad G. (2018) Automatic Fruits Classification Using Deep Learning for Industrial Applications. *IEEE Trans. Ind. Inform*. 15: 1027–1034, https://doi.org/ 10.1109/TII.2018.2875149.

[26] B. Dhiman, Y. Kumar, and M. Kumar (2022) Fruit quality evaluation using machine learning techniques: review, motivation and future perspectives. *Multimedia Tools and Applications*, *81*(12): 16255-16277. https://doi.org/10.1007/s11042-022-12652-2

[27] J. Tan, B. Balasubramanian, D. A. Sukha *et al.* (2019) Sensing fermentation degree of cocoa (Theobroma cacao L.) beans by machine learning classification models based electronic nose system. *J. Food Process Eng.* 42 (4): e13175. https://doi.org/ 10.1111/jfpe.13175

[28] J. Cruz-Tirado, J. A. Fernandez Pierna, H. Rogez *et al.* (2020) Authentication of cocoa (theobroma cacao) bean hybrids by nir-hyperspectral imaging and chemometrics. *Food Control* 118, 107445. https://doi.org/ 10.1016/j.foodcont.2020.107445

[29] A. Dey, S. Chowdhury, and J. K. Sing, Performance Evaluation on Image Fusion Techniques for face recognition, (2018) International Journal Computational Vision and Robotics. Vol. 8, No. 5, pp. 455- 475, https://doi.org/ 10.1504/IJCVR.2018.095000

[30] A. Dey, S. Chowdhury, M. Ghosh, S. Kahali (2023) T2-Fuzzy Multi-Fused Facial Image Fusion (T2FMIF): An Efficient Face Recognition, Journal of Intelligent & fuzzy system, Vol. 45, No. 1, pp.743-761. https://doi.org/10.3233/JIFS-224288

[31] S. Das, A. Akuli, S. Biswas *et al.* (2022) Discrimination of Cocoa Beans using Structural Image Features: An Experimental Analysis. *IEEE IAS Global Conference on Emerging Technologies (GlobConET)*, pp. 1138-1142. https://doi.org/10.1109/GlobConET53749.2022.9872329

[32] H. Tercan and T. Meisen (2022) Machine learning and deep learning based predictive quality in manufacturing: a systematic review. *J. Intell. Manuf.* 33: 1879-1905. https://doi.org/10.1007/s10845-022-01963-8

[33] T. H. E. Kim, H.R Kim, Y. J. Cho (2021) Product inspection methodology via deep learning: an overview. *Sensors* 21 (15): 5039. https://doi.org/10.3390/s21155039

[34] J. F. Lopes, V. G. T. da Costa, F. D. Barbin *et al.* (2022) Deep computer vision system for cocoa classification. *Multimed. Tool. Appl.* 81: 41059–41077. https://doi.org/10.1007/s11042-022-13097-3

[35] C. D. Anggraini, A. W. Putranto, Z. Iqbal *et al.* (2021) Preliminary study on development of cocoa beans fermentation level measurement based on computer vision and artificial intelligence IOP Conference Series: earth and Environmental Science. *IOP Publishing* 924 (1): 012019. https://doi.org/10.1088/1755-1315/924/1/012019

[36] I. O. Abu, Z. Szantoi, A. Brink *et al.* (2021) Detecting cocoa plantations in Coted'Ivoire and Ghana and their implications on protected areas. *Ecol. Indicat.* 129: 107863. https://doi.org/10.1016/j.ecolind.2021.107863

[37] M. M. Oliveira, B. V. Cerqueira, S. Barbon *et al.* (2021) Classification of fermented cocoa beans (cut test) using computer vision. *J. Food Compos*. Anal. 97: 103771. https://doi.org/10.1016/j.jfca.2020.103771

[38] M. Mukhopadhyay, A. Dey and S. Kahali (2023) A Deep-Learning-Based Facial Expression Recognition Method Using Textural Features. *Neural Computing and Applications*, 35 (9): 6499–6514. https://doi.org/10.1007/s00521-022-08005-7

[39] M. Ghosh, A. Dey and S. Kahali (2022) Type-2 fuzzy blended improved D-S evidence theory-based decision fusion for face recognition. Appl. Soft Comput. 125: 109179. https://doi.org/10.1016/j.asoc.2022.109179

[40] M. Ghosh, A. Dey, S. Kahali (2024) A weighted fuzzy belief factor-based D-S evidence theory of sensor data fusion method and its application to face recognition. *Mutimedia Tools and Applications*, 83: 10637–10659. https://doi.org/10.1007/s11042-023-16037-x