Scientific Research Publishing

Open Journal of Applied Sciences, 2024, 14, 2955-2972
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Detection of Cocoa Leaf Diseases Using the CNN-Based Feature Extractor and XGBOOST Classifier

Kouassi Simeon Kouassi[1,2], Mamadou Diarra[1], Kouassi Hilaire Edi[2], Brou Jean-Claude Koua[1]

[1]Mechanics and Computer Science Laboratory, Felix Houphouët-Boigny University, Abidjan, Ivory Coast
[2]Mathematics and Computer Science Laboratory, Nangui Abrogoua University, Abidjan, Ivory Coast
Email: simeon.kouassi@gmail.com

How to cite this paper: Kouassi, K.S., Diarra, M., Edi, K.H. and Koua, B.J.-C. (2024) Detection of Cocoa Leaf Diseases Using the CNN-Based Feature Extractor and XGBOOST Classifier. *Open Journal of Applied Sciences*, **14**, 2955-2972.
https://doi.org/10.4236/ojapps.2024.1410193

Received: September 28, 2024
Accepted: October 25, 2024
Published: October 28, 2024

Abstract

Among all the plagues threatening cocoa cultivation in general, and particularly in West Africa, the swollen shoot viral disease is currently the most dangerous. The greatest challenge in the fight to eradicate this pandemic remains its early detection. Traditional methods of swollen shoot detection are mostly based on visual observations, leading to late detection and/or diagnostic errors. The use of machine learning algorithms is now an alternative for effective plant disease detection. It is therefore crucial to provide efficient solutions to farmers' cooperatives. In our study, we built a database of healthy and diseased cocoa leaves. We then explored the power of feature extractors based on convolutional neural networks such as VGG 19, Inception V3, DenseNet 201, and a custom CNN, combining their strengths with the XGBOOST classifier. The results of our experiments showed that this fusion of methods with XGBOOST yielded highly promising scores, outperforming the results of algorithms using the sigmoid function. These results were further consolidated by the use of evaluation metrics such as accuracy, mean squared error, F score, recall, and Matthews's correlation coefficient. The proposed approach, combining state of the art feature extractors and the XGBOOST classifier, offers an efficient and reliable solution for the early detection of swollen shoot. Its implementation could significantly assist West African cocoa farmers in combating this devastating disease and preserving their crops.

Keywords

Machine Learning, Cocoa Leaf Diseases, Deep Learning, Convolutional Neural Network, Feature Extraction, Image Recognition, XGBOOST

1. Introduction

The Swollen Shoot viral disease, or Cocoa Swollen Shoot Virus (CSSV), has been a major threat to cocoa cultivation in West Africa for over 70 years [1] [2]. First described in Ghana in 1936, it appeared in Côte d'Ivoire in 1943 and has since spread extensively throughout the subregion, particularly in Côte d'Ivoire, Ghana, and Togo [2] [3]. This disease poses a significant challenge to the Ivorian economy, the world's largest cocoa producer [4] [5]. Indeed, the Swollen Shoot virus has devastated, and continues to seriously affect, cocoa production in the subregion, threatening the livelihoods of millions of smallholder farmers. This fully highlights the severity of the threat to food security and the incomes of rural communities.

The symptoms of the disease vary, but mainly include abnormal swelling of shoots, branches, and roots, leaf discolouration, leaf deformation and premature leaf drop, pod stunting, and a significant reduction in yield, which may even lead to the death of the tree [5]. The disease is spread by a vector mealybug [5] [6]. In response to this plague, current control protocols rely on symptom identification and diagnosis, eradication of infected trees, vector management through insecticide use, and replanting with resistant varieties [5]-[7]. However, the primary challenge remains the early detection of disease hotspots. Manual survey methods are rudimentary and unsuitable for large plantations. It is in this context that Deep Learning technologies offer promising prospects for automating symptom detection and enabling rapid intervention, thus limiting the virus's spread [8].

The objective of this paper is to explore recent advances in applying Deep Learning to the detection and monitoring of plant diseases, with a specific focus on the case of cocoa swollen shoot. We will examine hybrid methods based on CNN feature extractors and the XGBOOST classifier to assess AI's potential to enhance the resilience of cocoa crops and support farming communities in their fight against this persistent threat.

Our study's contributions are as follows:

- Creation of a database of swollen shoot-infected cocoa leaves and healthy leaves: We established a comprehensive and rigorously labelled database, essential for research on cocoa disease detection. This database will not only allow testing of different classification algorithms but also serve as a reference for future studies.
- Application of several state-of-the-art algorithms: We employed the VGG-19, Inception V3, DenseNet-201 algorithms and a custom CNN, each with a classifier based on the sigmoid function. These models are known for their effectiveness in image classification and allow the comparison of the performance of various deep neural networks in the context of cocoa disease detection.
- Evaluation of algorithms with the XGBOOST classifier: In addition to the sigmoid-based classifiers, we used the XGBOOST classifier for the same algorithms (VGG-19, Inception V3, DenseNet-201, and the custom CNN). XGBOOST is renowned for its high classification performance, enabling the evaluation of

the impact of different classifiers on model efficacy.

- Comprehensive comparison of methods: A detailed comparative analysis of the performance of different combinations of algorithms and classifiers. This comparison helps identify the best approaches for cocoa disease detection based on criteria such as accuracy, recall, and F1-score.

These contributions highlight not only the innovation and rigour of our approach but also its potential to significantly improve early detection and management of cocoa diseases.

Following this introductory section, the remainder of the paper is structured as follows: Section 2 reviews the literature. Section 3 describes the materials and proposed methodology. Section 4 presents the experimental results of the various test scenarios conducted. Section 5, dedicated to discussion, highlights the discrepancies between the results obtained from testing the different scenarios on the same dataset. Finally, Section 6 concludes the study and offers suggestions for future work.

2. State of the Art

Several studies have been conducted in the field of computer vision, particularly focusing on the detection of agricultural pathologies. Below, we present some works that are relevant to our study.

Kacoutchy Jean Ayikpa *et al.* [9] conducted an in-depth study primarily focused on evaluating the effectiveness of feature extractors and the impact of colour spaces on these extractors for classifying the maturity of cocoa pods using images from a database. Several similarity measures were applied in conjunction with feature extractors to classify pod images according to their maturity level. Classification performance was examined from various angles, using feature extractors based on convolutional neural networks (CNN) and the Gray-Level Co-occurrence Matrix (GLCM) across different colour spaces, such as RGB, HSV, Lab, and Luv [10] [11]. The results showed that the Lab colour space, combined with GLCM and the chi-square distance similarity measure, produced the best results, with an accuracy of 99.60%.

Ker Sing Soh *et al.* [12] proposed the use of convolutional neural networks to classify cocoa diseases, focusing on addressing the significant agricultural and economic impacts of black pod rot and pod borer. They utilised a dataset of 4390 images and evaluated five CNN architectures: Custom CNN, VGG-16, EfficientNetB0, ResNet50, and LeNet-5, assessing their ability to accurately identify disease presence [13] [14]. The Custom CNN model was the most effective, achieving an accuracy of 91.79%, precision of 91.79%, recall of 91.79%, an F1-score of 82.08%, sensitivity of 96.69%, and specificity of 98.40%, demonstrating a strong capability to accurately classify healthy and diseased plants.

Mamadou Coulibaly *et al.* [15] developed a system to recognize symptoms of the Swollen Shoot epidemic through feature extraction from cocoa pods, enabling better diagnosis of plants affected by the disease. Their study was based on

convolutional neural networks, with results showing that a deep CNN achieved record accuracy on a supervised learning dataset, with a rate of 84%.

Kacoutchy Jean Ayikpa *et al.* [16] employed various approaches for classifying and recognising coffee leaf diseases, using both traditional machine learning methods and deep learning techniques. The evaluation demonstrated high efficiency, with 100% accuracy for traditional methods such as SVM and Random Forest, as well as for deep learning methods like MobileNet and Custom CNN.

Bueno *et al.* [17] investigated a technique to determine cocoa maturity by analyzing the acoustic properties of cocoa beans and pods. Their approach involved extracting distinguishable features from acoustic signals and applying convolutional neural networks (CNN) to classify the sound of cocoa. Their model achieved a classification accuracy of 97.46% for determining the maturity of unharvested cocoa pods.

Sandra Kumi *et al.* [18] developed a smartphone application based on deep learning to detect Swollen Shoot disease and black pod rot in cocoa. This mobile application, designed with integrated machine learning techniques, allows cocoa farmers to take photos of cocoa pods and upload them for diagnosis, which is processed on a cloud service. They trained and tested four CNN models, achieving an accuracy of over 80% with the SSD MobileNet V2 model.

Darlyn Buenano Vera *et al.* [19] introduced a deep learning model for identifying cocoa pod diseases, focusing on "moniliasis" and "black pod rot" using EfficientDet-Lite4 [20], a lightweight and efficient object detection model. A dataset containing images of healthy and diseased cocoa pods was used to train the model to detect and localize disease manifestations with considerable accuracy. Improvements in model training and evaluation demonstrated its ability to recognize and classify diseases through image analysis. Additionally, the model's features were integrated into a native Android mobile application with a user-friendly interface, allowing young or inexperienced farmers to quickly and accurately assess cocoa pod health.

Ciro Rodriguez *et al.* [21] proposed a machine learning approach for identifying cocoa tree diseases (Theobroma cacao L.) and preventing crop losses, as farmers often lack immediate tools for timely disease detection. They used image processing and analysis techniques such as HoG (Histograms of Oriented Gradients) [22], LBP (Local Binary Patterns) [23], and SVM (Support Vector Machine) [24] [25] for classification, to determine whether the cocoa tree was infected with a disease. The results revealed that applying SVM, Random Forest, and artificial neural networks (ANN) with feature vectors extracted using HoG and LBP algorithms could predict the state of the cocoa tree, with accuracy improving as the dataset size increased.

Annisa Fitri Maghfiroh Harvyanti *et al.* [26] used a deep learning approach to identify one of the most common cocoa diseases, Vascular Streak Dieback (VSD) [27], enabling timely treatment to maintain productivity. Their method relied on analysing leaf images using convolutional neural networks to simplify and accelerate the detection process. They evaluated four CNN architectures, namely

AlexNet, SqueezeNet, Darknet, and a custom CNN, to identify cocoa plants infected with VSD [28] [29]. With a dataset of 1200 images, including 600 healthy and 600 VSD-infected samples, the best results were achieved with the DarkNet-19 model, achieving a test accuracy of 98.61%.

Several studies have been conducted on detecting Swollen Shoot based on cocoa pod images, yielding promising results. However, very few studies have focused on detection based on cocoa tree leaves, the most visible aerial part of the plant. In our study, we aim to detect Swollen Shoot based on symptoms present on cocoa leaves, which could serve as a basis for aerial detection of disease outbreaks.

3. Materials and Methods

This section outlines the proposed methodology for our study, including the dataset and materials used, the convolutional neural network models, the data preprocessing techniques applied, the deep learning methods, and the models proposed.

3.1. Materials

1) Dataset Description

The dataset used in this study consists of cocoa leaf images from plants belonging to the Forastero group, specifically from the Amelonado sub-variety and Trinitario group. The images were collected from three plantations in Côte d'Ivoire. The experimental plantation of the National Centre for Agronomic Research (CNRA) located in Bouaflé, in the central-western region of Côte d'Ivoire (Forastero) and a local farmer's plantation in the same area (Forastero group). The last is a local farmer's plantation in Divo (Forastero and Trinitario group), located in the southern part of Côte d'Ivoire, a region known for high cocoa production.

The dataset comprises 6000 images (2456 px × 3275 px), organized as follows: A first folder containing 2700 images of healthy plant leaves and the second folder containing 3300 images of infected plant leaves showing visible symptoms.

The images were captured in uncontrolled, real-world conditions during normal daylight hours, under clear skies (between 10 a.m. and 4 p.m.), with ambient temperatures ranging from 28˚C to 32˚C, solar irradiance between 200 W/m^2 and 500 W/m^2, and humidity levels between 70% and 90%. The distance between the camera and the subject ranged from one to five meters.

Figure 1 and Figure 2 display images of healthy cocoa leaves and leaves showing symptoms respectively.

Figure 1. Example of holy cocoa leaves.

Figure 2. Example of cocoa leaves showing symptoms of Swollen Shoot.

2) Data Pre-processing

Data pre-processing is a critical phase in any image classification process, as it significantly influences both the performance and efficiency of the models.

For our data pre-processing, we utilized the CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithm. CLAHE is an image processing technique designed to enhance local contrast while mitigating the noise typically amplified by global histogram equalization [30] [31]. The CLAHE pre-processing method consists of the following steps:

- Image Division: The image is divided into small blocks or regions, referred to as "tiles." In our study, we selected blocks of 256 × 256 pixels to retain the distinct characteristics of our images.
- Histogram Equalization: Each block undergoes individual histogram equalization, redistributing pixel brightness to achieve a uniform distribution, thereby enhancing local contrast in each block.
- Contrast Limiting: To prevent excessive noise in homogeneous areas of the image, a maximum threshold is applied to contrast enhancement. This is accomplished by clipping parts of the histogram that exceed a set threshold and redistributing the excess values uniformly.
- Interpolation: To avoid discontinuities between blocks, pixel values at the boundaries are inter polated with those of adjacent blocks, ensuring smooth transitions between the equalized tiles.

Upon processing all blocks, the resulting image displays enhanced local contrast without the artifacts commonly associated with global histogram equalization.

Figure 3 below are examples of images following the application of the pre-processing algorithm.

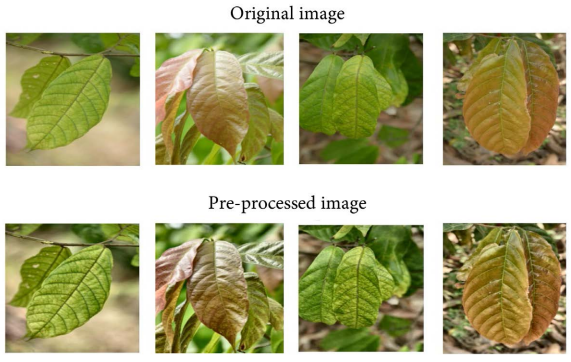

Figure 3. Example of pre-processed images.

3) Data Augmentation

To improve the model's generalization by exposing it to various perspectives of the original images, we applied data augmentation techniques. This was achieved by rotating the images at three angles: 90˚, 180˚, and 270˚, followed by flipping each of these rotated images (Figure 4). This approach effectively increases the size of our dataset by a factor of eight. The advantage of this technique lies in its ability to diversify the dataset while enhancing the model's resilience to noise. As a result, it improves the model's accuracy and sensitivity by optimizing feature detection and reducing biases.

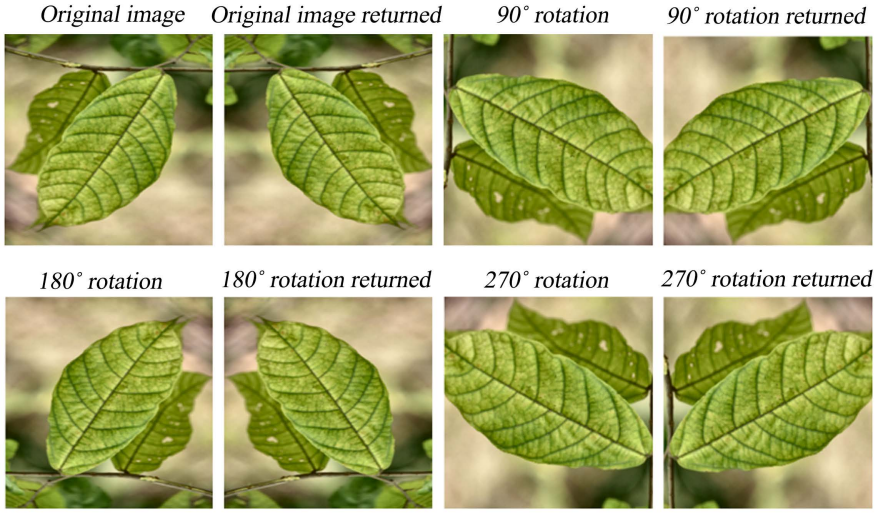

Figure 4. Example of augmented image.

Following the pre-processing steps, we present in Table 1 the proportions of the different categories of leaves in the dataset.

Table 1. Proportions of different categories of leaves.

Type of Leaves	Quantity	Percentages
Leaves of Healthy Plants	2700 Images	45%
Leaves Showing Symptoms	3300 Images	55%
Total	**6000 Images**	**100%**

4) Hardware

We selected Python as the programming language for implementing our project due to its simplicity, versatility, and the extensive range of libraries it offers. Python is also supported by most online platforms. For hardware, we used an HP All-in-One desktop PC equipped with an Intel(R) CORE I7-12700T 2.4 GHz processor, 16 GB RAM, a 1 TB SSD, and an NVIDIA Quadro P400 GPU. Additionally, we utilized the free Kaggle online platform for testing, which provides 16 GB of disk space, 16 GB RAM for the CPU, and 16 GB RAM for the GPU.

3.2. Methodology

We implemented three other algorithms in parallel using the pre-trained models VGG-19, Inception V3, and DenseNet-201.

1) CNN Algorithm used

a) VGG-19

The VGG-19 model is a deep convolutional neural network belonging to the VGG (Visual Geometry Group) family. Developed by the Visual Geometry Group at the University of Oxford, it consists of 19 convolutional layers using a 3×3 kernel. The model employs a nonlinear activation function and max-pooling for subsampling. Additionally, it includes two fully connected layers, each with 4096 nodes, followed by a Softmax layer for classification [32] [33].

b) Inception V3

Inception V3 is a widely adopted convolutional neural network architecture designed for classification tasks. The Inception V3 module forms the backbone of the GoogLeNet network. Typically, the Inception V3 module contains three different convolution sizes and a max-pooling operation. After performing convolutions on the output from the previous layer, channels are aggregated, and pooling of non-network elements occurs. Inception V3 introduces kernel factorization to break down larger convolutions into smaller ones, optimizing computation [34].

c) DenseNet-201

The Dense Convolutional Network (DenseNet) architecture includes four variants: DenseNet121, DenseNet169, DenseNet201, and DenseNet264. These networks use layers with 12 filters. In our study, we used DenseNet-201, where each layer has direct access to both the original input image and the gradients of the loss function. This architecture significantly reduces computational cost, making DenseNet-201 an excellent choice for image classification tasks [35].

2) Classifiers

a) Sigmoid Classifier

The sigmoid classifier employs the sigmoid function as an activation function within a model, commonly used in logistic regression or neural networks. The sigmoid function is mathematically defined by the following formula:

$$\xi = \frac{1}{1+e^{-x}} \tag{1}$$

This function takes an input value xxx and transforms it into an output ranging between 0 and 1, making it particularly suitable for binary classification tasks where the objective is to predict the probability of belonging to a specific class. In neural networks, the sigmoid function is often applied for several reasons:

- Nonlinear Activation: The sigmoid function introduces nonlinearity into the network, enabling it to model complex relationships between inputs and outputs.
- Bounded Output: The function's output lies within the range of 0 to 1, making it ideal for the final layer in neural networks used for binary classification.

- Gradient: The gradient of the sigmoid function is easy to compute, which simplifies the learning process via backpropagation.

However, the sigmoid function has certain limitations. It can lead to slow learning due to the very small gradients produced for large or small input values. This inefficiency, particularly as the function approaches 1 or 0, can hinder the learning process.

b) XGBOOST Classifier

XGBoost (Extreme Gradient Boosting) is an optimized machine learning library designed for implementing gradient boosting algorithms. Introduced by Tianqi Chen in 2014, it has become one of the most popular and efficient classifiers for both classification and regression tasks in machine learning [36]. XGBoost is recognized for the following:

- Performance: XGBoost is engineered to be extremely fast and efficient, leveraging techniques such as parallel decision tree optimization, effective memory management, and distributed computing. It enhances speed and performance through parallel processing and supports GPU utilization. Additionally, XGBoost employs the optimized DMatrix data structure to manage data efficiently, reducing memory consumption during training, and making it ideal for large datasets and computationally demanding tasks.
- Flexibility: XGBoost offers a variety of tunable parameters, supports multiple loss functions, and even allows the definition of custom loss functions, making it adaptable to a wide range of machine learning problems.
- Accuracy and Regularization: XGBoost incorporates regularization techniques to prevent overfitting, effectively handles missing data, and thus improves the generalization of models to new data. These features make XGBoost a powerful tool in machine learning, valued particularly for its speed, efficiency, and capacity to produce high-quality models.

c) SVM Classifier

The Support Vector Machine (SVM) is a supervised learning algorithm used for both classification and regression tasks. It seeks to maximize the margin between classes by finding an optimal hyperplane that separates them. The support vectors are the points closest to this hyperplane.

SVM uses kernel functions (linear, polynomial, RBF) to handle non-linear data by projecting it into a higher-dimensional space where separation becomes possible. The C parameter controls the trade-off between maximizing the margin and minimizing classification errors.

SVM is valued for its robustness and ability to generalize well, especially in high-dimensional spaces. However, it can be computationally expensive and often requires fine-tuning of parameters.

3) Evaluation Metrics

To evaluate the performance of the models in our study, we used several evaluation metrics, including accuracy, mean squared error (MSE), recall, F1 score, and the Matthews correlation coefficient (MCC), each calculated using the following formulas:

a) Accuracy

Accuracy is the simplest and most commonly used metric to evaluate the performance of deep learning algorithms. It represents the percentage of correct predictions out of the total predictions made on the selected sample.

$$\text{Accuracy} = \frac{\text{PC}}{\text{PT}} \tag{2}$$

b) Mean Squared Error (MSE)

Mean Squared Error measures the average of the squared differences between the predicted values and the actual values of the training data.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{3}$$

c) Recall

Recall assesses the true positive rate, which is the proportion of leaves showing symptoms that were correctly identified. It measures the model's ability to detect all infected leaves.

$$\text{Recall} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{4}$$

d) F1 Score

The F1 score is the weighted average of recall and precision, balancing both metrics to provide a single performance measure.

$$\text{F1 score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recalll}} \tag{5}$$

e) Matthews Correlation Coefficient (MCC)

MCC is used to evaluate the quality of binary classifications, offering a more balanced assessment than accuracy, especially in cases with imbalanced datasets.

$$\text{MCC} = \frac{\text{VP} * \text{VN} - \text{FP} * \text{FN}}{\sqrt{\left(\text{VP} + \text{FP}\right)\left(\text{VP} + \text{FN}\right)\left(\text{VN} + \text{FP}\right)\left(\text{VN} + \text{FN}\right)}} \tag{6}$$

where:

- PC = Number of Correct Predictions;
- PT = Total Number of Predictions;
- VP (True Positives) = correctly classified images with a true label;
- VN (True Negatives) = correctly classified images with a false label;
- FP (False Positives) = incorrectly classified images with a false label as positive;
- FN (False Negatives) = incorrectly classified images with a true label as negative;
- n = Total number of predictions in the evaluation dataset;
- y_i = True value for prediction i;
- $\hat{y}_i$ = Predicted value for prediction i.

These metrics were used to assess the overall performance and robustness of the implemented algorithms.

4) Implementation of the Algorithms

a) Architecture of the Algorithms

Our algorithm follows the general architecture of convolutional neural networks. It consists of a convolutional layer, a subsampling layer, a normalization layer, and a fully connected layer (e.g. Figure 5).

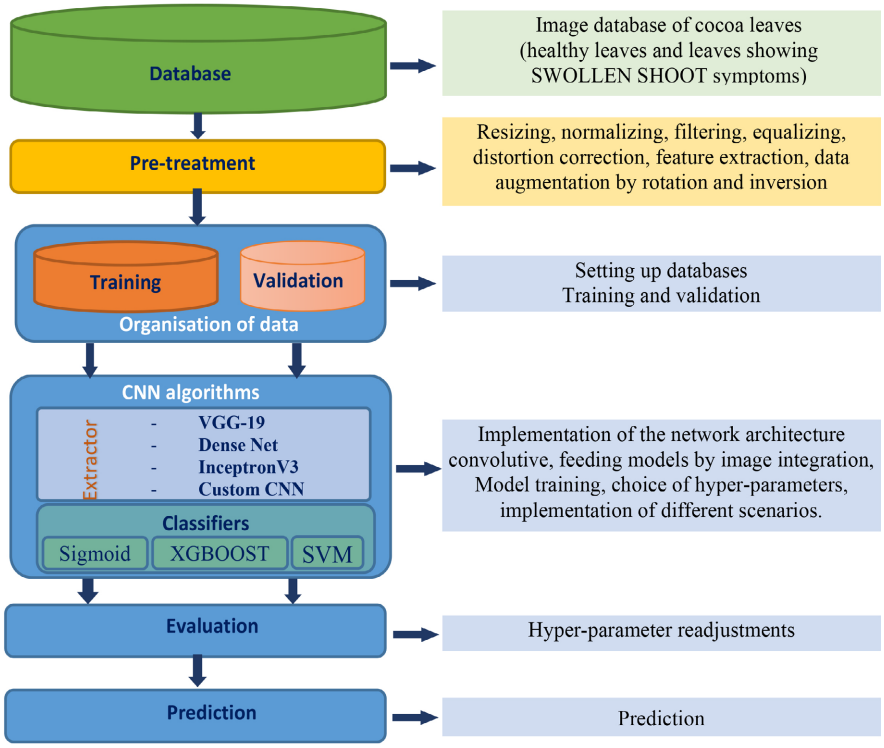

Figure 5. Methodology flowchart.

b) Hyper parameters used

We implemented this algorithm using the Python programming language with TensorFlow libraries, notably Keras. In the implementation, we used the following hyper parameters:

- Model Architecture: Sequential (Model structure is sequential);
- Number of Filters: 64, 256, 512 (Input layer, Intermediate layers, Output layers);
- Filter Size: (3, 3) (For each convolutional layer);
- Pooling: MaxPooling, size (2, 2) (Pooling window size);
- Activation Function: ReLU and Sigmoid (Input layer, Intermediate layers, Output layers);
- Batch Size: 128 (Number of samples processed before weight update);
- Number of Epochs: 10 (Number of times the algorithm passes through the training dataset);
- Number of Layers: 50 (Number of convolutional and fully connected layers);
- Dropout: 0.5 (To reduce overfitting);
- Optimizer: Adam, (Optimization algorithm for adjusting network weights).

4. Results and Analysis

To evaluate the performance of different algorithms and classifiers in detecting cocoa leaf diseases, we organized our tests into three scenarios.

4.1. Test Organization

1) Scenario 1

Scenario 1 involves detecting cocoa leaf diseases using feature extractors from VGG19, DenseNet, Inception V3, and Custom CNN with fully connected RNN layers as classifiers.

2) Scenario 2

Scenario 2 involves detecting cocoa leaf diseases using feature extractors from VGG19, DenseNet, Inception V3, and Custom CNN with XGBOOST classifiers.

3) Scenario 3

Scenario 3 involves detecting cocoa leaf diseases using feature extractors from VGG19, DenseNet, Inception V3, and Custom CNN with SVM classifiers.

4.2. Results of the Scenarios

This section presents all the experimental results obtained from our work, as well as a comparative study of the results from the VGG19, Dense Net, Inception V3, and Custom CNN algorithms with a sigmoid, XGBOOST and SVM classifier. For the three scenarios, we used the same dataset described earlier.

To highlight the differences and similarities between the three scenarios, we will use the standard metrics described above.

1) Results of scenario 1

Table 2 and Figure 6 below summarie the results of the tests conducted with the previously mentioned custom CNN, using the evaluation metrics discussed earlier for Scenario 1.

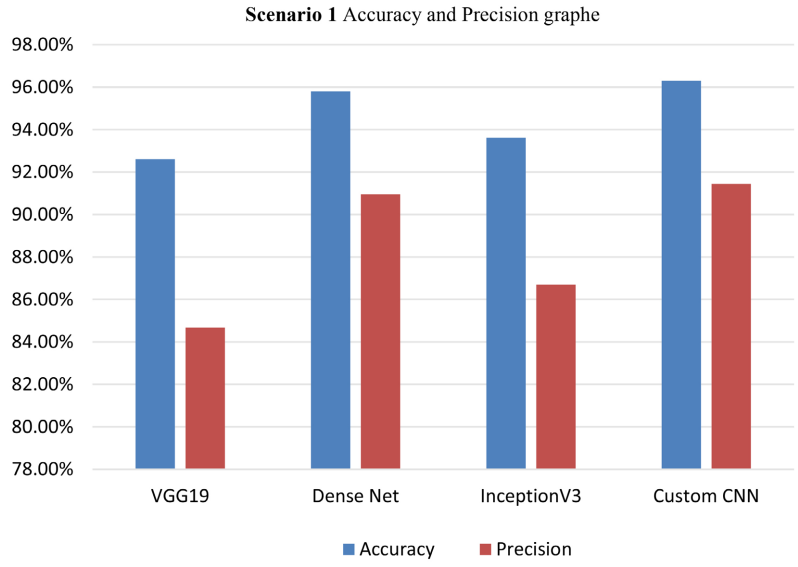

Figure 6. Performance of each deep learning model during the validation phases.

Table 2. Performance metrics for deep learning models.

Model	Accuracy	Precision	F1-score	MCC	Recall
VGG19	92.61%	84.67%	91.38%	85.80%	99.24%
Dense Net	95.80%	90.95%	94.92%	91.61%	99.24%
InceptionV3	93.61%	86.70%	92.43%	87.51%	98.99%
Custom CNN	96.30%	91.44%	95.53%	92.66%	100.00%

Table 2 presents the metric values collected during the validation and testing phases of the deep learning models used. We obtained accuracy values ranging from 92.61% to 96.30%, and recall values from 98.99% to 100%, demonstrating the overall predictive performance of the models and their ability to correctly classify both positive and negative cases. Precision values ranged from 84.67% to 91.44%, and F1-scores varied from 91.38% to 92.66%, confirming the balanced capability of the models to identify positive instances while maintaining a low false-positive rate.

In this scenario, the performance of the models, particularly VGG19, DenseNet, Inception V3, and a custom CNN with a sigmoid classifier, yielded very encouraging results. The best performance was achieved with our custom CNN.

2) Results of Scenario 2

Table 3 and Figure 7 below summarize the results of the tests conducted with the previously described Custom CNN, using the specified metrics for Scenario 2.

The four models were evaluated using the XGBOOST classifier, which significantly improved their performance as observed in Table 3. Accuracy values ranged from 95.94% to 99.10%, recall from 99.83% to 98.98%, precision from 90.78% to 97.77%, and F1-scores varied from 95.09% to 98.87%.

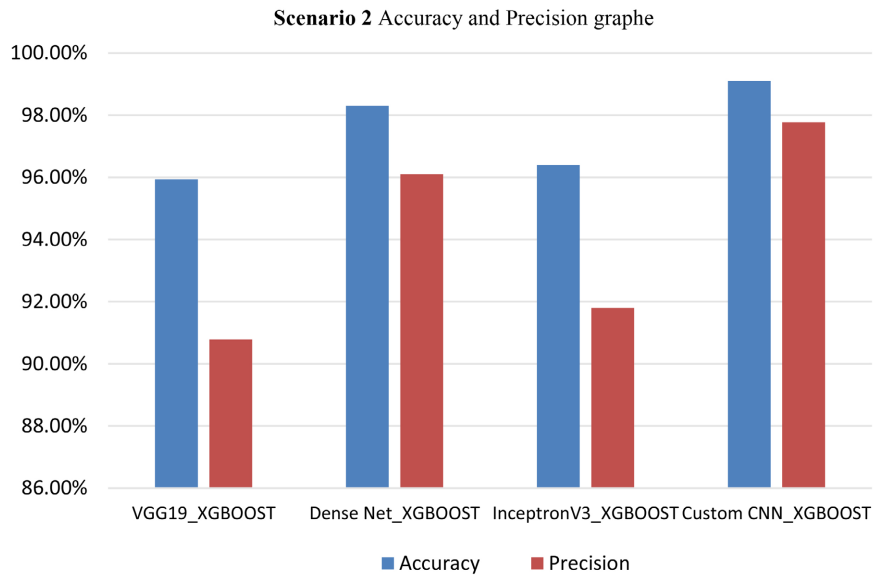

Figure 7. Performance of each deep learning model for validation phases.

Table 3. Performance measures for deep learning models.

Model	Accuracy	Precision	F1-score	MCC	Recall
VGG19_XGBOOST	95.94%	90.78%	95.09%	91.94%	99.83%
Dense Net_XGBOOST	98.30%	96.10%	97.89%	96.52%	99.89%
InceptronV3_XGBOOST	96.40%	91.80%	95.61%	92.80%	99.85%
Custom CNN_XGBOOST	99.10%	97.77%	98.87%	98.14%	99.98%

In this scenario, the performance of the four models significantly improved with the XGBOOST classifier. The best results were achieved with our custom CNN.

The graphs in Figure 7, below provide an overview of the performance (accuracy and loss) of each mode.

3) Results of Scenario 3

Table 4 and Figure 8 below summarize the results of the tests conducted with the previously described Custom CNN, using the specified metrics for Scenario 2.

Table 4. Performance measures for deep learning models.

Model	Accuracy	Precision	F1-score	MCC	Recall
VGG19_SVM	94.31%	90.92%	95.08%	88.87%	99.64%
Dense Net_ SVM	97.80%	96.86%	98.16%	94.07%	99.49%
InceptronV3_ SVM	95.61%	93.55%	96.26%	90.16%	99.12%
Custom CNN_ SVM	98.31%	97.85%	98.58%	99.12%	99.33%

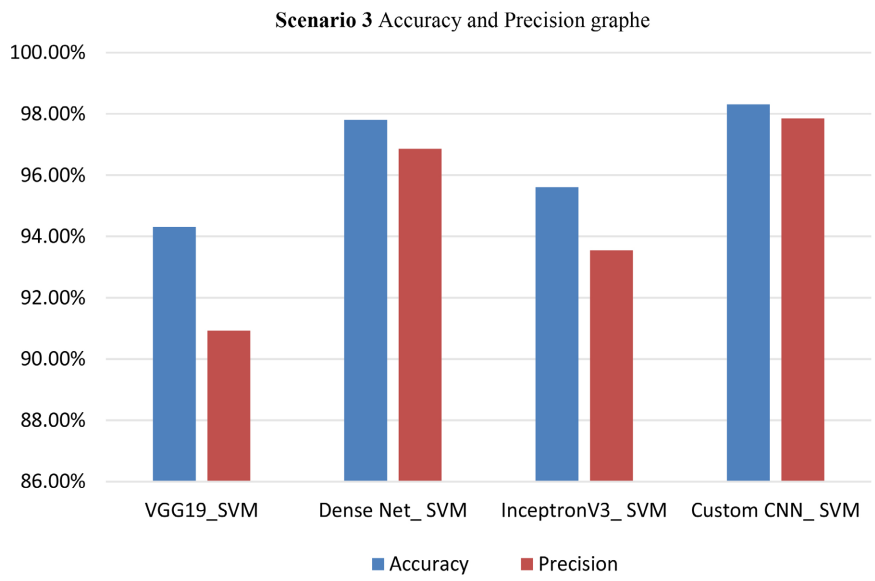

Figure 8. Performance of each deep learning model for validation phases.

In Scenario 3, the previous models were evaluated using the SVM classifier. We obtained accuracy values ranging from 94.31% to 98.31%, recall from 99.12% to 99.64%, while precision values ranged from 90.92% to 97.85%, and F1-scores varied from 95.08% to 98.58%. We observed a significant improvement in the results for all four models with the SVM classifier, although the performance remained below that of the XGBOOST classifier in Scenario 2. All four models showed better performance with both the SVM and XGBOOST classifiers.

The graphs in Figure 8 provide an overview of the performance (accuracy and loss) of each mode.

4) Results Analysis

The increase in accuracy based on the test data is presented in Table 5 below.

Table 5. Accuracy and precision variation.

Model	Scenario 1 Accuracy	Scenario 2 Accuracy	Scenario 3 Accuracy	Accuracy increase
VGG19	92.61%	95.94%	94.31%	3.47%
Dense Net	95.80%	98.30%	97.39%	2.54%
InceptionV3	93.61%	96.40%	95.57%	2.89%
Custom CNN	96.30%	99.10%	98.23%	2.83%

Table 5 above presents the variations in efficiency observed across the three scenarios during the validation and testing phases of the different models. We observed an Accuracy increase ranging from 2.54% to 3.47%. These values underscore the overall improvement in results in the second scenario.

All four models showed enhanced performance with both the SVM and XGBOOST classifiers.

However, the custom CNN stood out with an accuracy of 97.77% and a recall of 99.98%.

A comparative analysis of the three scenarios highlights the efficiency and precision gains brought by the use of XGBOOST in Scenario 2. These results illustrate the positive impact of integrating XGBOOST on model performance. The improved performance indicates a better ability of the models to generalize on validation and test datasets, reducing classification errors and increasing overall robustness.

Our study shows that the XGBOOST classifier is highly effective in classifying Swollen Shoot symptoms from cocoa leaf images.

5. Conclusions

In this study, we applied various deep learning methods to classify and recognize the Swollen Shoot disease using cocoa leaf images. The evaluation of these methods demonstrated the effectiveness of the models used in previous studies. However, our custom CNN, combined with the XGBOOST classifier, outperformed

the others, achieving a prediction accuracy of 99.10% for diseased leaves and an overall accuracy of 97.29%. The model also exhibited minimal loss, confirming its robustness. Our future goal is to apply this resilient approach in uncontrolled environments by integrating segmentation techniques to better target the affected areas of the leaves.

Our dataset was composed of two cocoa sub-varieties that are available as part of the Swollen Shoot control project. In our future work, we will expand this dataset to include other varieties cultivated in Brazil and globally. We also plan to leverage the advantages of new feature extraction and classification architectures to further enhance these results.

Acknowledgements

We would like to express our sincere thanks and gratitude to all the individuals who supported us throughout this study. We also extend our appreciation to the person in charge at Felix Houphouët-Boigny University.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Koffié, K. and Bi, K. (2011) Impact de la maladie virale du swollen shoot du cacaoyer sur la production de cacao en milieu paysan à bazré (côte d'ivoire).

[2] Lot, H., Djiekpor, E. and Jacquemond, M. (1991) Characterization of the Genome of Cacao Swollen Shoot Virus. *Journal of General Virology*, **72**, 1735-1739. https://doi.org/10.1099/0022-1317-72-7-1735

[3] Franck, O., Emmanuelle, M., François, B., Bernard, D., Komlan, W. and Christian, C. (2012) Analyse spatio-temporelle de la progression du Cacao Swollen Shoot Virus à l'échelle de parcelles cacaoyères.

[4] Kouassi, K.E. (2021) Menaces Sur L'Avenir Du Cacao Ivoirien. *Revue des Études Multidisciplinairers en Sciences Économiques et Socicles*, **6**, 103-120.

[5] Ofori, A., Padi, F.K., Ameyaw, G.A., Dadzie, A.M., Opoku-Agyeman, M., Domfeh, O., *et al.* (2022) Field Evaluation of the Impact of Cocoa Swollen Shoot Virus Disease Infection on Yield Traits of Different Cocoa (*Theobroma cacao* L.) Clones in Ghana. *PLOS ONE*, **17**, e0262461. https://doi.org/10.1371/journal.pone.0262461

[6] Herrbach, E.E., Maguet, J.L. and Hommay, G. (2012) Les cochenilles, des hémiptères peu connus comme vecteurs de virus. 8*th Conference of the Société Française de Phytopathologie*, Paris, 5-8 June 2012, 84.

[7] Couturier, G., Matile-Ferrero, D. and Richard, C. (1985) Sur les cochenilles de la région de Taï (Côte d'Ivoire), recensées dans les cultures et en forêt dense, (Homoptera, Coccoidea).

[8] Goebel, F. (2015) Bioagresseurs des cultures tropicales face au changement climatique: quelques exemples. https://www.academia.edu/16770599/Bioagresseurs_des_cultures_tropicales_face_au_changement_climatique_quelques_exemples

[9] Ayikpa, K.J., Mamadou, D., Gouton, P. and Adou, K.J. (2023) Classification of Cocoa

Pod Maturity Using Similarity Tools on an Image Database: Comparison of Feature Extractors and Color Spaces. *Data*, **8**, Article 99. https://doi.org/10.3390/data8060099

[10] Haralick, R.M., Shanmugam, K. and Dinstein, I. (1973) Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, **3**, 610-621. https://doi.org/10.1109/tsmc.1973.4309314

[11] d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G. and Sagun, L. (2022) Convit: Improving Vision Transformers with Soft Convolutional Inductive Biases. *Journal of Statistical Mechanics: Theory and Experiment*, **2022**, Article 114005. https://doi.org/10.1088/1742-5468/ac9830

[12] Sing Soh, K., Gubin Moung, E., John Julius Danker, K., Dargham, J.A. and Farzamnia, A. (2024) Cocoa Diseases Classification Using Deep Learning Algorithm. *ITM Web of Conferences*, **63**, Article 01014. https://doi.org/10.1051/itmconf/20246301014

[13] Gyamfi, A., Iddrisu, S.A. and Adegbola, O. (2020) Machine Learning-Based Cocoa E-Health System. 2020 13*th CMI Conference on Cybersecurity and Privacy* (*CMI*)—*Digital Transformation—Potentials and Challenges*(51275), Copenhagen, 26-27 November 2020, 1-7. https://doi.org/10.1109/cmi51275.2020.9322689

[14] Lomotey, R.K., Kumi, S., Orji, R. and Deters, R. (2023) Automatic Detection and Diagnosis of Cocoa Diseases Using Mobile Tech and Deep Learning. *International Journal of Sustainable Agricultural Management and Informatics*, **10**, 92-119.

[15] Coulibaly, M., Kouassi, K.H., Kolo, S. and Asseu, O. (2020) Detection of "Swollen Shoot" Disease in Ivorian Cocoa Trees via Convolutional Neural Networks. *Engineering*, **12**, 166-176. https://doi.org/10.4236/eng.2020.123014

[16] Ayikpa, K.J., Ayikpa, K.J., Ayikpa, K.J., Mamadou, D., Gouton, P. and Adou, K.J. (2022) Experimental Evaluation of Coffee Leaf Disease Classification and Recognition Based on Machine Learning and Deep Learning Algorithms. *Journal of Computer Science*, **18**, 1201-1212. https://doi.org/10.3844/jcssp.2022.1201.1212

[17] Bueno, G.E., Valenzuela, K.A. and Arboleda, E.R. (2020) Maturity Classification of Cacao through Spectrogram and Convolutional Neural Network. *Jurnal Teknologi dan Sistem Komputer*, **8**, 228-233. https://doi.org/10.14710/jtsiskom.2020.13733

[18] Kumi, S., Kelly, D., Woodstuff, J., Lomotey, R.K., Orji, R. and Deters, R. (2022) Cocoa Companion: Deep Learning-Based Smartphone Application for Cocoa Disease Detection. *Procedia Computer Science*, **203**, 87-94. https://doi.org/10.1016/j.procs.2022.07.013

[19] Buenaño Vera, D., Oviedo, B., Chiriboga Casanova, W. and Zambrano-Vega, C. (2024) Deep Learning-Based Computational Model for Disease Identification in Cocoa Pods (*Theobroma cacao* L.).

[20] Shukla, N. and Fricklas, K. (2018) Machine Learning with TensorFlow. Manning.

[21] Rodriguez, C., Alfaro, O., Paredes, P., Esenarro, D. and Hilario, F. (2021) Machine Learning Techniques in the Detection of Cocoa (*Theobroma cacao* L.) Diseases. *Annals of the Romanian Society for Cell Biology*, **25**, 7732-7741.

[22] Forsyth, D.A. and Ponce, J. (2002) Computer Vision: A Modern Approach. Prentice Hall Professional Technical Reference. Pearson.

[23] Zhang, G., Huang, X., Li, S.Z., Wang, Y. and Wu, X. (2004) Boosting Local Binary Pattern (LBP)-Based Face Recognition. In: *Lecture Notes in Computer Science*, Springer, 179-186. https://doi.org/10.1007/978-3-540-30548-4_21

[24] Noria, H. and Aniya, A. (2011) Application of Support Vector Machines (SVM) for Handwritten Digit Recognition.

[25] Bouillant, S., Mitéran, J., Yang, F. and Paindavoine, M. (2003) Détection de défauts

temps réel sur des objets à géométrie complexe: Étude par SVM et hyperrectangles. https://core.ac.uk/download/pdf/15494840.pdf

[26] Harvyanti, A.F.M., Baihaki, R.I., Dafik,, Ridlo, Z.R. and Agustin, I.H. (2023) Application of Convolutional Neural Network for Identifying Cocoa Leaf Disease. In: *Advances in Intelligent Systems Research*, Atlantis Press International BV, 283-304. https://doi.org/10.2991/978-94-6463-174-6_21

[27] Samuels, G.J., Ismaiel, A., Rosmana, A., Junaid, M., Guest, D., Mcmahon, P., *et al.* (2012) Vascular Streak Dieback of Cacao in Southeast Asia and Melanesia: In Planta Detection of the Pathogen and a New Taxonomy. *Fungal Biology*, **116**, 11-23. https://doi.org/10.1016/j.funbio.2011.07.009

[28] Deng, L. and Yu, D. (2014) Deep Learning: Methods and Applications. *Foundations and Trends in Signal Processing*, **7**, 197-387. https://doi.org/10.1561/2000000039

[29] Minu, M.S., *et al.* (2022) Optimal Squeeze Net with Deep Neural Network-Based Arial Image Classification Model in Unmanned Aerial Vehicles. *Traitement du Signal*, **39**, 275-281. https://doi.org/10.18280/ts.390128

[30] Reza, A.M. (2004) Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. *The Journal of VLSI Signal Processing-Systems for Signal, Image, and Video Technology*, **38**, 35-44. https://doi.org/10.1023/b:vlsi.0000028532.53893.82

[31] Paul, K.C. and Aslan, S. (2021) An Improved Real-Time Face Recognition System at Low Resolution Based on Local Binary Pattern Histogram Algorithm and CLAHE. *Optics and Photonics Journal*, **11**, 63-78. https://doi.org/10.4236/opj.2021.114005

[32] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. MIT Press.

[33] Mascarenhas, S. and Agarwal, M. (2021) A Comparison between VGG16, VGG19 and Resnet50 Architecture Frameworks for Image Classification. 2021 *International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications*, Bengaluru, 19-21 November 2021, 96-99. https://doi.org/10.1109/centcon52345.2021.9687944

[34] Himel, G.M.S., Islam, M.M. and Rahaman, M. (2024) Vision Intelligence for Smart Sheep Farming: Applying Ensemble Learning to Detect Sheep Breeds. *Artificial Intelligence in Agriculture*, **11**, 1-12. https://doi.org/10.1016/j.aiia.2023.11.002

[35] Team, K. (2023) Keras documentation: DenseNet, Keras Httpskeras Ioapiapplicationsdensenet.

[36] Wang, F., Tian, Y., Zhang, X. and Hu, F. (2022) An Ensemble of Xgboost Models for Detecting Disorders of Consciousness in Brain Injuries through EEG Connectivity. *Expert Systems with Applications*, **198**, Article 116778. https://doi.org/10.1016/j.eswa.2022.116778