Multimedia Tools and Applications (2023) 82:25083–25105
https://doi.org/10.1007/s11042-023-14369-2

# Comparative assessment of Pest damage identification of coconut plant using damage texture and color analysis

Utpal Barman [1] · Chhandanee Pathak [2] · Nirmal Kumar Mazumder [3]

Received: 11 February 2022 / Revised: 27 June 2022 / Accepted: 2 January 2023 /
Published online: 25 January 2023


## Abstract

Coconut cultivation is a promising agricultural activity. But to keep the coconut plants pest-free, the detection of various pest damage in coconut plants is of utmost importance for the cultivators. The processes that the cultivators use to detect pest damage in coconut plants are conventional methods, experts' views, or some laboratory techniques. But these procedures are not adequate in the detection of coconut damage identification. In this study, 16 different color and texture features are reported for 1265 coconut pest damage images by extracting the color and texture features of the damage images in the color and grey domain after the damage segmentation using the thresholding technique. The Gray Level Co-occurrence Matrix (GLCM) and Gray Level Run Length Matrix (GLRLM) techniques are applied to extract the texture features of the damages and two Artificial Neural Network (ANN) architectures are reported to classify the extracted data features of the damages into 5 different classes such as Eriophyid_Mite, Rhinoceros_Beetle, Red_Palm_Weevil, Rugose_Spiraling_White_fly, and Rugose_in_Mature with an average testing accuracy of almost 100% respectively. To compare the results with the other machine learning techniques, the Support Vector Machine(SVM), Decision Tree (DT), and Naïve Bayes (NB) are also introduced for damage identification where the SVM methods also report almost 100% accuracy on the fuse features of GLCM and GLRLM. The results of the ANN and SVM are

✉ Utpal Barman
utpalbelsor@gmail.com

Chhandanee Pathak
chhandaneepathak@gmail.com

Nirmal Kumar Mazumder
nmazumder66@gmail.com

[1] Department of CSE, The Assam Kaziranga University, Jorhat, Assam, India

[2] Department of CSE, GIMT, Guwahati, Assam, India

[3] Department of Plant Pathology, BN College of Agriculture, AAU, Biswanath Chariali, Assam, India

compared by finding the confusion matrix, precision, recall, and f-1 score of the ANN model with the DT and NB classifier. The ANN and SVM outperform in all matrices and they can be used as the base model for further study of coconut pest damage identification using deep learning techniques.

**Keywords** Coconut Pest · Image segmentation · GLCM · ANN · SVM · Decision tree · Naïve Bayes

## 1 Introduction

Coconut has its vivid versatility being used as a fruit, as a source of milk and oil, as a regular portion of the diets of many people in the tropics also in subtropics as seed nuts, as a fuel, and many more [5]**.** In coconut production, India is the third-largest producer in the World [16]**.** Like other palm trees, coconut plants are also vulnerable to various damages caused by pest infection. In laboratories early detection of coconut pests and diseases is possible. Also, the cultivators can get help from plant pathologists to detect the pest and diseases of the plant. But these methods are not convenient for the rural cultivators due to the unavailability of laboratories and experts in remote areas. Various computerized methods have been developed in recent years to identify and classify the diseases of plants using machine learning techniques [2]**.** Researchers used computer vision, machine learning, and deep learning techniques for plant pest and diseases detection such as Tomato [12], Potato [3], maize [4], and citrus [2] but the use of the machine and deep learning for coconut damage identification is very limited.

In the previous study of coconut plant, pests and diseases detection using machine and deep learning techniques were back propagation neural network, feed-forward neural network, and probabilistic neural network [6]. Along with these methods, image processing techniques such as morphological features extraction, wavelet-based image processing, zooming, and clipping with OSTU segmentation were also applied to identify coconut plant pests and diseases identification [9, 11]. In the paper [6], the Channel-spatial attention (CSA) and Region proposal network (RPN) was used to distinguish between the pest affected and normal regions. The SoC with ARM Cortex-based CPU and GPU were embedded in the drone for processing the images. Hence no requirement for pre-processing. Due to the SoC and fast interpreting algorithm, processing was done. [16] emphasized on detection of stem bleeding, leaf blight, and red palm weevil pest infection in coconut trees by applying the k-mean clustering segmentation technique along with MobileNet and customized the 2D-CNN model automatically. Nesarajan et al., (2020) reported the application of SVM for color and shape-based coconut disease detection in their study but the CNN model, reported by Nesarajan et al., (2020) was EfficientNetB0. Along with the plant disease detection, the application of CNN and hand-crafted features were already reported in the case of human disease identification such as skin cancer identification [14] and Covid-19 indentification [18]. Apart from disease identification, deep learning is massively used in other areas including stereo matching [19], and person indetification [13].

In the case of using drones mentioned above operational safety, privacy and insurance protection are some of the concerns. While applying CNN, the larger requirement of image datasets is another hurdle. So, identification of pest infection of coconut for a small image dataset is a very challenging task. In this paper, machine learning techniques such as ANN, SVM, DT, and NB were used to identify the different pest infections of coconut. Therefore, the objective of this work is to the application of efficient machine learning techniques to identify and classify the infected area in the images and to develop an optimized machine learning model to locate the infection from the segmented images to make the model more robust. The paper contributes the following towards the pest identification of coconut.

i) The coconut pest infection images are captured using a DSLR camera in a natural environment.
ii) The color and texture features of the images are extracted from the coconut images to handle the small dataset.
iii) The ANN, SVM, DT, and NB models are exported to identify pest damage infection.
iv) The ANN and SVM model outperform and are considered a robust methods for coconut pest infection.

## 2 Materials and methods

### 2.1 Overview of the proposed work

The proposed work for classifying coconut pest damage identification was carried out in six phases. The pictorial representation of the proposed method is presented in Fig. 1.

- Step 1: Creation of coconut pest infection image dataset.
- Step 2: Data augmentation and preprocessing.
- Step 3: Color and Feature Extraction of the pest infection image dataset.
- Step 4: Pest identification using ANN.
- Step 5: Pest identification using SVM, DT, and NB.
- Step 6: Choosing the model with the best performance.
- Step 7: Assessment and evaluation of the model.

### 2.2 Creation of pest infection image dataset

The process of image acquisition is to obtain various images from the samples using digital image sensing gadgets such as digital cameras, smartphones, DSLRs, etc. For this study, coconut plant growing in the Assam region was chosen. The datasets were collected from the different districts of Assam. A comparison study was conducted to identify the disease plants from the healthier plant-bearing of coconut. The image datasets for the study were collected through an image captured by the digital camera (Canon PowerShot A590 IS) with an image quality set as 3.8 MP with 6000*4000 resolution. The file size of images taken with these settings varied between 1.4 MB to 12.3 MB. For the variation in image size, the average dimensions of the images were 5568*3712 and saved in JPEG format by keeping it in 8-bit RGB mode.

### 2.3 Image augmentation and preprocessing of coconut plant images

The images of various parts of a coconut plant present in the datasets were augmented and preprocessed using the following steps.


**Fig. 1** Sample of coconut pest infection

A. The augmentation techniques are required to increase the number of pest infection images in the dataset. Five different image dataset classes with the number of images are presented in Table 1. To make the dataset balanced and reduced overfitting during training and testing, the following augmentation techniques were applied to the dataset.

   a. The images were rescaled by a factor up to 1/255.
   b. The images were randomly flipped in the horizontal as well as sheared in the counter-clockwise direction up to 0.2 degrees.
   c. Brightness level was set to be in the range of (0.5,1.5).

Thus, for the 5-classes of image datasets, the Image-Data-Generator class generated images which increased the dataset to a collection of approximately 1265 images from 448 images. In

**Table 1** Total numbers of images in coconut pest infection image dataset

| Sl no | Image dataset class | Number of images |
|---|---|---|
| 1 | Eriophyid Mite | 58 |
| 2 | Rhinoceros Beetle | 93 |
| 3 | Red Palm Weevil | 72 |
| 4 | Rugose Spiraling White fly | 87 |
| 5 | Rugose in Mature plants | 138 |

Fig. 2. a few images are presented after applying the data augmentation technique to the original coconut pest-infected image datasets.

The following Table 2 represents the number of augmented images of each class of the pest infection.

B. Preprocessing

i) The images were converted into grayscale.

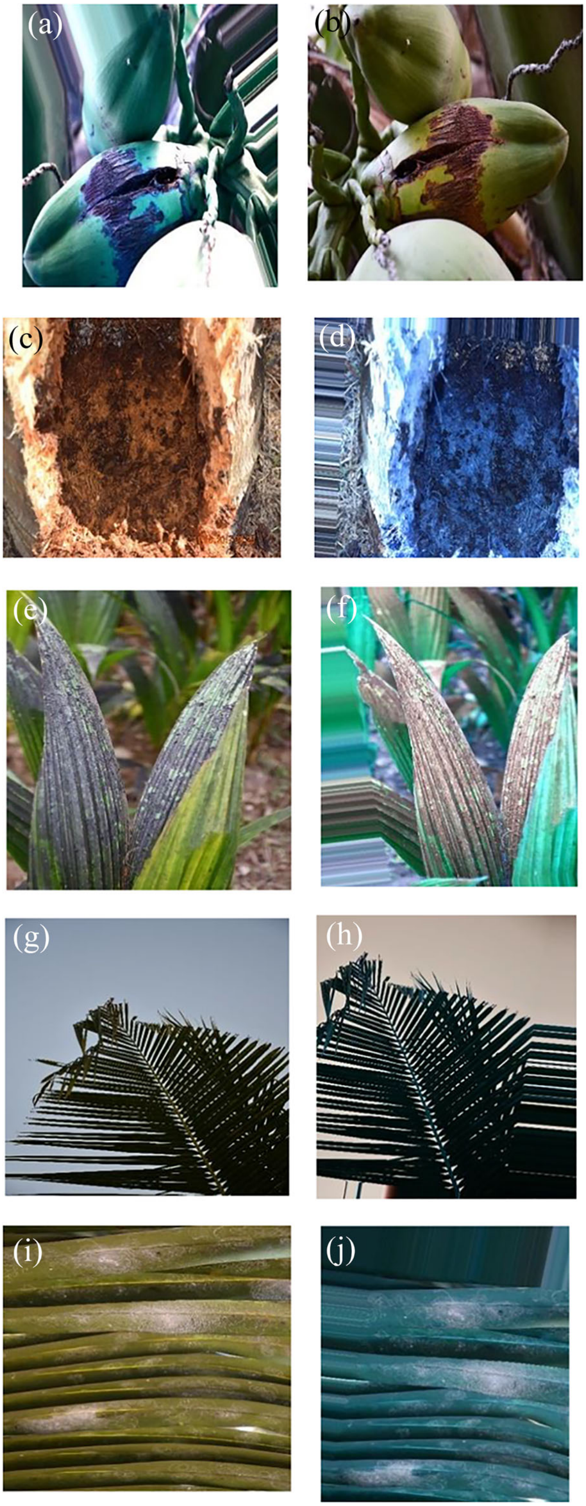
Images may contain noise due to the camera sensor and the process of removing the noise is known as image smoothing or blurring. Image denoising and image restoration were the two basic features for image processing and are used as a measure of noise removal from the image and as a restoration of images [7]. In this paper, a Gaussian filter was used for image blurring because no overshoot was present in the Gaussian filter to minimize the rise and fall time in the step function. The sharp edges of the coconut pest infection images were smoothed during the minimization of too much blurring.

ii) In this paper the thresholding, the simplest image segmentation was used to segment the grayscale pest infection image into a binary format using the cv2.THRESH_BINARY_INV library of open CV in python.
iii) In the next step, a kernel of size 50 × 50 was used to find the dilation followed by erosion morphology of the pest infection image using the cv2.morphologyEx.

III. **Important features of the coconut plant**

Color and texture extraction are the two main features of the Image Segmentation technique to locate the infected area in an image of plant disease. The experiment was carried out implementing in Python as a source code in Jupyter Notebook. It was observed that the color and texture of the pest-infected parts of the coconut plant were changing. All these characteristics, i.e., color and texture features of the image can be extracted using color and texture extracting techniques of computer vision. The Gray Level Co-occurrence Matrix (GLCM) is an advanced and special feature extraction technique that was applied to image datasets to describe the color and texture properties of the image dataset. Both color and texture features were described in the color and gray domain of the images. Before the texture extraction, pest-infected coconut images were converted from BGR to RGB format using an open cv library. To find the texture feature of the images, RGB images were again converted to grayscale. In this work, total 11 numbers of features of coconut images were calculated in two domains. In the gray domain, contrast, correlation, dissimilarity, energy, and homogeneity were determined for texture extraction. Along with these parameters, the class label is shown as one of the parameters for the execution (Table 3). Later on, color mean and standard deviation for RGB were calculated as color feature extraction (Table 3).
The feature extraction algorithm is as follows.

Step 1: Convert the color image into RGB format.
    Step 1.1: Evaluate the color mean features of Red, Green, and Blue.
    Step 1.2: Evaluate the Standard Deviation of Red, Green, and Blue.


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)

◀ **Fig. 2** (**a**)-(**b**) normal image v/s contrast-enhanced images of Eryophite mite infected part (**c**)-(**d**) normal image v/s contrast-enhanced images of Red Palm weevil infected part (**e**)-(**f**) normal image v/s contrast-enhanced images of Rugose spiraling whitefly infected part (**g**)-(**h**)normal image v/s contrast-enhanced images of Rhicerous beetle leaf symptoms (**i**)-(**j**) normal image v/s contrast-enhanced images of Rugose in mature plants damage.

Step 2: Convert the RGB image into a gray format.

Step 3: Evaluate the GLCM Texture Feature.

Step 3.3: Evaluate the texture feature from the gray image i.e., Energy, Correlation, Dissimilarity, Homogeneity, and Contrast.

Step 4: Evaluate the final feature vector by taking 6 color features and 5 texture features from the GLCM denotation.

In Table 3, m (i, j) is the intensity value of the pixel and N is the gray level. After calculating the GLCM parameters for the five classes of coconut image datasets, the results of the .csv files generated are represented with the help of line charts. (Figure 3a-b).

Along with the GLCM, the Gray Level Run Length Matrix (GLRLM) is also reported in the paper for texture feature extraction. It is a matrix from which higher-order statistical texture features can be extracted for texture analysis [20]. A gray level run can be defined as a line of pixels with the same intensity value in a particular direction. The number of pixels describes the gray level run length and the number of occurrences defines the run-length value. Five texture features can be extracted using GLRLM. The features are Short Run Emphasis (SRE), Long Run Emphasis (LRE), Grey level Uniformity (GRU), Run Length Uniformity (RLU), and Run percentage (RP) were calculated using the formula presented in Table 4. However, from the performance and accuracy perspective, both the feature extractors (GLCM and GLRLM) are efficient enough and in this paper, both the techniques are applied to extract the features using the equations (Table 4) and accordingly ANN, SVM, DT, and NB were applied in both the data features.

In the equation, the Pij is the ij$^{th}$ entry of the GLRLM. The $N_r$ reflects the run length of the image and Ng is the different gray level of the image.

### 2.4 Data standardization of color and texture features of the coconut pest damage images

Data standardization is the process of bringing different variables to the same or uniform scale on account of comparing values between different types of variables [4]. It is performed to compare different variables.

From the above feature charts (Fig. 3) for color and texture features extracted from 5-classes of coconut pest damaged image dataset, it was found that the values for color features (mean

**Table 2** Number of image datasets before and after applying Data Augmentation

| Sl. no. | Name of the Class | No of images | New Augmented images |
|---|---|---|---|
| 1 | Eryophite Mite damage | 58 | 186 |
| 2 | Red Palm Weevil damage | 72 | 264 |
| 3 | Rugose Spiraling White fly damage | 87 | 244 |
| 4 | Rhicerous Beetle leaf symptoms | 93 | 253 |
| 5 | Rugose in Mature plants damage | 138 | 318 |

**Table 3** GLCM parameters and their related equations

| Eq. no | Parameter | Equation |
|---|---|---|
| 1 | Energy | $\sum_{i,j=0}^{N-1} m(i,j)^2$ |
| 2 | Correlation | $\sum_{i,j=0}^{N-1} m_{i,j} \frac{((i-\mu)(j-\mu))}{\sigma^2}$ |
| 3 | Dissimilarity | $\sum_i 1 \sum_j \mid i-j \mid m(i,j)$ |
| 4 | Homogeneity | $\sum_{i,j=0}^{N-1} \frac{m(i,j)}{1+(i-j)^2}$ |
| 5 | Contrast | $\sum_{i,j=0}^{N-1} (i,j)^2 m(i,j)$ |
| 6 | Mean | $\sum_{i,j=0}^{N-1} i.m_{i,j}$ |
| 7 | Standard Deviation | $\sum_{i,j=0}^{N-1} (i-Mean)^2 m(i,j)^{\frac{1}{2}}$ |

and standard deviation of RGB) were comparatively higher than the values for texture features (Energy, Correlation, Dissimilarity, Homogeneity, Contrast, SRE, LRE, GRU, RLU, and RP). Because of this higher value of the color feature, it was difficult to compare and plot with the texture feature values which might over-impose and impact the overall result of the classification process.

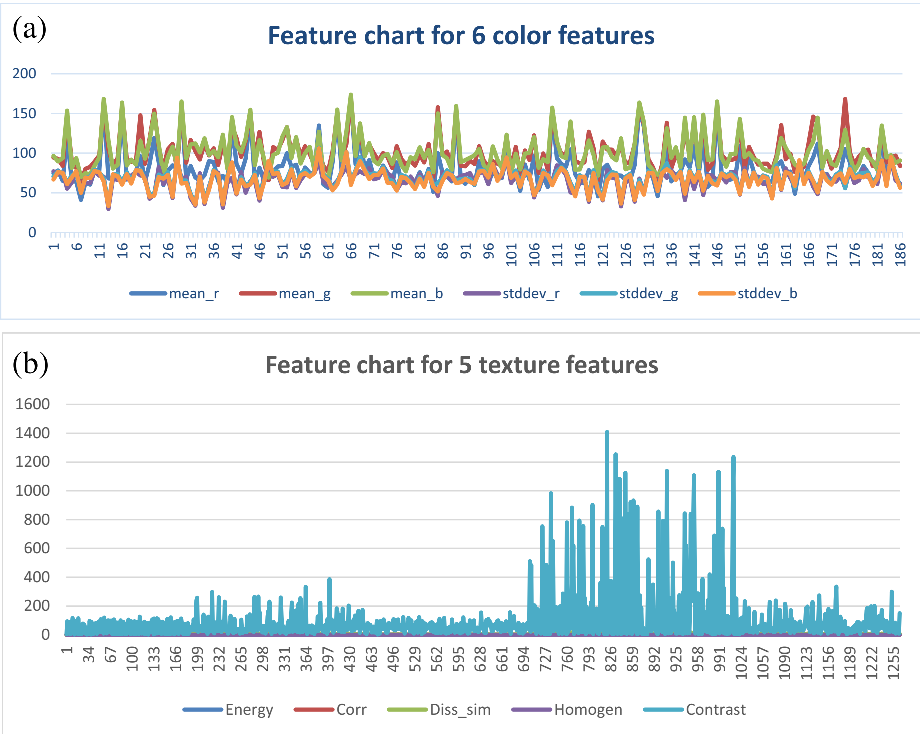



**Fig. 3** (**a**)-(**b**) Features Chart of color and texture features

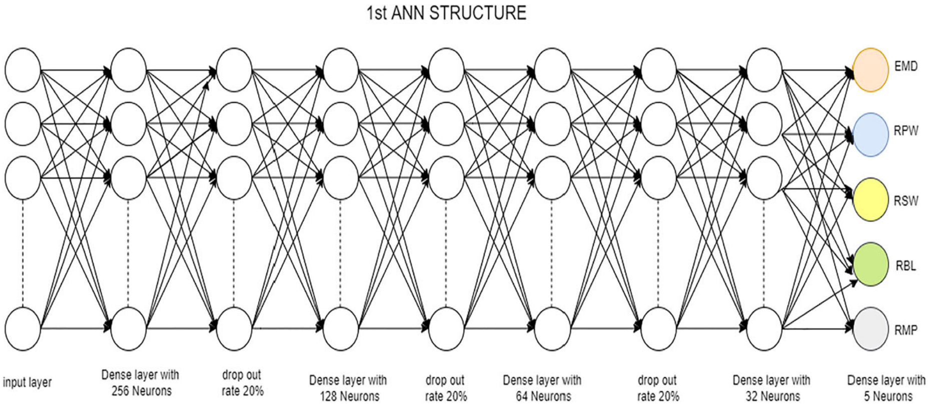



**Fig. 4** 1st ANN Structure to classify the pest damage of Coconut

To overcome the situation of overfitting and to maintain the balance between the two feature values, the StandardScaler library of Python was used in this work. StandardScaler was used to resize the color and texture distribution values to a unit variance scale by eliminating the mean [8]. The standardization was done independently of each of the feature values. It was performed as a preprocessing step before applying any machine learning models to standardize the range of variables of the input datasets.

As mentioned above, color and texture features differ hugely between their ranges, we were using this scaler library so that both the features contributed equally to the output as well as the performance of the machine learning models.

## 2.5 Pest damage identification using ANN

In the present study, Artificial Neural Network (ANN) was used to classify different pest damage identification on coconut leaves, stems, and fruit. Two ANN structures were applied in the study work to get a clear visual classification. Both ANN structures were considered by applying 4 dense (hidden) layers.

In the first ANN structure, the first dense layer received 11 input feature parameters and it contained 256 hidden neurons. The second hidden layer and third hidden layer contained 128

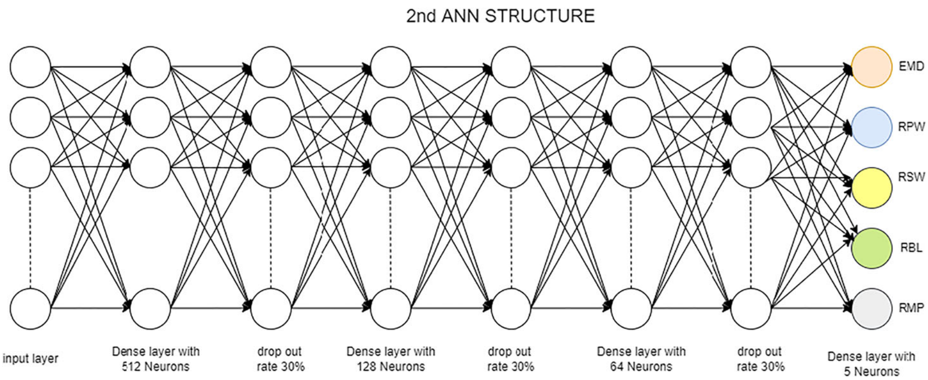



**Fig. 5** 2nd ANN structure to classify the pest damage of Coconut

**Table 4** GLRLM parameters and their related equations

| Eq. no | Parameter | Equation |
|---|---|---|
| 1 | Grey level Uniformity (GRU) | $\sum_{i=Ng}\left(\sum_{j=Nr} Pij\right)^2 / \sum_{i=Ng}\sum_{j=Nr} Pij$ |
| 2 | Long Run Emphasis (LRE), | $\sum_{i=Ng}\sum_{j=Nr} j2.Pij / \sum_{i=Ng}\sum_{j=Nr} Pij$ |
| 3 | Short Run Emphasis (SRE) | $\sum_{i=Ng}\sum_{j=Nr} \frac{Pij}{j2} / \sum_{i=Ng}\sum_{j=Nr} Pij$ |
| 4 | Run Length Uniformity (RLU) | $\sum_{i=Nr}\left(\sum_{j=Ng} Pij\right)^2 / \sum_{i=Ng}\sum_{j=Nr} Pij$ |
| 5 | Run percentage (RP) | $\sum_{i=Ng}\sum_{j=Nr} Pij/\mathrm{N}$ |

and 64 hidden neurons respectively. The fourth hidden layer contained 32 hidden neurons. The final dense layer contained 5 hidden neurons with a SoftMax activation function for the classification of pest damage identification. Besides the final dense layer, a relu activation function was added to each of the hidden layers to get the desired output.

To reduce the overfitting problem, three dropouts with a dropout rate of 0.2 were applied to the ANN model. To the second ANN structure, along with the 11-input feature parameter, 512 hidden neurons were added to the first dense layer. The second and third hidden layers received 128 and 64 hidden neurons respectively. The final dense layer contained 5 hidden neurons for the classification purpose. Same as the first ANN structure, a relu activation function was added to each of the hidden layers to get the desired output except for the final dense layer. Three dropouts with 0.3 as a dropout value were applied to get rid of the overfitting issue.

With a learning rate of 0.001, batch size = 10, and epoch = 20, the two ANN models were optimized with the ADAM optimizer. The loss of both ANN models was evaluated using the sparse categorical cross-entropy loss. For the better evaluation of the both ANN model, another 20% of the training feature in the first ANN structure and 30% of the training feature in the second ANN structure were used for validation (Table 5).

## 2.6 Pest damage identification using SVM, DT and Naïve Bayes classifiers

To compare the results with the other machine learning techniques, the Support Vector Machine (SVM), Decision Tree (DT), and Naïve Bayes (NB) classifiers were also introduced for damage identification of coconut plant.

Support Vector Machine (SVM) is a supervised machine learning algorithm that is used for classification problems [1]. After applying two different ANN structures to classify the pest damaged images of coconut plant, SVM with two of the kernels Radiant Basis Function (RBF) and Polynomial (poly) were applied to transform the input datasets into the required form.

**Table 5** ANN structure for training and testing dataset

| Sl no | ANN structure | Total dataset | Training dataset | Testing dataset | Validation set |
|---|---|---|---|---|---|
| 1 | 1st | 1265 | 810 | 253 | 202 |
| 2 | 2nd | 1265 | 619 | 380 | 266 |

i. The purpose of using the SVM-RBF kernel was due to transform the non-linear information coconut dataset. Apart from that, it overcomes the space complexity issue as the kernel requires storing the support vectors during training rather than the entire dataset [17].
ii. The SVM-POLY kernel was also introduced in our study work. The kernel added polynomial feature combinations to the existing coconut datasets to create more features.

The gamma parameter in the SVM kernel defines the extension of the influence of a single training dataset, with low values mean ‘far’ and high values mean ‘close’. In this study work for both the RBF and POLY kernel, the gamma value was taken as ‘scale’ which was the default value of the parameter indicating gamma = 1.

Decision Tree (DT) is a supervised learning technique; mostly it is preferred for solving classification problems which can also be used for regression problems. In this study, the DT was using the DecisionTreeClassifier python library by considering the random as 42. Along with the DT, the Gaussian model of the Naïve Bayes algorithm was also reported which assumed that features (color and texture) follow a normal distribution. The classifier assumed that the value of the features is independent of each other. The Naïve Bayes classifier requires a small training dataset to estimate the parameters needed for classification. As the image dataset of this study is not so very large, the Gaussian Naïve Bayes classifier is a good choice for classification purposes.

## 3 Results and discussion

### 3.1 Environmental set-up

All experiments were conducted in a Python 3 environment on the Keras and TensorFlow frameworks. Experiments were conducted in an Intel(R) Core (TM) i3-7020U CPU @ 2.30GHz for all the 5-classes of image datasets.

### 3.2 Metrics used to measure the performance of the present work

The different metrics were used in this study to measure the performance of the present work i.e., the two ANN models, SVM, DT, and NB. The lower the loss in the training, testing, and validation, the higher the accuracy in all three categories is the best model for the classification of pest infection. The parameters are as follows:

I. Precision: Precision determines how accurate the model is out of those predicted positive, how many of them are positive [15].
II. Recall: Recall calculates how many of the actual positives we record through labeling it as positive (True Positive) by the proposed models [15]
III. F1 score: The F1 score is a function of precision and recall which is used to check the balance between the two parameters and combine them into one value [10]
IV. Accuracy: It is the model’s ability to measure the accurate measurements to a specific value.
V. Confusion matrix: A Confusion matrix is for evaluating the performance of a classification model which is defined by an N x N matrix, where N is the number of target classes. The different machine learning models compare the actual target values with the predicted values using Confusion Matrix.

### 3.3 Results of ANN models

At first, the classification of pest damage identification in various parts of the coconut plant was done using two ANN structures on GLCM features (Table 6 and Table 7), and then the best ANN model is applied to GLRLM features and GLCM + GLRLM features (Tables 8 and 9). The results were compared for evaluating the two different ANN structures. Both ANN models were optimized with the ADAM optimizer and trained with a learning rate of 0.001 for 20 epochs. Along with the tabular representation (Tables 6, 7), the graphical representation of the results of the two ANN structures is presented in Fig. 6 and Fig. 7. Figure 8, Fig. 9, and Tables 8, 9, and 10 depict the ANN1 results on GLCM and GLCM + GLRLM texture features.

The observations of ANN models on GLCM features are presented below:

a) Initially, both the ANN models showed under-fitting (Table 5), and their training accuracies were very less such as 60% in epoch 1 of ANN1 and 69% in epoch 1 of ANN2. With the increase of the epoch, the model trained more and more parameters and the under-fitting was reduced in both cases. At epoch 20, the training accuracy of the first ANN model was 97% whereas the validation accuracy was 97%. It means that the model was neither over-fitted nor under-fitted during the training and the validation process. Like the first ANN, ANN 2 also reported 97% and 98% accuracy in the training and validation process, respectively. The training accuracies of the ANN 1 and ANN 2 models were 99% and 98%.
b) By observing both the ANN models (Table 6), it was found that the precision value of the 1st ANN model, for the classes 0, 1, 2, 3, and 4 were 1, 1, 0.98, 0.98 and 1, and the 2nd ANN model had 0.94, 1, 0.98, 0.99, and 1, respectively. With recall values of 1, 1, 1, 0.98, and 0.98 for the 1st ANN model whereas the 2nd ANN model has recall values of 0.96, 1, 0.99, 0.97, and 1 for the five respective classes. The 1st ANN structure was satisfactory with an accuracy of 99% while the 2nd has an accuracy of 98%.
c) By observing the confusion matrix for the ANN models, it is reported that 1st ANN model predicted the entire 35 pest damaged images correctly for class 0, it correctly predicted all the 48 images for class 1 as well as it classified 59 images correctly for class 2. For class 3, out of 61 images, 60 images were predicted correctly and 1 image was misclassified as class2. In the classification for the 4th class, it is found that out of 50 images; only 1 image was misclassified as class 3. The other 49 images were predicted correctly by the classifier.

**Table 6** Accuracy and Losses of the Training, Validation, and Testing of ANN1 and ANN2 on GLCM Features

| Structure | Epoch | Training Accuracy | Training Loss | Validation Accuracy | Validation Loss | Testing Accuracy | Testing Loss |
|---|---|---|---|---|---|---|---|
| ANN 1 | 1 | 0.6047 | 1.0048 | 0.977 | 0.0928 | 0.9368 | 0.1743 |
| | 5 | 0.9042 | 0.2557 | 0.973 | 0.0799 | 0.9960 | 0.0269 |
| | 10 | 0.9466 | 0.1346 | 0.977 | 0.0548 | 0.9960 | 0.0170 |
| | 15 | 0.9575 | 0.1242 | 0.977 | 0.0592 | 0.9960 | 0.0211 |
| | 20 | 0.9723 | 0.0970 | 0.973 | 0.0682 | 0.9921 | 0.0127 |
| ANN 2 | 1 | 0.6384 | 0.9456 | 0.973 | 0.0726 | 0.9368 | 0.2191 |
| | 5 | 0.8983 | 0.2845 | 0.981 | 0.0483 | 0.9684 | 0.0931 |
| | 10 | 0.9424 | 0.1720 | 0.985 | 0.0448 | 0.9553 | 0.1803 |
| | 15 | 0.9571 | 0.1395 | 0.977 | 0.0678 | 0.9789 | 0.0654 |
| | 20 | 0.9751 | 0.0848 | 0.985 | 0.0505 | 0.9895 | 0.0334 |

**Table 7** Model performance of 1st and 2nd ANN on GLCM Features

| Model | Class | Precision | Recall | F1-score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| 1st ANN | 0 | 1 | 1 | 0.95 | 35 | [[35 0 0 0 0] | 99% |
| | 1 | 1 | 1 | 1 | 48 | [0 48 0 0 0] | |
| | 2 | 0.98 | 1 | 0.99 | 59 | [0 0 59 0 0] | |
| | 3 | 0.98 | 0.98 | 0.98 | 61 | [0 0 16 0 0] | |
| | 4 | 1 | 0.98 | 0.99 | 50 | [0 0 0 1 49]] | |
| 2nd ANN | 0 | 0.94 | 0.96 | 1 | 52 | [[50 0 1 10] | 98% |
| | 1 | 1 | 1 | 1 | 76 | [0 76 0 0 0] | |
| | 2 | 0.98 | 0.99 | 0.98 | 86 | [1 0 85 0 0] | |
| | 3 | 0.99 | 0.97 | 0.98 | 91 | [2 0 1 88 0] | |
| | 4 | 1 | 1 | 1 | 75 | [0 0 0 0 75]] | |

The 2nd ANN model reported 50 pest-damaged images positively for class 0 and predicted 76 images correctly for class 1. For class 2, the model predicted 85 images out of 86 correctly while misclassifying 1 image as class 1. In the same way, 3 out of 91 images were misclassified and 88 images were predicted correctly for class 3. For class 4; all the 75 images were predicted and classified correctly.

The observations of the best ANN model (ANN1) on GLRLM features are presented below:

a) From starting of training, the ANN1 model reported positive results on GLRLM data features except epoch 1. At epoch 20, the training, validation, and testing accuracy of the ANN1 model were almost 100% respectively (Table 8).

**Table 8** Accuracy and Losses of the Training, Validation, and Testing of best ANN (ANN1) on GLRLM Features

| Structure | Epoch | Training Accuracy | Training Loss | Validation Accuracy | Validation Loss | Testing Accuracy | Testing Loss |
|---|---|---|---|---|---|---|---|
| ANN 1 | 1 | 0.7352 | 0.6433 | 0.9972 | 0.0162 | 0.9980 | 0.0080 |
| | 5 | 0.9822 | 0.0661 | 0.9986 | 0.0012 | 0.9960 | 0.0026 |
| | 10 | 0.9921 | 0.0260 | 1.000 | 0.0005 | 0.9921 | 0.0178 |
| | 15 | 0.9970 | 0.0111 | 1.000 | 0.0008 | 0.9723 | 0.1075 |
| | 20 | 0.9980 | 0.0076 | 0.9986 | 0.0032 | 0.9992 | 0.0029 |

**Table 9** Accuracy and Losses of the Training, Validation, and Testing of best ANN (ANN1) on GLRLM + GLCM Features

| Structure | Epoch | Training Accuracy | Training Loss | Validation Accuracy | Validation Loss | Testing Accuracy | Testing Loss |
|---|---|---|---|---|---|---|---|
| ANN 1 | 1 | 0.8073 | 0.0531 | 0.9944 | 0.0195 | 0.9960 | 0.0100 |
| | 5 | 0.9852 | 0.0452 | 0.9986 | 0.0031 | 0.9960 | 0.0070 |
| | 10 | 0.9970 | 0.0077 | 0.9944 | 0.0144 | 1.0000 | 0.0001 |
| | 15 | 0.9941 | 0.0163 | 0.9972 | 0.0062 | 1.0000 | 0.0024 |
| | 20 | 0.9993 | 0.0008 | 1.0000 | 0.0001 | 0.9960 | 0.0079 |

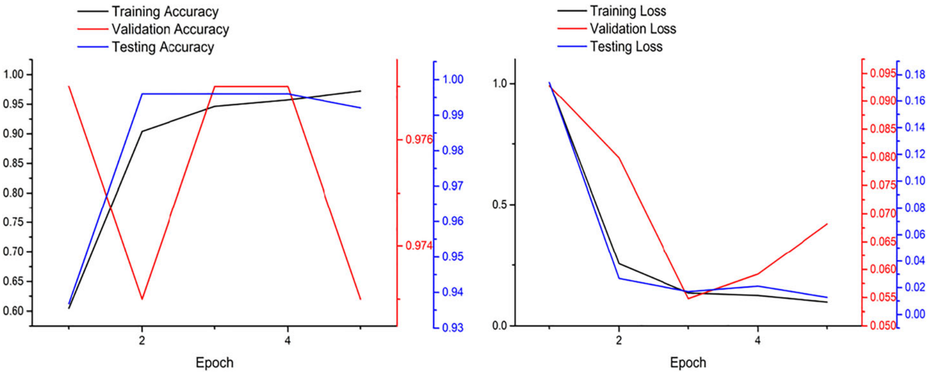



**Fig. 6** Train and validation accuracy and Loss of the 1st ANN structure on GLCM data feature

b) By observing the confusing matrix of the ANN-1 model (Table 10), it was found that the precision value of the ANN 1 model, for the classes 0, 1, 2, 3, and 4 were 1, 1, 1, 1, and 1. With recall values of 1, 1, 1, 1, and 1, the ANN1 model reported 1,1,1,1, and 1 f1 score for each of the 5 classes.
c) By observing the confusion matrix for the ANN1, it is reported that ANN 1 model predicted perfectly for the 5 classes (Table 10).

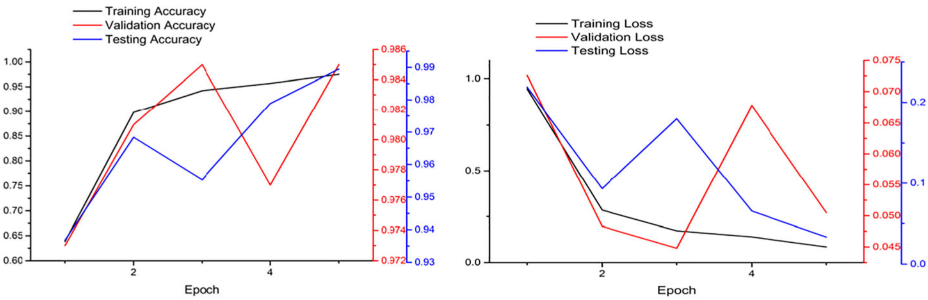



**Fig. 7** Train and validation accuracy and loss of the 2nd ANN structure on GLCM data feature

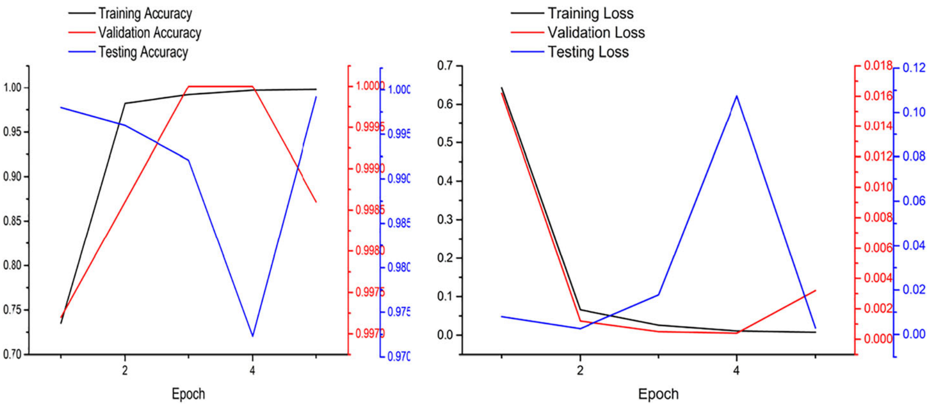



**Fig. 8** Train and validation accuracy and Loss of the best ANN (ANN1) on GLRLM Features

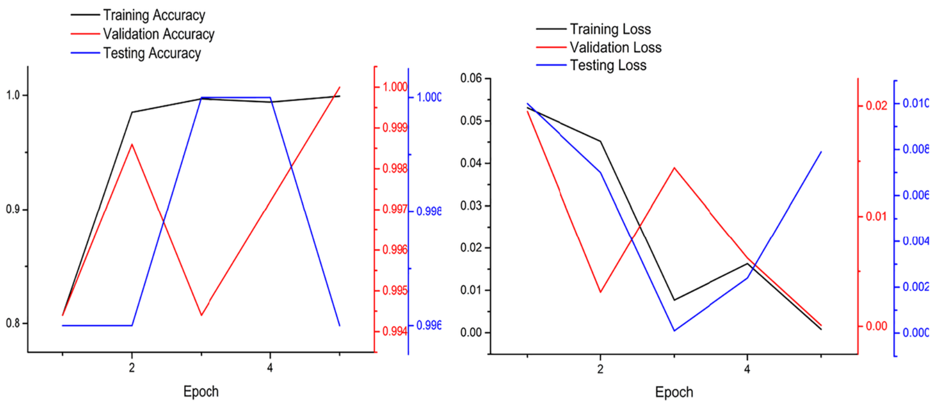



**Fig. 9** Train and validation accuracy and Loss of the best ANN (ANN1) on GLCM + GLRLM Features

The observations of the best ANN model (ANN1) on GLRLM + GLCM features are presented below:

a) Like on GLCM features, the ANN 1 model also reported positive results on GLRLM + GLCM data features with an accuracy of almost 100% for the 5 different pest classes. (Table 9).
b) The confusion matrix of the ANN 1 (Table 10) depicts the precision 1, 1, 1, 1, and 1 for the 5 different classes.With recall values of 1, 1, 1, 1, and 1, the ANN1 model reported 1,1,1,1, and 1 f1 score for the 5 different classes.
c) By observing the confusion matrix for the ANN 1 on GLRLM + GLCM data features, it is reported that ANN 1 model predicted perfectly for the 5 classes (Table 10).

### 3.4 Results of SVM, DT, and Naïve bayes

The performance of the SVM, DT, and Naïve Bayes on GLCM data features are presented in Table 11 and Table 12 whereas the performance of the SVM on GLRLM data features and GLRLM + GLCM data features are presented in Table 13 and

**Table 10** Model Performance of best ANN (ANN1) on GLRLM and GLCM +GLRLM data Features

| Data | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| GLRLM | 0 | 1 | 1 | 1 | 34 | [[34 0 0 0 0] | 100% |
| | 1 | 1 | 1 | 1 | 53 | [0 53 0 0 0] | |
| | 2 | 1 | 1 | 1 | 66 | [0 0 66 0 0] | |
| | 3 | 1 | 1 | 1 | 53 | [0 0 0 53 0] | |
| | 4 | 1 | 1 | 1 | 47 | [0 0 0 0 47]] | |
| GLRLM + GLCM | 0 | 1 | 1 | 1 | 34 | [[34 0 0 0 0] | 100% |
| | 1 | 1 | 1 | 1 | 53 | [0 53 0 0 0] | |
| | 2 | 1 | 1 | 1 | 66 | [0 0 66 0 0] | |
| | 3 | 1 | 1 | 1 | 53 | [0 0 0 53 0] | |
| | 4 | 1 | 1 | 1 | 47 | [0 0 0 0 47]] | |

**Table 11** Performance of SVM classifier with kernel Radiant Base Function (RBF) and Polynomial (Poly) on GLCM data feature

| Model | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| SVM (RBF) | 0 | 0.80 | 0.91 | 0.85 | 35 | [[32 0 0 3 0] | 90% |
| | 1 | 1 | 0.98 | 0.99 | 48 | [1 47 0 0 0] | |
| | 2 | 0.91 | 0.90 | 0.91 | 59 | [3 0 53 1 2] | |
| | 3 | 0.87 | 0.87 | 0.87 | 61 | [4 0 2 53 2] | |
| | 4 | 0.91 | 0.86 | 0.89 | 50 | [0 0 3 4 43]] | |
| SVM (POLY) | 0 | 0.75 | 0.26 | 0.38 | 35 | [[9 0 1 25 0] | 75% |
| | 1 | 1 | 0.96 | 0.98 | 48 | [2 46 0 0 0] | |
| | 2 | 0.95 | 0.59 | 0.73 | 59 | [1 0 35 21 2] | |
| | 3 | 0.52 | 0.97 | 0.67 | 61 | [0 0 1 59 1] | |
| | 4 | 0.93 | 0.82 | 0.87 | 50 | [0 0 0 9 41]] | |

Table 14, respectively. The SVM classifier with two of its kernel 'RBF' and 'Poly' performance measures are as follows:

a) The 'RBF' kernel reported precision values for the classes 0, 1, 2, 3, and 4 as 0.80, 1, 0.91, 0.87, and 0.91, respectively while the 'POLY' kernel reported as 0.75, 1, 0.95, 0.52, and 0.93, respectively. The 'RBF' kernel measured recall values for the 5 classes as 0.91, 0.98, 0.90, 0.87, and 0.86 whereas for the 'POLY' kernel it showed

**Table 12** Performance of DT and NB classifier on GLCM data feature

| Model | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| Decision Tree | 0 | 0.84 | 0.89 | 0.86 | 35 | [[31 1 0 3 0] | 90% |
| | 1 | 0.94 | 1 | 0.96 | 48 | [0 48 0 0 0] | |
| | 2 | 0.92 | 0.92 | 0.92 | 59 | [2 0 54 2 1] | |
| | 3 | 0.85 | 0.85 | 0.85 | 61 | [3 2 2 52 2] | |
| | 4 | 0.93 | 0.84 | 0.88 | 50 | [1 0 3 4 42]] | |
| Naïve Bayes | 0 | 0.41 | 0.69 | 0.52 | 35 | [[24 2 1 8 0] | 58% |
| | 1 | 0.88 | 0.90 | 0.89 | 48 | [5 43 0 0 0] | |
| | 2 | 0.69 | 0.58 | 0.63 | 59 | [10 0 34 8 7] | |
| | 3 | 0.32 | 0.21 | 0.25 | 61 | [16 4 12 13 16] | |
| | 4 | 0.59 | 0.69 | 0.62 | 50 | [3 0 2 12 33]] | |

**Table 13** Performance of SVM classifier with kernel Radiant Base Function (RBF) and Polynomial (Poly) on GLRLM Features

| Model | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| SVM (RBF) | 0 | 1 | 1 | 1 | 34 | [[34 0 0 0 0] | 99% |
| | 1 | 1 | 0.96 | 0.98 | 53 | [0 51 0 0 2] | |
| | 2 | 1 | 1 | 1 | 66 | [0 0 66 0 0] | |
| | 3 | 1 | 1 | 1 | 53 | [0 0 0 53 0] | |
| | 4 | 0.96 | 1 | 0.98 | 47 | [0 0 0 0 47]] | |
| SVM (POLY) | 0 | 1 | 1 | 1 | 34 | [[34 0 0 0 0] | 96% |
| | 1 | 1 | 0.79 | 0.88 | 53 | [0 42 0 0 11] | |
| | 2 | 1 | 1 | 1 | 66 | [0 0 66 0 0] | |
| | 3 | 1 | 1 | 1 | 53 | [0 0 0 53 0] | |
| | 4 | 0.81 | 1 | 0.90 | 47 | [0 0 0 0 47]] | |

**Table 14** Performance of SVM classifier with kernel Radiant Base Function (RBF) and Polynomial (Poly) on GLRLM + GLCM Features

| Model | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| SVM (RBF) | 0 | 1 | 0.97 | 99 | 34 | [[33 1 0 0 0] | 100% |
| | 1 | 0.98 | 1 | 0.99 | 53 | [0 53 0 0 0] | |
| | 2 | 1 | 1 | 1 | 66 | [0 0 66 0 0] | |
| | 3 | 1 | 1 | 1 | 53 | [0 0 0 53 0] | |
| | 4 | 1 | 1 | 1 | 47 | [0 0 0 0 47]] | |
| SVM (POLY) | 0 | 1 | 0.88 | 0.94 | 34 | [[30 1 0 3 0] | 95% |
| | 1 | 0.98 | 0.92 | 0.95 | 53 | [0 49 0 4 0] | |
| | 2 | 1 | 0.95 | 0.98 | 66 | [0 0 63 0 3] | |
| | 3 | 0.88 | 0.98 | 0.93 | 53 | [0 0 0 52 1] | |
| | 4 | 0.92 | 1 | 0.96 | 47 | [0 0 0 0 47]] | |

0.26, 0.96, 0.59, 0.97, and 0.82. The 'RBF' kernel yields 90% accuracy while the 'Poly' kernel shows 75% accuracy.

b) To measure the performance with the confusion matrix it was seen that the 'RBF' kernel classified 32 images correctly as damaged for class 0 while 3 images were misclassified. For class 1, out of 48 images, 47 images were correctly predicted and 1 image was misclassified. The classifier misclassified 6 images for class 2 while 53 images were predicted correctly for the respective class. For class 3, 53 images were correctly classified while 8 were misclassified, and for class 4; out of 50 images only 43 were correctly identified by the kernel while 7 images were misclassified.

c) The SVM 'POLY' kernel performed lower than the 'RBF' kernel. It classified only 25 images correctly out of 35 images for class 0 while 10 images were misclassified. Out of 48 images 46 were classified positively while 2 were misclassified for class 1 image datasets. For class 2 image datasets; out of 59 only 35 images were classified correctly while 24 images were misclassified by the classifier. A total of 2 images were misclassified, while 59 images were positively predicted for class 3 image datasets. For class 4; out of 50 images 41 were correctly classified while 9 were misclassified.

The SVM classifier with two of its kernel 'RBF' and 'Poly' performance measures on GLRLM data features are as follows:

a) The 'RBF' kernel of the SVM reported 1,1,1,1, and 0.96 precision values for the class label 0, 1, 2, 3, and 4 but the 'POLY' kernel reported 1,1,1,1, and 0.86 precision values. The 'RBF' kernel of the SVM on GLRLM data features yielded 99% accuracy while the 'Poly' kernel showed 96% accuracy (Table 13).

b) To measure the performance with the confusion matrix it was seen that the 'RBF' kernel misclassified only 2 images for class label 1 as class label 4. Other images of the different classes were perfectly classified by the 'RBF' kernel (Table 13).

c) The SVM 'POLY' kernel reported lower accuracy than the 'RBF' kernel. A total of 11 images of class label 1 were misclassified by SVM 'POLY' kernel as class 4. The other class images were perfectly classified by SVM 'POLY' kernel (Table 13).

The SVM classifier with two of its kernel 'RBF' and 'Poly' performance measures on GLRLM + GLCM data features are as follows:

a) The 'RBF' kernel of the SVM reported 1, 0.98, 1, 1, and 1 precision values for the class labels 0, 1, 2, 3, and 4 with an accuracy of 100%. The model shows better results on GLRLM + GLCM data than the GLRLM and GLCM data features. The recall value of the kernel for the 5 different class labels on the GLRLM and GLCM data features were 0.97, 1, 1, 1, and 1 (Table 14).
b) The confusion matrix reported that the 'RBF' kernel perfectly classified all the images for the 5 different class labels whereas the 'POLY' kernel misclassified 4 images as class label 4 (Table 14).
c) The SVM 'POLY' kernel reported lower accuracy than the 'RBF' kernel. The kernel depicted 1, 0.98, 1, 0.88, and 0.92 precision and 0.88, 0.92, 0.95, 0.98, and 1 recall for the 5 different classes. Three images of class 2 and class 3 were misclassified as class label 4 (Table 14).

The performance measures for DT and NB classifiers on GLCM, GLRLM, and GLCM + GLRLM are discussed as follows:

a) In DT classifier, precision values for the classes 0, 1, 2, 3 and 4 were 0.84, 0.94, 0.92,0.85 and 0.93. Respective recall values were 0.89, 1, 0.92, 0.85 and 0.84. The accuracy of the model was found to be 90% on GLCM feature data (Table 12) but the DT reported 98% (Table 15) and 99% (Table 16) in the case of GLRLM and GLCM + GLRLM data feature. The precision values reported on GLRLM data features for the five classes were 1, 0.90, 1, 1, and 1 but the respective recall values were 1,1,0.94,1, and 0.96. In case of GLCM + GLRLM data features, the respective precision values were 1, 1, 0.99, 0.96, and 1 with the recall values 1, 0.98, 0.99, 0.98, and 0.99. The confusion matrix performance (Table 12) on GLCM showed that the classifier correctly classified 31 images for class 0 while 4 images were misclassified. For class 1 images all 48 images were predicted positively. Out of 59 images, 54 were predicted correctly and 5 images were misclassed for class 2. In class 3; out of 61 images, 52 images were positively classified while predictions for 9 images were false. In class label 4; 42 images out of 50 images were classified correctly and 8 were misclassified. In the case of GLRLM (Table 15), except the class labels 2 and 4 the other class perfectly classified all the images. Two images of class

**Table 15** Performance of DT and NB classifier on GLRLM Feature

| Model | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| Decision Tree | 0 | 1 | 1 | 1 | 34 | [[34 0 0 0 0] | 98% |
| | 1 | 0.90 | 1 | 0.95 | 53 | [0 53 0 0 0] | |
| | 2 | 1 | 0.94 | 0.97 | 66 | [0 4 62 0 0] | |
| | 3 | 1 | 1 | 1 | 53 | [0 0 0 53 0] | |
| | 4 | 1 | 0.96 | 0.98 | 47 | [0 2 0 0 45]] | |
| Naïve Bayes | 0 | 1 | 0.85 | 0.92 | 34 | [[29 0 1 4 0] | 83% |
| | 1 | 0.98 | 0.91 | 0.89 | 53 | [0 43 0 0 10] | |
| | 2 | 0.86 | 1 | 0.92 | 66 | [0 0 66 0 0] | |
| | 3 | 0.75 | 0.72 | 0.73 | 53 | [0 0 7 38 8] | |
| | 4 | 0.65 | 0.72 | 0.69 | 47 | [0 1 3 9 34]] | |

**Table 16** Performance of DT and NB classifier on GLRLM + GLCM Feature

| Model | Class | Precision | Recall | F1 score | Support | Confusion Matrix | Accuracy |
|---|---|---|---|---|---|---|---|
| Decision Tree | 0 | 1 | 1 | 1 | 34 | [[34 0 0 0 0] | 99% |
| | 1 | 1 | 0.96 | 0.98 | 53 | [0 51 1 1 0] | |
| | 2 | 0.99 | 1 | 0.99 | 66 | [0 0 66 0 0] | |
| | 3 | 0.96 | 1 | 0.98 | 53 | [0 0 0 53 0] | |
| | 4 | 1 | 0.98 | 0.99 | 47 | [0 0 0 1 46]] | |
| Naïve Bayes | 0 | 0.68 | 0.82 | 0.75 | 34 | [[28 3 0 3 0] | 88% |
| | 1 | 0.95 | 1 | 0.97 | 53 | [0 53 0 0 0] | |
| | 2 | 0.96 | 1 | 0.98 | 66 | [0 0 66 0 0] | |
| | 3 | 0.84 | 0.58 | 0.69 | 53 | [13 0 3 31 6] | |
| | 4 | 0.88 | 0.94 | 0.91 | 47 | [0 0 0 3 44]] | |

label 4 were misclassified as class label 2, and 4 images of class label 2 were misclassified as class label 1. In GLCM + GLRLM-based classification using DT, 2 images of class label 1 are misclassified as class label 2 and class label 3. Another 1 image of class label 4 was misclassified as class label 3.

b) In NV classifier, precision values for the classes 0, 1, 2, 3 and 4 were 0.41, 0.88, 0.69, 0.36 and 0.59 on GLCM data features (Table 12). Respective recall values are 0.69, 0.90, 0.58, 0.21 and 0.66. The accuracy of the model was calculated as 58%. The respective precision value of the NB classifier on GLRLM data features were 1, 0.98, 0.86, 0.75, and 0.65 with an accuracy of 83%. The same algorithm reported 88% accuracy on GLCM + GLRLM data features with the precision values 0.68, 0.95, 0.96, 0.84, and 0.88. The confusion matrix performance of the NB on GLCM data showed that the classifier performed the least among the models and hence the lowest accuracy rate. Only 24 images are correctly classified by the classifier as true positive for class 0 while 11 images were misclassified. For class 1; 43 out of 48 images were correctly predicted and 5 were predicted as false. Out of 59 images only 34 images were correctly classified for class 2, the remaining 25 images were misclassified. By observing class 3, it is found that only 13 images out of 61 images were positively classified while 48 images were misclassified. For the last class label, 4; 33 images were correctly classified and 17 out of 50 images were misclassified. But the perfect classification of class label 0 of NB on GLRLM data was 29 out of 34 images wheres it was 28 in the case of GLCM + GLRLM data. For class label 1; the NB classifier reported 43 perfect classifications and 10 misclassifications on GLCM data (Table 15). The perfect and misclassification was increased in case GLCM + GLRLM data where class label 3 reported 22 misclassifications (Table 17).

## 4 Best ML model for the present study and discussion

After the evaluation of all the performance parameters of the above-mentioned ML models, it was observed that the ANN model and SVM (RBF) can be taken as the base model for pest damage identification of Coconut Plant using damage texture and color analysis (Table 17) on GLRLM and GLCM + GLRLM data features. From the experiments and observation, it was found that both the ANN models had lower rate loss in the training, testing, and validation. Hence the higher the accuracy in all three categories (Precision, recall, f1 score).

**Table 17** Accuracy Comparison of ML models on different data features

| sl no | Model | Data Features | Accuracy |
|---|---|---|---|
| 1 | ANN (1st) | GLCM | 99% |
| 2 | ANN(2nd) | GLCM | 98% |
| 3 | SVM(RBF) | GLCM | 90% |
| 4 | SVM(POLY) | GLCM | 75% |
| 5 | DT | GLCM | 90% |
| 6 | NB | GLCM | 58% |
| 7 | ANN (1st) | GLRLM | 100% |
| 8 | ANN (1st) | GLCM + GLRLM | 100% |
| 9 | SVM(RBF) | GLRLM | 99% |
| 10 | SVM(POLY) | GLRLM | 96% |
| 11 | DT | GLRLM | 98% |
| 12 | NB | GLRLM | 83% |
| 13 | SVM(RBF) | GLCM + GLRLM | 100% |
| 14 | SVM(POLY) | GLCM + GLRLM | 95% |
| 15 | DT | GLCM + GLRLM | 99% |
| 16 | NB | GLCM +GLRLM | 88% |

In this study, it is observed that the outcome and the performance of the ANN models were quite satisfactory. The ANN models executed almost 100% accuracy on GLRLM and GLCM + GLRLM data features. Apart from ANN 1, the SVM with RBF Kernel also reported almost 100% accuracy for the pest damage identification which is far better than the study reported by Singh et al., (2021) which yields an accuracy of 82.10% by using 2D CNN MobileNet model. Another study on disease detection in coconut plants using image processing was reported by Manjula, (2021) with an accuracy of 90%. In Table 17 and Table 18, the five ML models along with their performance parameters to obtain the best model for coconut plant pest damaged identification were compared. Table 17 shows that the accuracy of the DT was increasing on GLCM + GLRLM data features. The accuracy of the NB was also increased from 58% to 83% to 88% from GLCM data to GRLM data to GLCM + GLRLM data. On GLCM data, the SVM RBF reported 90% accuracy which was equivalent to the Decision tree. But the accuracy of the NB was only 58% due to the independent nature of the color and texture features of the pest infection images. In Tables 9, 10, 11, 12, 13, 14, 15, 16 and 17 the performance parameters of the five ML models are presented. By observing the parameter values, it is seen that the precision, recall, and f1 scores for the 1st ANN model were higher than the values of the 2nd ANN model for all the five classes of image datasets on the GLCM data. By observing the

**Table 18** Comparison analysis of pest damage identification

| Sl. no. | Author | Plant | Features | Algorithm | Testing Accuracy |
|---|---|---|---|---|---|
| 1 | Singh et al., (2021) | Coconut | Deep Feature | (MobileNet) | 82.10% |
| 2 | Manjula, (2021) | Coconut | RGB | Image Processing | 90% |
| 3 | Nesarajan et al., (2020) | Coconut | Deep Feature | CNN and SVM | 93.72% and 93.72% |
| 4 | Chandy, (2019) | Coconut | Deep feature | DL Algorithm | not known |
| 5 | Proposed Method | Coconut | GLRLM | ANN1 | Almost 100% |
| 6 | Proposed Method | Coconut | GLCM + GLRLM | ANN1 | Almost 100% |
| 7 | Proposed Method | Coconut | GLCM + GLRLM | SVM(RBF) | Almost 100% |
| 8 | Proposed Method | Coconut | GLRLM | SVM(RBF) | 99% |
| 9 | Proposed Method | Coconut | GLCM + GLRLM | DT | 99% |

confusion matrices for both the ANN models, it is reported that accurate classification was reported higher in the case of the 1st ANN model while the misclassification rate was higher in the case of the 2nd ANN model. Hence the 1st ANN model showed a higher rate of testing accuracy (99%) in GLCM data and later on it was increased to 100% on GLRLM data and GLCM + GLRLM data. In the same way, by comparing the SVM kernels (RBF and POLY), the 'RBF' kernel had better values for precision, recall, and f1 score than the 'POLY' kernel on all the data features. Comparing the confusion matrices for both the kernels, it is noted that the 'RBF' kernel had a better performance in the correct classification of image datasets for the five respective classes than the 'POLY' kernel. The accuracy rate for the testing image datasets was higher in the case of the 'RBF' kernel (99% on GLRLM and 100% on GLCM + GLRLM) than the 'POLY' kernel (96% to 99%). Another two classifiers DT and NB were implemented to opt for a greater comparison as well as classification. In the DT classifier, it performed far better than the NB classifier by noticing the performance parameters i.e., precision, recall, and f1 score. By comparing the confusion matrices of the two classifiers it is found that the NB classifier performed lower than the DT classifier in the case of positive classification of pest-damaged image datasets. Besides, the DT classifier had a much better testing accuracy (99% on GLCM + GLRLM) than the NB classifier (88% on GLCM + GLRLM).

It is perceived from the above comparison of the five ML models that the 1st ANN model and SVM (RBF) are the best choice for classification as well as identification of pest damage in Coconut plants using damage texture and color. After applying five different ML classifiers, it is noticed that the testing accuracy was highest in the ANN and SVM(RBF) models which are shown in Table 18. The SVM and DT classifiers made a good justification for the classification while the NB classifier performed the lowest in all parameters.

Here presents a comparative analysis of pest damage identification in coconut plants using different Machine learning and other Deep learning techniques and the performance of those techniques is reported as follows:

From Table 18, it is found that the proposed system ANN1 and SVM(RBF) reported the best results even though the model is based on the human-made color and texture feature. Singh et al., (2021) and Nesarajan et al., (2020) reported the deep feature to classify the coconut diseases and pests, but they reported only 82.10% and 93.72% for using MobileNet and CNN, respectively. The accuracy reported in this study using ANN1 and SVM(RBF) is also more than the accuracy reported by Chandy, (2019).

## 5 Conclusion

The present study provides accurate results for pest damage identification using color and texture analysis. A hand-crafted color and texture-based damage analysis is presented to classify the 5 different pest attacks on coconut. The coconut pest attack images are collected for this study by using a digital camera in a robust environment. Different classifier models, such as ANN, SVM, DT, and NB are reported in the paper with good accuracy but the satisfactory outcome is reported in ANN and SVM with an accuracy of almost 100% on GLCM + GLRLM data features. The results reported by the GLRLM and GLCM indicate that the fuse features of these methods can be considered a good texture method for pest damage identification. The developed computerized automated system will be more useful to the cultivators in improving the yield and will be a great contribution in the field of agriculture as it provides accurate results for the images collected by own-self which can be used for early pest detection in plants. The advantages of a developed computer-based disease detection system for coconut can prove of much value to agricultural organizations and cultivators by designing low-computational requirement models which can be implemented at low costs.

**Acknowledgments** This research project is supported by the CDB (Kerala), GoI under the Research Scheme, vide Ref. No.: File No. 1345/2018/12671 dated March 23, 2019.

**Data availability** The data generated and used in the current study are already presented partially in the manuscript in tabular format and graphs. The complete data will be provided upon request.

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

## References

1. Barman U, Choudhury RD (2020) Soil texture classification using multi class support vector machine. Inf Process Agric 7:318–332
2. Barman U, Choudhury RD, Sahu D, Barman GG (2020) Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease. Comput Electron Agric 177:105661
3. Barman U, Sahu D, Barman GG, Das J (2020) Comparative assessment of deep learning to detect the leaf diseases of potato based on data augmentation. In: 2020 international conference on computational performance evaluation (ComPE). IEEE, pp 682–687
4. Barman U, Sahu D, Barman GG (2021) A deep learning based android application to detect the leaf diseases of maize. In: Proceedings of the Sixth International Conference on Mathematics and Computing. Springer, pp. 275–286
5. Bharathi S, Harini P (2020) Early detection of diseases in coconut tree leaves. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). IEEE, pp 1265–1268
6. Chandy A (2019) Pest infestation identification in coconut trees using deep learning. J Artif Intell capsule Netw 01:10–18. https://doi.org/10.36548/jaicn.2019.1.002
7. Kumar N, Dahiya AK, Kumar K (2020) Image restoration using a fuzzy-based median filter and modified firefly optimization algorithm. Int J Adv Sci Technol 29:1471–1477
8. Loukas S (2021) How Scikit-Learn's StandardScaler works. In: Medium. https://towardsdatascience.com/how-and-why-to-standardize-your-data-996926c2c832. Accessed 5 Jan 2022
9. Manjula DG (2021) Coconut tree disease identification using image processing 8:5
10. Mohajon J (2021) Confusion matrix for your multi-class machine learning model. In: Medium. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.
11. Nesarajan D, Kunalan L, Logeswaran M, Kasthuriarachchi S, Lungalage D (2020) Coconut disease prediction system using image processing and deep learning techniques. In: 2020 IEEE 4th international conference on image processing, Applications and Systems (IPAS). IEEE, pp. 212–217
12. Ngugi LC, Abelwahab M, Abo-Zahhad M (2020) Tomato leaf segmentation algorithms for mobile phone applications using deep learning. Comput Electron Agric 178:105788
13. Ning X, Gong K, Li W, Zhang L, Bai X, Tian S (2020) Feature refinement and filter network for person re-identification. IEEE Trans Circuits Syst Video Technol 31:3391–3402
14. Sharma AK, Tiwari S, Aggarwal G, Goenka N, Kumar A, Chakrabarti P, Chakrabarti T, Gono R, Leonowicz Z, Jasiński M (2022) Dermatologist-level classification of skin Cancer using cascaded Ensembling of convolutional neural network and handcrafted features based deep neural network. IEEE Access 10:17920–17932
15. Shung KP (2020) Accuracy, precision, recall or F1? In: Medium. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9. Accessed 7 Jan 2022
16. Singh P, Verma A, Alex JSR (2021) Disease and pest infection detection in coconut tree through deep learning techniques. Comput Electron Agric 182:105986. https://doi.org/10.1016/j.compag.2021.105986

17. Sreenivasa S (2020) Radial basis function (RBF) kernel: the go-to kernel. In: Medium. https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a. Accessed 21 Jan 2022
18. Verma SS, Prasad A, Kumar A (2022) CovXmlc: high performance COVID-19 detection on X-ray images using multi-model classification. Biomed Signal Process Control 71:103272
19. Wang C, Wang X, Zhang J, Zhang L, Bai X, Ning X, Zhou J, Hancock E (2022) Uncertainty estimation for stereo matching based on evidential deep learning. Pattern Recogn 124:108498
20. Zhang H, Hung C-L, Min G, Guo J-P, Liu M, Hu X (2019) GPU-accelerated GLRLM algorithm for feature extraction of MRI. Sci Rep 9:1–13