Intelligent Disease Detection in Sugarcane Plants: A Comparative Analysis of Machine Learning Models for Classification and Diagnosis

[1]Santhrupth B. C., [2]Devaraj Verma C.

Abstract: Sugarcane is an important crop, but its production is hampered by problems such as water shortages and disease. This article presents a machine learning-based approach to accurately detect and classify sugarcane leaf diseases using convolutional neural networks (CNNs), random forests, and support vector machine models. This study focuses on the global importance of sugarcane, the prevalence of sugarcane in medical applications, and the main cultivation regions. The implementation includes the use of Python programming and deep learning algorithms, specifically his SVM, random forest, and CNN to classify sugarcane leaf diseases based on color, texture, and shape features. This process includes data collection, local binary patterns, texture analysis using Gabor filters and his GLCM, and disease classification.

Keywords: *Sugarcane, Leaf Diseases, Convolution Neural Network (CNN), Random Forests, Support Vector Machine, Image Processing, Machine Learning, Texture Analysis, Disease Classification, Crop Health, Agricultural Intelligence*

1. Introduction

Sugarcane is a long-duration crop. Long-duration crops various types of problems like shortage of water, weather changes, and floods are faced. These are the reasons for fungal diseases like red rot, rust, mosaic, and yellow leaves for sugarcane leaves. These types of diseases stopped the production of sugarcane. That's why detecting these diseases and stopping these diseases is important. For accurately detecting these sugarcane leaf diseases and classifying them an effective technique made by machine learning is described here. The Convolution Neural Network model, Random Forests Model, and Support Vector Machine model are built to detect these sugarcane leaf diseases.

Sugarcane world stands first among its most used products, in the overall world. Sugarcane is used by the medical department. The sugarcane world's largest use is in Brazil out of a 655million tones approximately. The largest making place in Uttar Pradesh. The sugarcane is provided by 80% Karnataka. Uttar Pradesh, Maharashtra, or Karnataka in India. The sugar can plants are 6-20ft marksman height, totaling, in the Austronesian around 1200-1000BC in prehistoric China and India. The seeds, water, soil, or fertilizers of efficiency on the resource.in always sugarcane examined with necessary to the identify and detect. Keep track of the disease. This plant growth gets infected a lot by diseases the color dost and

streaks on the plants are disease symptoms. The viruses or fungi are usually identified by eyes and hands. Sugarcane is used the detect a variety of diseases. The sugarcane grows 6 to 7 months to the disease. The foliar mannose and copper 0.3% and 0.2% over the 10 to 15 days minimize with 2 to 3 sprays in sugarcane

2. Implementation

The work is being done with Python programming in Jupiter Notebook. The coding part consists of Deep learning algorithms like *"Support Vector Machines (SVM), Random Forests"*, and *"Convolutional Neural Networks (CNN)"*. The coding is used to create accuracy for the classification of the sugarcane leaves for which the quality of leaves determines the quality of the plant. The algorithm development includes multiple steps of the process, load to fit the classifier of the data, and find the predictions of the sugarcane leaf to find whether the leaf is facing some disease or not. This algorithm checks the color of the leaves using Histogram models, the moments of colors, and color capture variations caused by these diseases. The data is collected and checked by the classifiers to determine the disease present in the Sugarcane plant and find out whether it is affected by disease or not (Usha Rani *et al.* 2022). The processing phase collects the data and converts the data into many sub-data and creates a table to analyze those data using classifiers. Then the data are used for texture analysis methods like Local Binary Patterns (LBP), Gabor Filters, and grey-level- co-occurrence matrix (GLCM). These methods are used for texture analysis and capture patterns of the leaf textures. The texture and patterns were classified, and then many classifications and algorithms were used to check the textural classes. CNN

[1]*Research Scholar, Jain University, Bangalore*
[1]*Assistant Professor, Dept. of CSE, Christ (Deemed to be) University, Bangalore*
[2]*Professor, Dept, of CSE, Jain (Deemed to be) University, Bangalore*
[1]*santhrupth23@gmail.com*
[2]*c.devaraj@jainuniversity.ac.in*

is used in images for texture analysis which includes classification and segmentation which includes generally transform-based, model-based methods.

Sugarcane is the major problem in real-time decisions on the intelligence system to be obtained. The crops or its Cole's environment of the date as result from their distribution their growth variety the natural drought such and disaster arises on their sugarcane its croup or distribution spatial from harmful as extraordinary important in the expert system. (Alencastre-Miranda *et al.* 2020). In the last place this paper automatically to sugarcane on the pager of the planting areas it's mixed and hilly of the drought of the growth or classification of extraction for the mixed and hilly region using on their high temporal on the sugar can. (Shakoor *et al.* 2017)By the EOS data and water are used in the expert system they combined their use to their advantage (Thilagavathi *et al. 2020)* The framework and method for expressing and mechanism of the real-time.

Classification and detection of disease

Classification of disease is a crucial aspect of sugarcane leaf disease allowing one to categories them through image processing techniques. This paper gives an elaborate view of the technique to detect and classify various leaf diseases. Shape feature and color features are joined together and then the features of vectors are randomly selected from each class. The use of machine learning in identification and detection helps to counter this problem. *"Convolutional Central Network (CNN)"* and *"Supporting Vector Machine (SVM)"* modes have been used to classify the disease. Some disease symptoms cannot be properly detected as they are spread all over the infected areas of the leaf (Alencastre-Miranda *et al.* 2020). Leaf disease manifests various symptoms at different stages, so all these stages have to be considered to get the result accurate. The main cause for leaf disease of sugarcane is infected by various viruses. The spindle leaves infected by the Red-rot virus display drying and later stalks become discolored and hollow. Gradually it spreads all over the leaves and becomes larger. It turns red-brown to brown on both sides of the leaf then it indicates dead leaves.

3. Result

Fig 1: Importing of Necessary Libraries

(Source: Created by learner)

According to the above image, the essential libraries are imported in this algorithm are shown.

The sugarcane that was imported has been enhanced by 25% to 40%, has been enhanced by 50%, and the

importing necessary libraries used by machine learning in sugar syrup.

Fig 2: Adding paths to the image folders

(Source: Created by learner)

Based on the above image, how the image folder is added to the algorithm directory from the C-drive of the computer is shown. Here five folders are added to the directory.

train_datagen = ImageDataGenerator

(

rescale=rescale_factor,

rotation_range=rotation_range,

width_shift_range=width_shift_range,

height_shift_range=height_shift_range,

shear_range=shear_range,

zoom_range=zoom_range,

horizontal_flip=horizontal_flip,

fill_mode=fill_mode

)

Fig 3: Data augmentation for training set

(Source: Created by learner)

The above pseudo code demonstrates that the process of data augmentation has been implemented with the selected training set. Based on the above code, augmenting if necessary the training data to image data generator with the resale on the height shift range, of the nearest full mode.


```
Data augmentation for test set (only rescaling)

]: test_datagen = ImageDataGenerator(rescale=1.0 / 255)
   # Load and preprocess data
   batch_size = 32
   image_size = (224, 224)
   num_classes = len(class_folders)

   train_generator = train_datagen.flow_from_directory(
       'C:/dataset/Sugarcane Leaf Disease Dataset/Sugarcane Leaf Disease Dataset',
       target_size=image_size,
       batch_size=batch_size,
       class_mode='categorical'
   )

   test_generator = test_datagen.flow_from_directory(
       'C:/dataset/Sugarcane Leaf Disease Dataset/Sugarcane Leaf Disease Dataset',
       target_size=image_size,
       batch_size=batch_size,
       class_mode='categorical'
   )

Found 2521 images belonging to 5 classes.
Found 2521 images belonging to 5 classes.
```

Fig 4: Data augmentation for test set

(Source: Created by learner)

The provided code is for setting up data generators using the Keras ImageDataGenerator class for training and testing a neural network on a dataset of sugarcane leaf images. This is typically used when working with image data for tasks like image classification. According to the above attached figure, the data augmentation process has been done for the test data set to evaluate the rescaling process of the training generator and also the test generator. After running this rescaling process, there are 2521 images have been founded to 5 separate classes as per the data taken for testing. The code sets up data generators that will generate batches of augmented images during the training and testing phases. These generators facilitate efficient loading and preprocessing of image data, which is crucial when working with large datasets in machine learning models, particularly deep neural networks.
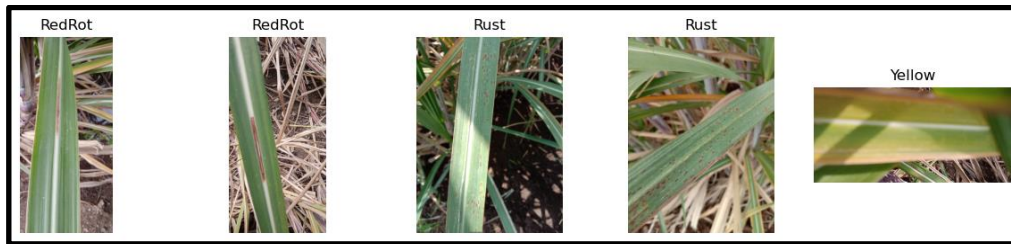
Fig 5: Symptoms of leaf disease of sugarcane

(Source: Developed by the learner)

The above leaf images of sugarcane indicate the different diseases of sugar cane leaves. The red rot is mainly spread in the rainy season and at the first stage, it is transmitted through soil and diseased sets. The other image, the rust disease symptom, is shown in the images. At the first stage rust on the leaves is small and yellowish. Initial symptoms of yellow leaf disease appear on leaves 3 to 6 counting down from the top expanding spindle leaf. The yellowing expands out from the leaf midrib into the leaf blade when a general yellowing is not observed in time. The leaves of the spindle drying in the display become hollow and discolored. After splitting open diseased small emanates or diseased stalks, they reddened with transverse spots of the witch.

Building the CNN model

```python
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(image_size[0], image_size[1], 3)),
    MaxPooling2D(2, 2),

    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),

    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),

    Flatten(),
    Dense(256, activation='relu'),
    Dropout(0.5),
    Dense(num_classes, activation='softmax')
])
```

Compiling the model

```python
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Fig 6: Building and Compiling the CNN Model

(Source: Created by learner)

The algorithm of the Convolution Neural Network model and the compiling of that model are shown in the above image,

```
27
Epoch 5/5
78/78 [==============================] - 261s 3s/step - loss: 1.0161 - accuracy: 0.5886 - va
62

: # Evaluate the model
test_loss, test_acc = model.evaluate(test_generator, steps=num_test_samples // batch_size)
print(f"Test accuracy: {test_acc}")

78/78 [==============================] - 73s 936ms/step - loss: 0.8992 - accuracy: 0.6458
Test accuracy: 0.6458333134651184
```

Fig 7: Evaluate of CNN model

(Source: Created by learner)

In the above image, the algorithm of evaluation for the Convolution Neural Network model is shown. The test accuracy of this model is around 64%.

```
# Load and preprocess data

data = []

labels = []

# Loop through classes and images

for class_name, folder_path in class_folders.items():

    for image_name in os.listdir(folder_path):

        # Read and resize the image

        image_path = os.path.join(folder_path, image_name)

        image = imread(image_path)

    # Resize images to a consistent size

        image_resized = resize(image, (224, 224, 3))

    # Convert color image to grayscale

        gray_image = rgb2gray(image_resized)

# Extract color-based features

        color_histogram = np.histogram(image_resized.ravel(), bins=256, range=[0, 256])[0]

 # Extract texture-based features

        gray_image_ubyte = img_as_ubyte(gray_image)

        lbp_image = local_binary_pattern(gray_image_ubyte, P=8, R=1, method='uniform')

        glcm = graycomatrix(gray_image_ubyte, distances=[1], angles=[0], symmetric=True, normed=True)

# Extract shape-based features

        entropy = shannon_entropy(gray_image)

        hog_features = hog(gray_image_ubyte, pixels_per_cell=(16, 16), cells_per_block=(1, 1))

 # Combine all features

        features = np.concatenate([color_histogram, lbp_image.ravel(),

                        glcm.ravel(), [entropy], hog_features])

# Append features and class label

        data.append(features)

        labels.append(class_name)
```

Fig 8: Extraction of Necessary Feature

(Source: Created by the learner)

Based on the above Pseudocode , the algorithm for the extraction of necessary features is shown. This code is responsible for loading and preprocessing image data for a machine learning model. Here three types of features are extracted. Those features are color-based, texture-based, and shape-based. The features are then concatenated into a single feature vector, and both the features and class labels are stored in lists (data and labels, respectively).

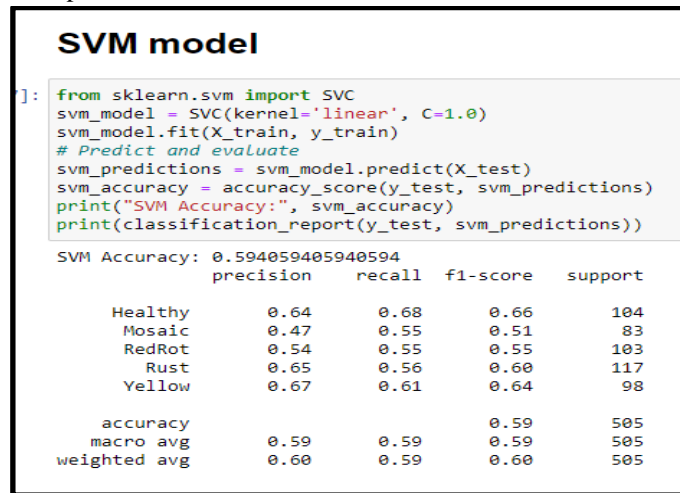## Spliting data into training and testing sets

```
]: from sklearn.model_selection import train_test_split
   X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=42)
```

Fig 9: Data Splitting

(Source: Created by learner)

According to the above image, the algorithm for the splitting of data is shown. This data splitting is done into testing sets and training sets. The dataset requirement for the successful deployment of deep architectures to the task of the problem of the specific collection of data collection and requires significant efforts in the sugarcane database creation.
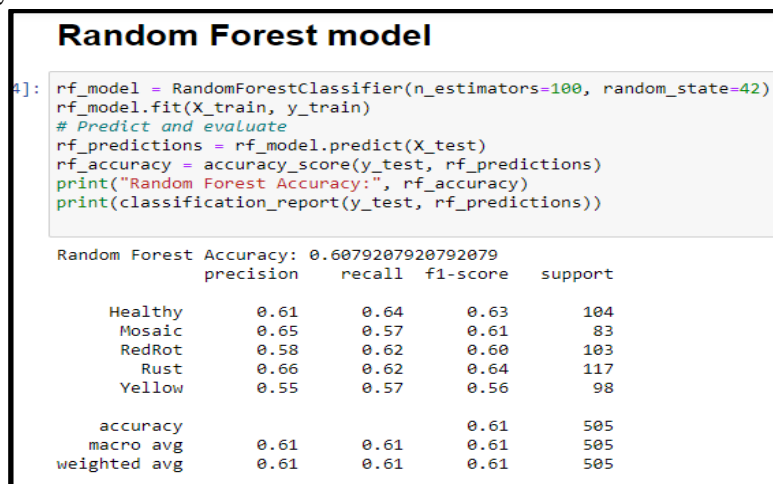
SVM model

```
]: from sklearn.svm import SVC
   svm_model = SVC(kernel='linear', C=1.0)
   svm_model.fit(X_train, y_train)
   # Predict and evaluate
   svm_predictions = svm_model.predict(X_test)
   svm_accuracy = accuracy_score(y_test, svm_predictions)
   print("SVM Accuracy:", svm_accuracy)
   print(classification_report(y_test, svm_predictions))

   SVM Accuracy: 0.594059405940594
                 precision    recall  f1-score   support

        Healthy       0.64      0.68      0.66       104
         Mosaic       0.47      0.55      0.51        83
         RedRot       0.54      0.55      0.55       103
           Rust       0.65      0.56      0.60       117
         Yellow       0.67      0.61      0.64        98

       accuracy                           0.59       505
      macro avg       0.59      0.59      0.59       505
   weighted avg       0.60      0.59      0.60       505
```

Fig 10: Support Vector Machine Model

(Source: Created by learner)

According the above image, the algorithm of the Support Vector Machine model is shown. The prediction and evaluation function for this model is shown. The accuracy of this model is around 59%.

The classification results in very important for the selected paper. The ways of method artificial immune and intelligent algorithms etc. If the range is range vin order value SVM model used by the healthy, mosaic, Redroot, Rust, Yellow, accuracy, macro age, weighted avg.

Random Forest model

```
4]: rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
    rf_model.fit(X_train, y_train)
    # Predict and evaluate
    rf_predictions = rf_model.predict(X_test)
    rf_accuracy = accuracy_score(y_test, rf_predictions)
    print("Random Forest Accuracy:", rf_accuracy)
    print(classification_report(y_test, rf_predictions))

    Random Forest Accuracy: 0.6079207920792079
                  precision    recall  f1-score   support

         Healthy       0.61      0.64      0.63       104
          Mosaic       0.65      0.57      0.61        83
          RedRot       0.58      0.62      0.60       103
            Rust       0.66      0.62      0.64       117
          Yellow       0.55      0.57      0.56        98

        accuracy                           0.61       505
       macro avg       0.61      0.61      0.61       505
    weighted avg       0.61      0.61      0.61       505
```
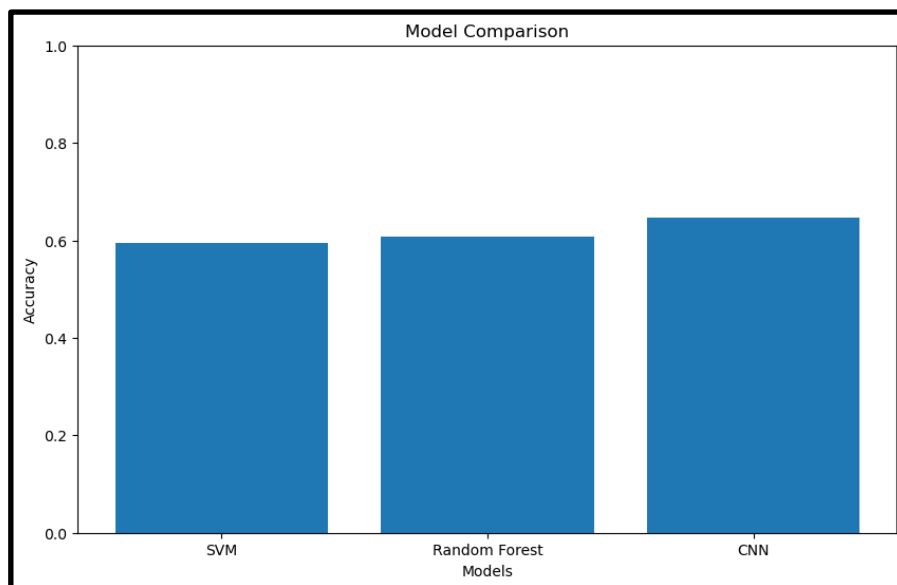
Fig 11: Random Forest Model

(Source: Created by learner)

Based on this above image, the algorithm for the Random Forest Model is shown. The prediction and evaluation function of this model is shown here. The accuracy of this model is around 61%.

Fig 12: Model Comparison

(Source: Created by learner)

The comparison of accuracy between those three models is shown in this figure. The X-axis determines the three types of models and the Y-axis determines the accuracy of those models.

Convolution Neural Network has the highest accuracy.

This model comparison by the X-axis determines accuracy types and the Y-axis SVM, Random Forest models, and CNN determines in their neural network.

4. Discussion

The long-duration crops face a lot of disease due to shortage of water, differentiation in water, floods, and nutritional lags. Fungal infections such as red rot, rust, yellow, and mosaic viruses on sugarcane leaves may impact the overgrowth and health issues of cane. The disease leaf of sugar cane is acquired through an image processing technique where the unwanted image data are suppressed. The described process is executed using CNN and Machine Learning. A detailed review of the cause and the symptoms of sugarcane disease is highlighted in this project. A detailed classification of the leaf virus is described clearly (Narmilan *et al.* 2022). There has been a significant advantage in the field of sugarcane leaf disease detection. It is very useful for agriculture and the farmer and it affects the productivity of cane.

The Sugarcane discussion leaf of acquired or through technique processing the unwanted suppressed data. (Sinha *et al.* 2020). The sugarcane is enhanced through AHE, a process done through clustering. Sugarcane is used by PCA or GLCM on the final (Elavarasan *et al.* 2018). The classification of diseases is done by SVM. Above the obtained results for six diseases as shown in the figures in sugarcane.

5. Conclusion

The research reviews different plant identification by finding their advantages and disadvantages. The total accuracy from the respective ML models was more than 60%. High-resolution images are taken to implement the Machine Learning algorithm to use pixel-by-pixel classification to find the type of leaf. The machine learning methods are being evaluated to find the accuracy of the leaf structure, color, and texture to find the accuracies based on a pixel scale on a plant scale. The research can guide the sugarcane plantation to find the disease of the plant by pinpointing the infected area of the sugarcane fields.

Sugarcane in the present situation, is difficult for the farmers' eyes or for each plant to keep growing and manifest in infection. The sugarcane functions as a watchdog the detecting the disease or determining the disease in the leaf. The sugarcane used by GLCM or PCA various feature extractions with SVM classifier has developed the combination in the various extractions on their implemented. The system value of 95% review of the causes or symptoms of all diseases are highlighted and symptoms of all the sugarcane diseases in this article. For all forms of sugarcane in the diseases. The information made the regional language.

References

[1] Alencastre-Miranda, M., Johnson, R.M. and Krebs, H.I., 2020. Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties. IEEE Transactions on Industrial Informatics, 17(2), pp.787-794.

[2] Alencastre-Miranda, M., Johnson, R.M. and Krebs, H.I., 2020. Convolutional neural networks and transfer learning for quality inspection of different sugarcane varieties. *IEEE Transactions on Industrial*

[3] Elavarasan, D., Vincent, D.R., Sharma, V., Zomaya, A.Y. and Srinivasan, K., 2018. Forecasting yield by integrating agrarian factors and machine learning models: A survey. *Computers and electronics in agriculture*, *155*, pp.257-282.

[4] Geetha, G., Samundeswari, S., Saranya, G., Meenakshi, K. and Nithya, M., 2020, December. Plant leaf disease classification and detection system using machine learning. In *Journal of Physics: Conference Series* (Vol. 1712, No. 1, p. 012012). IOP Publishing.

[5] *Informatics*, *17*(2), pp.787-794.

[6] Narmilan, A., Gonzalez, F., Salgadoe, A.S.A. and Powell, K., 2022. Detection of white leaf disease in sugarcane using machine learning techniques over UAV multispectral images. Drones, 6(9), p.230.

[7] Shakoor, N., Lee, S. and Mockler, T.C., 2017. High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. *Current opinion in plant biology*, *38*, pp.184-192.

[8] Sinha, A. and Shekhawat, R.S., 2020. Review of image processing approaches for detecting plant diseases. *IET Image Processing*, *14*(8), pp.1427-1439.

[9] Thilagavathi, K., Kavitha, K., Praba, R.D., Arina, S.V. and Sahana, R.C., 2020. Detection of diseases in sugarcane using image processing techniques. *Bioscience Biotechnology Research Communications, Special Issue*, (11), pp.109-115.

[10] Usha Rani, M., Saravana Selvam, N. and Jegatha Deborah, L., 2022. An improvement of yield production rate for crops by predicting disease rate using intelligent decision systems. Int. J. Softw. Sci. Comput. Intell.(IJSSCI), 14(1), pp.1-22.

[11] Verma, A.K., Garg, P.K. and Hari Prasad, K.S., 2017. Sugarcane crop identification from LISS IV data using ISODATA, MLC, and indices based decision tree approach. *Arabian Journal of Geosciences*, *10*, pp.1-17.