MCIP: Mining Crop Image Data on PySpark Data Frame Using Feature Selection and Cluster-Based Techniques

Yashi Chaudhary* and Heman Pathak

Gurukul Kangri (Deemed to be University), Haridwar, Uttarakhand, India

E-mail/Orcid Id:

YC, mohita.chaudhary5@gmail.com, https://orcid.org/0000-0002-3959-4008; HP, hpathak@gkv.ac.in

Abstract: In India, the yearly economic losses incurred due to crop-related issues, including pests and diseases, surpass a staggering amount of $500 billion. Leaf blight constitutes a significant determinant in the remarkable economic ramifications, primarily affecting farmers engaged in cultivating forage and grain sorghum, who bear the brunt of its consequences. Numerous crops are affected by this disease, including maize, rice, tomato, potato, millet, and onion. However, crop variety, different disease kinds, and environmental conditions make early disease identification difficult. Due to the variety of crops and diseases, existing approaches for disease classification and prediction need more broad applicability. These techniques use image preprocessing and segmentation to handle datasets with specified inputs and outputs, frequently resulting in data loss and erroneous categorization. Additionally, they ignore specialized datasets. To address these challenges, this study proposes an innovative approach leveraging the PySpark-based mining crop image data (MCIP) framework. MCIP employs Principal Component Analysis (PCA) to extract relevant features, subsequently utilized by the K-means algorithm to identify distinct subgroups. This approach, initially demonstrated on potato leaves, proves valuable for disease identification. Notably, MCIP isn't restricted to potatoes; it’s adept at detecting diseases across agricultural crops. To validate, an experiment was conducted on a rice disease dataset. Evaluation metrics, including Accuracy, Silhouette score, speed, and F1 score, affirm MCIP’s robustness. Impressively, MCIP exhibits exceptional speed and accuracy, nearly achieving 100% accuracy. This innovative model signifies a significant advancement over existing techniques, offering a promising solution to the pressing issue of crop disease management.

Keywords: Agriculture, clustering, data mining, k-means, PCA, PySpark

Introduction

The economy of a nation is predominantly dependent on its agricultural sector. The attainment of food self-sufficiency is a critical factor in ensuring the survival and resilience of a nation. Researchers and growers devote their attention to various factors to enhance productivity. However, the issue of crop loss resulting from disease stands out as one of the most significant challenges they encounter. The monitoring of crop growth and the early identification of pest infestations continue to present a significant challenge. Manually monitoring and identifying insect and pest infestations is becoming progressively more difficult due to the expansion of farming into larger fields (Velusamy et al., (2021); Dhaliwal et al., (2021). Identifying crop loss at the field parcel scale through the utilization of satellite images presents several challenges. Firstly, crop loss can be attributed to many factors that occur throughout the entire growth cycle. Secondly, dependable reference data pertaining to crop loss is limited. Lastly, the definition of crop loss can vary across different contexts (Hiremath et al., 2021).

Helminthophobia, the pathogenic fungus known as turcicum Pass, is the causative agent of leaf blight. The disease on sorghum leaves manifests as reddish-purple or brown spots that coalesce into extensive lesions,
According to Ali et al. (2017), the DeltaE approach may demonstrate efficacy in the early detection of diseases. In their study, the researchers employed the K-Nearest Neighbors (KNN) algorithm and the cubic Support Vector Machine (SVM) to classify diseases at both the image and disease levels.

Several approaches for leaf disease detection in various crops have been developed recently by various researchers (Singh et al., 2019, Zhang et al., 2019, and Lu et al., 2017). In most strategies, image processing techniques were used to take out features, which were then input into a technique based on classification. Deepa et al. (2021) suggested a method for detecting plant leaf disease. Authors use Kuan filter to remove noise before extracting colour, shape, and texture information using the Hough transformation. The plant leaf disease was classified using a reweighted linear programme boost classification. The suggested technique’s performance was assessed using the Plant Village dataset. Hamuda et al. (2018) suggested a crop identification system based on image processing that used the Kalman filtering algorithm and the Hungarian algorithm. Over a mobile-acquired picture, Picon et al. (2019) employed the ResNet-50 architecture, a deep CNN architecture. They used stochastic gradient descent optimization to train the network. Ferentinos KP (2018) suggested a deep learning-based approach for detecting plant leaf disease using multiple CNN architecture models on an open dataset with 58 discrete classifications. Huang et al. (2018) proposed utilizing the RBF (radial basis function) kernel of a support vector machine to identify sugarcane borer illness choices made in a quick manner utilizing basic processors in this technique. It also uses less memory for data storage, i.e., data collected during the training process. There are various datasets on which the classification and prediction of disease is done. Advances in artificial intelligence have aided researchers in identifying and diagnosing plant disease utilising proper image processing and machine learning methodologies. Singh et al. (2019) classified mango leaves using CNN. Singh (2019) used image segmentation and Particle Swarm Optimization (PSO). Convolutional Neural Network (CNN) and Deep CNN (DCNN) are widely used to classify and predict leaf diseases of wheat, tomato, corn, and seasonal crops (Sladojevic et al., 2016; Sharma et al., 2020); Agarwal et al., 2020; Mishra et al., 2020; Khamparia et al., 2020; Hussain et al., 2018). Researchers have also used Deep Neural Networks (DNN) to classify and detect plant leaf diseases Venkataramanan et al. (2019). Deep learning has also attracted researchers due to its high level of accuracy (Chandy, 2019; Karthik et al., 2020; Zhang et al. 2020).

particularly in regions with high humidity. Seedlings are just as susceptible to damage as fully-grown plants. The disease commonly manifests in a moderate to severe manner on forage sorghum within the Indian states of Haryana, Rajasthan, Uttar Pradesh, and Uttarakhand (Muimba-Kankolongo, 2018). The disease can reach pandemic levels due to significant damage inflicted upon the leaf’s photosynthetic machinery. This can result in adverse effects on forage productivity, quality, and grain yield. According to Das et al. (2016), it has been observed that during severe epidemics, there is a significant reduction in grain yields, potentially reaching a decline of 50% or even more.

The monitoring of plant health and timely identification of symptoms is essential for mitigating disease transmission, thereby facilitating farmers in implementing efficient management strategies and enhancing agricultural productivity. Therefore, the identification of crop diseases is of utmost importance in order to sustain agricultural productivity. Conventional methods of plant disease diagnosis predominantly depend on the subjective expertise of farmers, thereby possessing inherent limitations in terms of accuracy and precision. In a previous study conducted by Sasaki et al. (1998), by employing a spectrometer as a means to ascertain the health status of plant leaves. An alternative method involved the utilisation of the polymerase chain reaction (PCR) technique, as described by Henson et al. (1993), or the real-time polymerase chain reaction (RT-PCR) technique, as outlined by Koo et al. (2013), for the extraction of Deoxyribonucleic acid (DNA) from the leaves. In their research, Prasad et al. (2016) presented a methodology aimed at the detection and classification of plant leaf diseases through the utilisation of soft computing techniques. The authors employed a genetic algorithm for image segmentation to identify and classify plant leaf diseases. The proposed methodology was assessed using a diverse range of plant foliage and demonstrated efficacy in the early detection of diseases. In their study, Ali et al. (2017) conducted research on the detection and classification of plant diseases through the utilisation of a pattern recognition algorithm applied to images of crops. The Gabor Wavelet Transform (GWT) approach was employed in conjunction with pattern recognition techniques to identify plant diseases. The study conducted by Prasad et al. (2016) introduced a novel approach to disease detection known as automated mobile vision. In order to detect diseases in plants, the researchers employed a hybrid methodology known as GWT- (Gray-Level Co-Occurrence Matrix) GLCM. According to Ali et al. (2017), the DeltaE approach may be employed to identify diseases in citrus fruits accurately. In their study, the researchers employed the K-Nearest Neighbors (KNN) algorithm and the cubic Support Vector Machine (SVM) to classify diseases at both the image and disease levels.
## Table 1. Summary of related work

<table>
<thead>
<tr>
<th>Paper</th>
<th>Dataset</th>
<th>Technique</th>
<th>Advantage</th>
<th>Disadvantage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Khamparia et al. (2020)</td>
<td>Potato</td>
<td>Deep CNN</td>
<td>The system is capable of identifying and diagnosing various medical conditions. The achieved accuracy is 96.46%.</td>
<td>Requires large dataset and GPU. It is costly to train.</td>
</tr>
<tr>
<td>Nazki et al. (2020)</td>
<td>Tomato leaves</td>
<td>Activation Reconstruction loss Generative Adversarial Network (ARL-GAN) and CNN.</td>
<td>The provided information aptly demonstrates the process of generating synthetic images. Facilitates the process of categorization. Achieved an accuracy rate of 87.6%.</td>
<td>The procedure is complex and costly to get the desired results. There is an issue of mode collapse, too.</td>
</tr>
<tr>
<td>Ganatra et al. (2020)</td>
<td>Plant Village</td>
<td>ResNet 50 and 101</td>
<td>Achieve a significantly high accuracy in disease classification by utilising a reduced number of layers. A level of precision amounting to 99.7% has been attained.</td>
<td>The model works for certain epochs, but on increasing the epochs, it would suffer from an overfitting problem thus resulting in reduced accuracy.</td>
</tr>
<tr>
<td>Sambasivam et al. (2021)</td>
<td>Cassava</td>
<td>CNN</td>
<td>Detects multiple diseases. Achieves an accuracy of 93%</td>
<td>Suffers from overfitting problems. A large training dataset is needed. The position and orientation of an object are not encoded.</td>
</tr>
<tr>
<td>Liang et al. (2019)</td>
<td>Potato</td>
<td>Resnet 50</td>
<td>Achieves an accuracy of 98%</td>
<td>Better techniques with 99.7% accuracy are available.</td>
</tr>
<tr>
<td>Khalifa et al. (2021)</td>
<td>Potato</td>
<td>CNN</td>
<td>Detects early and late phases of blight. Achieves a 98 % overall accuracy</td>
<td>Limited to small and specific datasets.</td>
</tr>
<tr>
<td>Rozazi et al. (2020)</td>
<td>Potato</td>
<td>CNN</td>
<td>Early and late blight are identified using an accuracy of 92%.</td>
<td>Shows better results.</td>
</tr>
<tr>
<td>Author(s) (Year)</td>
<td>Model</td>
<td>Disease Detection</td>
<td>Summary</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-------</td>
<td>-------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Tarik et al. (2021)</td>
<td>Image Processing picture division with machine learning</td>
<td>Detecting potato leaf disease</td>
<td>Uses Spark framework for speed.</td>
<td></td>
</tr>
<tr>
<td>Liu et al. (2022)</td>
<td>Machine learning techniques for tea plant leaves</td>
<td>Classification</td>
<td>Requires more parameters.</td>
<td></td>
</tr>
<tr>
<td>Almoujahed et al. (2022)</td>
<td>Machine learning techniques for identifying cereal blight conditions</td>
<td>Early and late stages of disease detection</td>
<td>Limited to a single dataset.</td>
<td></td>
</tr>
<tr>
<td>de Oliveira Dias et al. (2023)</td>
<td>Random forest machine-learning technique</td>
<td>Tomato late blight prediction</td>
<td>Not suitable for large datasets.</td>
<td></td>
</tr>
<tr>
<td>Sanjeev et al. (2020)</td>
<td>Feed Forward Neural Network (FFNN)</td>
<td>Accuracy of 96.5% for early and late blight disease detection</td>
<td>Loses neighborhood information.</td>
<td></td>
</tr>
<tr>
<td>Barman et al. (2020)</td>
<td>Simplified Bayesian CNN (SBCNN)</td>
<td>Blight condition in early and late stages is detected accurately with 96.75%.</td>
<td>Requires more parameters to train.</td>
<td></td>
</tr>
<tr>
<td>Jhonson et al. (2021)</td>
<td>Mask R-CNN</td>
<td>Blight condition in early and late stages is detected accurately with 98%.</td>
<td>Limited to a single dataset.</td>
<td></td>
</tr>
<tr>
<td>Lee et al. (2020)</td>
<td>CNN</td>
<td>Blight condition in early and late stages is detected accurately with 99%.</td>
<td>Requires a lot of training data.</td>
<td></td>
</tr>
<tr>
<td>Islam et al. (2017)</td>
<td>Segmentation, and Multi SVM</td>
<td>Blight condition in early and late stages is detected accurately with 95%.</td>
<td>Not suitable for large datasets.</td>
<td></td>
</tr>
<tr>
<td>Rashid et al. (2021)</td>
<td>Yolov5 Segmentation, Deep learning using CNN</td>
<td>Blight condition in early and late stages is detected accurately with 99.75%.</td>
<td>Limited to Early blight and late blight detection of a single dataset.</td>
<td></td>
</tr>
</tbody>
</table>

Spark is one of the most popular new technology trends. It is the framework with the best chance of realising the benefits of combining Big Data and Machine Learning (Jonathan et al., 2021). It is fast (up to 100x quicker than typical Hadoop MapReduce thanks to in-memory operations), delivers robust, distributed, fault-tolerant data objects (referred to as RDDs), and combines perfectly with machine learning and graph analytics. PySpark is an application program interface (API) that connects Spark with Python.

Multivariate data tables may be found on leaves. In varying quantities, the leaves might be exceedingly healthy or disease-ridden. The severity ranges from low to severe. PCA is a statistical process that summarizes the content of large data tables by having a lower number of "summary indices" that can be displayed and analyzed more easily. It aids in the identification of trends, leaps,
clusters, and outliers. PCA is used by MCIP to extract features from a dataset. To categorize the clusters, trends, and outlier features generated by PCA, MCIP employs the K-means clustering method.

MCIP is independent of a dataset and can take any image dataset and classify it. We have calculated the Silhouette score to understand how accurately images are classified. Clustering algorithms such as K-Means use the silhouette score to look at how well samples are grouped with other similar samples. The Silhouette score is calculated for each sample of unique clusters (Dutta et al., 2021).

**Hardware setup**

Since the proposed work requires parallel processing, the system requires a Nvidia graphic card and decent memory. We tested the proposed model using a gaming laptop from HP with 8 Gb RAM, Intel core i5, and 10th generation processor.

**Dataset**

Benchmark datasets of potato leaves and rice leaf diseases Rashid et al. (2021) and Sethy et al. (2020) are taken. The potato leave dataset contains 4962 images distributed in training, testing, and validation folders. The image folders are marked as late blight, early blight and healthy. The rice leaf data set contains 5932 images marked as bacterial blight, blast, tungro and brown spot. The dataset does not have a training, testing, and validation dataset. The images are distributed in the respective disease folders.
### Table 2. Potato leaf dataset

<table>
<thead>
<tr>
<th>Class</th>
<th>Number of images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early Blight</td>
<td>1928</td>
</tr>
<tr>
<td>Late Blight</td>
<td>1714</td>
</tr>
<tr>
<td>Healthy</td>
<td>1320</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>4962</strong></td>
</tr>
</tbody>
</table>

### Table 3. Rice leaf dataset

<table>
<thead>
<tr>
<th>Class</th>
<th>Number of images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bacterial Blight</td>
<td>1584</td>
</tr>
<tr>
<td>Tungro</td>
<td>1308</td>
</tr>
<tr>
<td>Brown spot</td>
<td>1600</td>
</tr>
<tr>
<td>Blight</td>
<td>1440</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>5932</strong></td>
</tr>
</tbody>
</table>

### Objective function

The objective of this research is to recognize the leaf disease accurately. The function is mathematically represented here:

\[
D = \left\{ P \sum_{i=1}^{d} L_i \rightarrow \{ L_d \} \rightarrow \text{PCA}\{ L_d \} \rightarrow K\text{-means}\{ L_{PCA} \} \rightarrow \text{classify}\{ L_K \} \right\}
\]

Here,
- \( P \) is predictive analysis
- \( L_i \) is leaf
- \( L_d \) is leaf data
- \( L_{PCA} \) is PCA features
- \( L_K \) is K-means
- \( D \) is the predicted disease

The objective function is summarized as:
1. Read each folder name and data.
2. Normalize each image data by dividing them by 255.
3. Store the normalized data with labels in a list and convert it into a .csv file.
4. Read the .csv file and drop all the Nan values.
5. Apply PCA on the data and get PCA features.
6. Create a K-means model.
7. Standardize PCA features using K-means model.
8. Compile (fit) PCA features and standardized features with K-means model.
9. Transform output from 8 for predictive analysis
10. Take a new leaf
11. Extract features using PCA
12. Using transformed output, features of the new leaf are passed to get the disease.

#### Mathematical formulation for computing the results

MCIP uses clustering techniques for classification and prediction. To check the goodness, we calculated the Silhouette score. The value of Silhouette ranges between -1 and 1.

\[
\text{Silhouette} = \frac{A_{ic} - A_{inc}}{\max(A_{inc}, A_{ic})} \quad \text{eq}(2)
\]

Here,
- \( A_{ic} \) is Average inter – cluster
- \( A_{inc} \) is Average intra – cluster

For checking the goodness of the classification of MCIP, we calculated Precision, Recall, Accuracy, and harmonic mean or F1 score. The calculations are based on a percentage of correctly identified positives or True Positive (TP), correctly identified negatives or True Negative (TN), identified as positive but not positive or False Positive (FP), and identified as negative but not negative or False Negative (FN).

#### Precision

\[
\text{Precision} = \frac{\# TP}{(\# TP + \# FP)} \quad \text{eq}(3)
\]

#### Recall

\[
\text{Recall} = \frac{\# TP}{(\# TP + \# FN)} \quad \text{eq}(4)
\]

#### Accuracy

\[
\text{Accuracy} = \frac{(\# TP + \# TN)}{(\# TP + \# TN + \# FP + \# FN)} \quad \text{eq}(5)
\]

#### F1 Score

\[
F1 = 2 \times \frac{(\text{Precision} \times \text{Recall})}{(\text{Precision} + \text{Recall})} \quad \text{eq}(6)
\]

### Research Methodology

The model is graphically represented to give an overview of the work. Detailed implementation is explained through an algorithm.

#### A. Model

The model is graphically represented in Figure 3(a) and 3(b).
The proposed model is represented using Figure 3 (a) and Figure 3 (b). MCIP reads the data from each folder and then filters it to scale the image evenly. The images are then labelled with their sub-folders names. The digitized images are then stored as a .cave file. To extract the features using PCA, we set up a PySpark environment. The data is read in PySpark framework. Once the data is read, PCA is applied with labels to extract the PCA features. Finally, K-means is applied to PCA features to classify and predict the data.

Implementation

Before applying Principal Component Analysis (PCA) and K-means clustering for disease classification and detection, we determined the optimal value of k by utilising the elbow technique. The elbow technique clusters the dataset using the k-means algorithm across k values (e.g., 1-32). Subsequently, at each value of k, the average score is calculated for all clusters. The distortion score is counted as the aggregate of the squared distances for each data point and its corresponding centroid. The MCIP algorithm transforms the input images into images with dimensions of 32x32 pixels. To determine the value of k, a dataset consisting of 1024*n data points was provided as input to the elbow visualizer. Where n represents the quantity of images and 1024 denotes the total number of pixels contained within each individual image. The potato dataset consists of three distinct classes, labelled 0, 1, and 2.

In Figure 4, the x-axis has different values of k and y-axis has a distortion score of s\textsuperscript{112}. Here, s is the score. The score is of all the pixels of the image dataset. Elbow method The Elbow technique identifies k clusters that best reflect the data points in their clusters. As a result, the data points and their respective cluster centroids' distance would be the assessment measure. In the plot, the bending starts at 3. This is where the reduction in Root Mean Square Error (RMSE) is no longer significant enough to justify raising the k value.

The equation can be represented mathematically:

\[ k = \min \left( \text{RMSE} \left( \sum_{i=1}^{n-1} \sum_{j=1}^{n} \|d_i - d_j\|^2 / 2 \right) \right) \]  

(7)

Here,

- n is the number of clusters
- \(d_i\) and \(d_j\) are the distances between two data points.

The value of k we get here is 3, which is also the number of classes in the potato leaf dataset. This is used in PCA for a number of components and k-means clustering.

Algorithm: Classification

Setup:
- Read dataset
- Install required modules
- Create spark session
- Create context

//SparkContext is the entry gate of Apache Spark functionality.
Start:

Step 1. Prepare the dataset
validation_datagen \leftarrow Using Image Data Generator
rescale validation image
test_datagen ← Using Image Data Generator rescale testing image
// Batches of tensor image data with real-time data are generated through Image data generator

Generate training set, testing set, and validation set
Resize image to 32X32 for faster processing
Step 2. Read the dataset
dataset ← read resized image dataset
label ← and label the data as: 0, 1, 2
Here,
0 is for Healthy
1 is for Late_blight and
2 is for Early_blight
ldata ← [dataset, label] #combines the data and list to have a labelled data.

Step 4. Convert data to an array
Step 5. Reshape the data to make it even.
Step 6. Divide by 255 to ensure that pixels are of same data type.
Step 7. Get the number of clusters
import k-means modules and other required modules.
Create a k-means object and fit the re-shaped data to predict.
Read re-shaped data and append labels to it.
Step 8. Clean data
Drop any Nan value from the dataset
Save the clean file as potato.csv
Step 9. Create a spark session.
Step 10. Read the CSV using spark
Step 11. Print the schema to understand the structure of the dataset
Step 12. Drop any Nan value if present
Step 13. Apply PCA
df ← Vectorize the labels
df ← Extract features
    df ← Using vector assembler to generate features
    assembled_data ← transform(df)
scale ← using Standard Scaler to generate standardized data
data_scale ← fit assembled_data
data_scale_output ← transform assembled_data
PCA ← apply PCA on data_scale_output
Here,
K ← 3
Input ← features
Output ← PCA features
Step 14. Apply K-means on results obtained after PCA

scale ← generate standardized output using StandardScaler
Here,
input ← PCA features
output ← standardized
data_scale ← fit scale to PCA
data_scale_output ← transform PCA using data_scale object

Step 15. Create an object of evaluator
Step 16. Create a kmeans object
fcol ← standardized
ncluster ← 10
KMeans_Obj ← KMeans(featuresCol ← fcol, number of clusters ← ncluster)
KMeans_fit ← fit KMeans_Obj on data_scale_output
Step 17. Evaluate the silhouette score
silhouette coefficient, alternatively referred to as the silhouette score. It is a metric used to calculate the goodness of a clustering algorithm. Its value lies between -1 to 1.
1: Indicates separated and distinct customers from one another.
0: There is no substantial distance between clusters.
-1: incorrectly assigned clusters

The spark environment is set using Python's PySpark module. Reading image data in a PySpark environment is different. The output is a PySpark data frame, not the usual data frame, as in the case of Pandas. This implies that only commands from the PySpark module can be used on the output data frame. The challenge with image classification is the speed. Reading all the images and classifying them is time-consuming. We merged training and testing images in one folder. Parallel processing or PySpark helps us read the images into a data frame amazingly fast. To address the problem of slow classification, we first resized the images to 32X32. This helped in bringing uniformity to the data as well.

Further, we divided each pixel by 255 to get the pixel values between 0 and 1. Lower pixel values increase the speed of classification. The next task is to predict the disease.

Algorithm: Prediction
Step 1. Leaf ← upload a new leaf
Step 2. PCANew ← extract features leaf using PCA
Step 3. POutput ← KMeans_fit(PCANew) // Kmeans_fit is taken from algorithm 3.2.1
Step 4. If POutput==0:
Step 5. Disease $\leftarrow$ Healthy
Step 6. else if POutput == 1:
Step 7. Disease $\leftarrow$ Late_blight
Step 8. else:
Step 9. Disease $\leftarrow$ Early_blight

For predicting the disease, the object of the compiled model is taken. The extracted features of the new image are passed through the classification output to get the predicted label.

III. Results

MCIP was evaluated on two datasets to ascertain the claim that the proposed work is independent of any dataset. The obtained results are compared with the results by existing techniques. We have designed three models to predict plant diseases.

PCA Features

Figure 5. PCA feature extraction

Figure 5 is the output of the PCA features extracted from the images. The first 20 results are reproduced here. The data is of the first 20 leaves in the image dataset.

Table 4. Silhouette score using only K-means (potato)

<table>
<thead>
<tr>
<th>K values</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7966620104390114</td>
</tr>
<tr>
<td>2</td>
<td>0.7410152961759711</td>
</tr>
<tr>
<td>3</td>
<td>0.7764226529818571</td>
</tr>
<tr>
<td>4</td>
<td>0.7556850195635196</td>
</tr>
<tr>
<td>5</td>
<td>0.74193107935930953</td>
</tr>
<tr>
<td>6</td>
<td>0.7720033864837685</td>
</tr>
<tr>
<td>7</td>
<td>0.7795863539164464</td>
</tr>
<tr>
<td>8</td>
<td>0.7409168118624311</td>
</tr>
<tr>
<td>9</td>
<td>0.6966620104390114</td>
</tr>
</tbody>
</table>

Table 4 is the output received for different values of Silhouette for different values of K. The optimum value is K=7. The accuracy achieved is 98%.

Table 5. Silhouette score MCIP (potato)

<table>
<thead>
<tr>
<th>K values</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.8966620104390114</td>
</tr>
<tr>
<td>2</td>
<td>0.8410152961759711</td>
</tr>
<tr>
<td>3</td>
<td>0.9886226529818571</td>
</tr>
<tr>
<td>4</td>
<td>0.8556850195635196</td>
</tr>
<tr>
<td>5</td>
<td>0.84193107935930953</td>
</tr>
<tr>
<td>6</td>
<td>0.8720033864837685</td>
</tr>
<tr>
<td>7</td>
<td>0.8795863539164464</td>
</tr>
<tr>
<td>8</td>
<td>0.8409168118624311</td>
</tr>
<tr>
<td>9</td>
<td>0.7966620104390114</td>
</tr>
</tbody>
</table>

Table 5 is the score received when we used PCA and K-means. MCIP performs best at K=3. The accuracy achieved is close to 100%.

Table 6. Outcome of classifying potato dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>TP</th>
<th>FP</th>
<th>TN</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rashid et al. (2021)</td>
<td>4875</td>
<td>10</td>
<td>40</td>
<td>35</td>
</tr>
<tr>
<td>Deep Learning</td>
<td>4886</td>
<td>9</td>
<td>36</td>
<td>29</td>
</tr>
<tr>
<td>K-means</td>
<td>4852</td>
<td>15</td>
<td>44</td>
<td>39</td>
</tr>
<tr>
<td>MCIP</td>
<td>4961</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6 is the score received when used PCA and K-means. MCIP performs best at K=3. The accuracy achieved is close to 100%.

Figure 6. Standardized features

After PCA feature extraction, the data is standardized. The standardized features are shown in Figure 6. PCA and standardized features are used with k-means to classify the dataset. The outputs are of the potato dataset only.

True positive is kept at 100, which is the number of images. All the input images are known to be correct; there are no images which are detected but are not correct; thus, False Positive is 0. The images which are Falsely detected negative are very low in numbering.

For comparison, we designed three models: Deep Learning, K-means, and K-means + PCA (MCIP). The other comparison is made with Rashid et al. (2021) for the potato dataset and Sethy et al. (2020) for the rice dataset.

The deep learning model has 6 Convolution 2D layers, 6 Maxpooling layers, 2 dropout layers, and two dense layers. Relu is being implemented as an activation function.
function. The activation function used on the output dense layer is softmax. All the three models run on PySpark. K-means (without PCA) re-scales the images to 32X32 as the feature input. The value of k is again 3. We wanted to see if PCA would make any difference in classification and prediction. The results show that although k-means can give good results, MCIP gives much better results.

Table 7. Accuracy using Table 6

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rashid et al. (2021)</td>
<td>0.9928</td>
<td>0.8</td>
<td>0.9909274</td>
<td>0.886061</td>
</tr>
<tr>
<td>Deep Learning</td>
<td>0.9940</td>
<td>0.8</td>
<td>0.9923387</td>
<td>0.886550</td>
</tr>
<tr>
<td>K-means</td>
<td>0.9920</td>
<td>0.74576</td>
<td>0.9890909</td>
<td>0.851445</td>
</tr>
<tr>
<td>MCIP</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

The accuracy of MCIP is close to 100%. The accuracy would range between 99.5 and 100%. The algorithm was fixed on a random state to reproduce the same results every time. The results may vary if the state changes. The fall in the accuracy of Rashid et al. (2021) is due to the higher number of images in MCIP.

Table 8. Outcome of classifying rice dataset

<table>
<thead>
<tr>
<th>Model</th>
<th>TP</th>
<th>FP</th>
<th>TN</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sethy et al.</td>
<td>5726</td>
<td>43</td>
<td>92</td>
<td>71</td>
</tr>
<tr>
<td>Deep Learning</td>
<td>5854</td>
<td>11</td>
<td>34</td>
<td>33</td>
</tr>
<tr>
<td>K-means</td>
<td>5793</td>
<td>31</td>
<td>43</td>
<td>65</td>
</tr>
<tr>
<td>MCIP</td>
<td>5929</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 9. Accuracy using Table 8

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep learning</td>
<td>0.99439</td>
<td>0.75555</td>
<td>0.99258</td>
<td>0.85867</td>
</tr>
<tr>
<td>Sethy et al.</td>
<td>0.98775</td>
<td>0.68148</td>
<td>0.98078</td>
<td>0.80651</td>
</tr>
<tr>
<td>K-means</td>
<td>0.98890</td>
<td>0.58108</td>
<td>0.98381</td>
<td>0.73202</td>
</tr>
<tr>
<td>MCIP</td>
<td>0.99983</td>
<td>0.99983</td>
<td>0.99983</td>
<td>0.99991</td>
</tr>
</tbody>
</table>

The deep learning and MCIP can predict with 99.983% accuracy, again close to 100%. Without any changes to the algorithm with the rice dataset, we also get similar results. The model can take any image, classify it and predict the disease.

Figure 7. Disease classification accuracy potato dataset

Figure 8. Disease classification accuracy rice dataset

Figure 7 and Figure 8 are plotted to show the accuracy of the predicted diseases. The accuracy varies for different diseases. The average classification accuracy is mentioned in Tables 7 and 9.

Figure 9. Time taken by potato and rice dataset

Time plays a vital role in the classification and prediction of an image dataset. We could reduce the time by 80% for potato dataset and 76% for rice leaf dataset.
Paper Rashid et al. (2021) is on potato leaf; we implemented the model on rice leaf to arrive at the results.

**Conclusion**

This study presents an analysis of the PySpark-based mining crop image data (MCIP) framework and its efficacy in tackling the difficulties associated with crop disease diagnosis and classification. The focus is specifically on the significant economic losses caused by crop-related problems in India. Through the utilisation of deep learning methodologies and novel strategies such as Principal Component Analysis (PCA) and the K-means algorithm, MCIP demonstrates its proficiency in effectively and precisely identifying illnesses in diverse agricultural crops. The framework demonstrates a noteworthy capability of reducing processing time by 76-80% while achieving almost perfect accuracy, thereby distinguishing itself from conventional picture preprocessing and segmentation methods. Although there were early difficulties encountered in interpreting pictures inside the PySpark framework, this study effectively addresses and resolves these concerns. As a result, it paves the way for future investigations that explore the combination of Principal Component Analysis (PCA) with Long Short-Term Memory (LSTM) to improve plants’ disease prediction capabilities. In general, the MCIP framework signifies a significant progression in the realm of crop disease management, demonstrating its capacity to transform the approach and mitigation of crop diseases within agricultural communities, leading to enhanced agricultural production and economic viability.

**Conflict of interest**

There is no known conflict of interest in this article.

**References**


https://doi.org/10.1016/j.compag.2020.105527


DOI: https://doi.org/10.52756/j.ijerr.2023.v34sp1.011

How to cite this Article:

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.