INTERNATIONAL JOURNAL FOR INNOVATIVE RESEARCH IN MULTIDISCIPLINARY FIELD
ISSN(O): 2455-0620 [Impact Factor: 9.47]
Monthly, Peer-Reviewed, Refereed, Indexed Journal with IC Value : 86.87
Volume - 12, Issue - 6, June - 2026

DOIs:10.2015/IJIRMF/202606001 --:-- Research Paper / Article / Review

Enhancing Cotton Crop Management: A Novel Method for Detecting Nitrogen Deficiency Using DenseNet121

[1] Swapnil S. Ayane, [2] Mukesh Tiwari
[1]Ph.D Scholar, Department of Electronics & Communication, Shree Satya Sai University of Technology & Medical Sciences, Sehore (M.P).
[2]Professor, Department of Electronics & Communication, Shree Satya Sai University of Technology & Medical Sciences, Sehore (M.P).
Email - [1]swapnil.ayane@pccoepune.org, [2]mukeshtiwari_79@yahoo.co.in

Abstract: *In India, agriculture is an important industry and source of revenue. A variety of crops are cultivated for both financial gain and food. Among the most important crop grown in many parts of India is cotton. The state of Maharashtra is the top producer of cotton. However, many diseases usually affect cotton crops, especially during growth. As a result, there are considerable yield losses. Every living organism in this world requires a sufficient amount of a variety of proteins and nutrients for its average growth. Deficiency of any of the nutrients in the crop impacts average growth and, in turn, results in poor quality of yield and financial loss. Nitrogen is a crucial nutrient for the average growth of plants, and its deficiency can lead to various symptoms in cotton leaves, such as yellowishness on the leaf and stunted growth. Early detection of nitrogen deficiency is must in guaranteeing the best yield output and prompt management. In this research, the combination of image processing and DenseNet 121 is used to identify and diagnose the deficiency. This combination gives percentage estimates of nitrogen deficiency along with the classification of unhealthy leaves. The model delivers an F_1 score of 98% and validation accuracy of 98.26%.*

Key Words: *Cotton, DenseNet-121, Deep learning, leaf disease, Nutrient deficiency.*

1. INTRODUCTION

The primary origin for the nutrient deficiency analysis is the leaf of crop. Approximately 80 % to 90 % of the disease of the cotton crop is on its leaves. The most prevalent diseases that affect cotton crops are like foliar leaf spots of cotton and fungus. All these diseases reflect the symptoms on the leaf and stems of crop. Hence, focusing on the pattern changes in cotton leaves is better. These diseases mainly occur due to the deficiency of nutrients. The cultivator determines pesticide or fertilizer to be use based on the pattern they saw on the plant leaf, but they were unsure of quantity. Hence, sometimes, a waste of corpus and an overdose of fertilizer may affect the crop life cycle and the surroundings [1]. In the early days, several laboratory methods were developed to analyze plant leaves and detect plant nutrient deficiencies using chemical processes. However, chemical processing methods are time-consuming and costly. Later on, with technological development, methods were developed to classify and detect diseases on cotton plants without laboratory procedures. Nitrogen(N) is an essential nutrient in cotton, and the requirement ranges between 2.20% – 3.50% [2]. This research paper proposes DenseNet121 to determine nitrogen deficiency in cotton plants. DenseNet-121 is a good choice under situations with limited resources and smaller datasets, feature reuse, and efficient parameters; this architecture is perfect for high accuracy and real-time analytic applications because it excels in multi-scale feature extraction and computing efficiency.

2. LITERATURE SURVEY

Numerous Solutions have been developed to identify diseases of plant leaves using segmentation, feature fusion, and image classification which makes these techniques appropriate for disease detection of cotton leaves, flowers, and fruits. In recent years, multiple approaches have been put forth for automatically recognizing diseases affecting cotton leaves

using machine learning or deep learning algorithms. A few research articles that uses these techniques for crop disease detection are explored in this section.

This paper uses Blending of segmented and RGB images processed through a multi-headed DenseNet-based architecture. The performance is evaluated through 54,183 images containing 38 classes. The model achieved metric score of over 98%, showcasing the robustness and effectiveness of the model in classifying various plant diseases [3]. During the experimentation, various learning rates and optimizers are used. The model achieved its highest accuracy with the Adam optimizer at a learning rate of 0.01, reaching a training accuracy of 96.53% and a testing accuracy of 97.43%. This shows the strong performance of DenseNet in identifying maize leaf diseases [4]. The paper describes the methods to estimate the nitrogen status in strawberries. Attention is given to non-destructive techniques. The SPAD (Soil Plant Analysis Development) meter estimates chlorophyll content in leaves. There is a positive correlation between chlorophyll concentration and leaf nitrogen content. The article used multivariate calibration analyses like principal component regression and least squares support vector machine(LSSVM) built from leaf spectral data to better predict nitrogen status [5]. Different cotton diseases were identified and classified through experiments using VGG16, ResNet50, and Mobile Net. An exceptional accuracy rate of 98% for diagnosing cotton leaf diseases was achieved. About 2,385 images of nutritious and diseased cotton leaves made up comprehensive dataset through data augmentation methods [6]. The author proposes a few-shot learning approach along CNN with improved leaf disease classification accuracy, particularly with limited training data. The method classifies disease spots using threshold segmentation and a Support Vector Machine. The DenseNet model with spatial structure optimization (SSO) notably obtained the highest accuracy, with a 7.7% improvement over the unoptimized version [7]. An accuracy of 94.9% is obtained through CNN model for classifying cucumber leaf disease. CNN has limitations, including significant parameter requirements and overfitting issues like other methods. To prevent over-fitting, the dropout concept in fully linked layers minimizes the number of parameters [8]. Using the CNN model for 25 plants over 58 different plant disease groups, the author attained a classification accuracy of 99.53% [9]. Optical annotated leaf images are employed to enhance the performance of DL architectures compared to ML models for leaf disease identification [10]. In order to get meaningful classification results for identifying leaf disease, several CNN models are employed. Another CNN variant that offers notable outcomes is DCNN, which has additional layers including VGG 16, Inception V4, ResNet, and DenseNet [11]. Leaf disease is identified by fine-tuning DenseNet with 50, 101 layers [12]. To achieve better efficiency on these datasets, DenseNet 121 is used to detect nitrogen deficiency because it overcomes the problem of vanishing gradient, enhances feature reuse, and reduces parameter usage. Furthermore, it has proven effective in leaf disease diagnosis [13].

3. METHODOLOGY

The beneath sections describe the primary steps taken to extract the nutrient deficiency from the cotton plant.

3.1. Image Acquisition and Preprocessing

The images of cotton leaves are captured using a high-resolution camera to ensure leaf's features are clearly visible for subsequent analysis. Once the images are captured, preprocessing is performed to isolate the leaf portion from the background. This is achieved using background subtraction techniques, which remove non-leaf areas, leaving only the leaf for analysis. Along with this steps such as cropping and maintaining a uniform aspect ratio of images ensure that they meet the input requirements of deep learning models and help standardize the dataset. The aim is to ensure that the DenseNet-121 focuses solely on the leaf's visual features. The images are classified in two classes considering features reflected on leaves. Consider the sample images of cotton shown in Fig.1(a). Healthy Cotton leaf and Fig.1(b). N-deficient cotton leaf. The Nitrogen deficiency results in pale yellowish color on leaves [14]. The detection and analysis of pale yellowish color is done using the OpenCV tool. The main goal is to identify pixel areas that match pale yellowish color ranges and calculate their percentage against the area of the leaf. To achieve better result, various techniques such as masking, thresholding are used. The percentage of pixels for a yellow color is calculated as the ratio of pixels that are not-zero in the mask to the total count of pixels in the grayscale version of the image, which represents the leaf area. Mathematically, it is expressed by Eq. (1).

$$\%\text{YellowColor} = \frac{\text{Count of pixels in color mask}}{\text{Total count of pixels in the leaf area}} * 100 \quad (1)$$

(a) Healthy Cotton leaf (b) N-deficient cotton leaf.

Fig.1. Sample images of cotton leaves

3.2. Data Augmentation

The Data set is artificially expanded using augmenting data by operation such as flip, rotate and contrast adjustments [15]. For effective supervised learning, accurate annotation and labeling of images are important, as they provide the ground truth that the model learns to predict. In addition, to permit efficient model training and evaluation, the dataset is separated into subsets for testing, validation, and training. Considering these factors can improve the robustness and accuracy of deep learning model. A random angle is created for rotation within a predetermined range to flip the images in either way. The rotation of the image can be either clockwise or counterclockwise depending on the angle, which can be either positive or negative. Assume that the initial image coordinates are (x, y). Then, rotating the image by an angle θ around a point (x0, y0) results in (x_rotation, y_rotation). The logical statement for rotating an image around a point (x0, y0) is

$$x\_rotation = cos(\theta) \times (x - x0) - sin(\theta) \times (y - y0) + x0. \quad (2)$$

$$y\_rotation = sin(\theta) \times (x - x0) + cos(\theta) \times (y - y0) + y0. \quad (3)$$

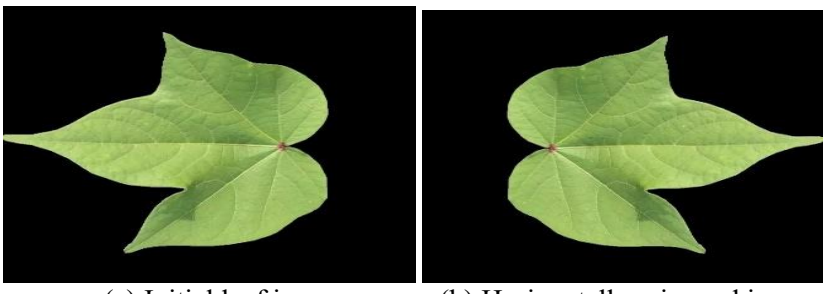
Similarly, the image is rotated in a vertical direction in horizontal flip augmentation. Mathematically this operation is expressed as

$$I\_flip\ (x, y) = I\ (width - 1 - x, y) \quad (4)$$

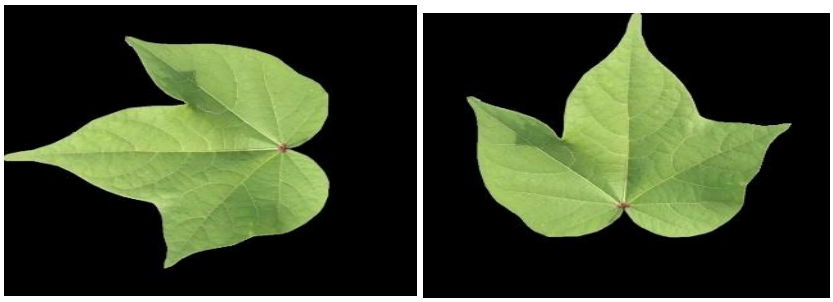
In vertical flip augmentation, the image is rotated in a horizontal direction. Mathematically this operation is expressed as

$$I\_\ flip\ (x, y) = I\ (x, height - 1 - y) \quad (5)$$

The initial leaf image Fig.2(a), the horizontally mirrored image Fig.2(b), and the vertically mirrored image Fig.2(c) are shown in Fig.2. Expansion or compression of an image's pixel value range results in contrast enhancement. It increases the visibility of specific image details. Fig.2(d) is an image after rotation.



(a) Initial leaf image (b) Horizontally mirrored image



(c) Vertically mirrored image (d) Image after rotation

Fig.2. Augmented images of cotton leaves

After data Augmentation, the total samples number is 400 images of the healthy cotton leaf, 400 images of nitrogen deficient cotton leaf. In the context of detecting nutrient deficiencies in cotton leaves, the dataset is critically important in training and the model performance evaluation. The augmented data is grouped into two classes as healthy leaf and N deficient leaf.

4. MODEL ARCHITECTURE

The proposed method uses DenseNet-121 model to determine the nitrogen deficiency in cotton plant. The DenseNet-121 architecture is a specific variant of the Densely Connected Convolutional Networks family. In proposed model, each layer gets input from all early layers and transmits its output to all following layers within the same dense block Better performance and more effective training are the results of this architecture, because it encourages feature reuse and enhances gradient flow throughout the network. The DenseNet-121 structural summary is presented in Table 1. The propose methodology is implemented in the same flow with slight variations in the parameters. In the model training phase, images from the dataset are processed using DenseNet121 for binary classification, focusing on distinguishing between healthy and Nitrogen-deficient cotton leaves.

Table 1. Summary of the structure of DenseNet-121

Layer Type	Output Size	Layers
Convolution	2x112	7x7 conv, 64 filters, stride 2
Max Pooling	56x56	3x3 max pool, stride 2
Dense Block-1	56x56	[1x1 conv, 128] + [3x3 conv, 32] * 6 (growth rate = 32)
Transition Layer-1	28x28	1x1 conv, 128 filters + 2x2 average pooling, stride 2
Dense Block-2	28x28	[1x1 conv, 128] + [3x3 conv, 32] * 12
Transition Layer-2	14x14	1x1 conv, 256 filters + 2x2 average pool, stride 2
Dense Block-3	14x14	[1x1 conv, 128] + [3x3 conv, 32] * 24
Transition Layer-3	7x7	1x1 conv, 512 filters + 2x2 average pool, stride 2
Dense Block-4	7x7	[1x1 conv, 128] + [3x3 conv, 32] * 16
Binary Layer	1x1	Global average pool, 1000-d fully connected

The training involves setting several key parameters as:

4.1. Batch Size:

Set at 32, The quantity of images processed in a single training cycle depends on the batch size. A batch size of 32 balances computational efficiency and system performance, ensuring that the model trains effectively without excessive memory consumption.

4.2. Number of Epochs:

Configured to 10, the count of epochs defines the number of times the model is trained using the complete dataset. Training for 10 epochs ensures that the model has ample exposure to the data, allowing it to learn and refine its feature representations effectively. This iterative process helps the model to converge and improve its classification accuracy over time. During training, the hyper parameters are fine-tuned to optimize performance. Validation metrics, including validation loss, accuracy, precision, recall, and F1 score, are monitored to estimate the performance of the model and prevent overfitting. The use of DenseNet-121 in training is for image classification in binary form, exemplifying a robust approach to use cutting-edge CNN architectures for accurate nutrient deficiency detection. In the proposed DenseNet-121 model shown in Fig.3. the images are processed in different phases as mentioned bellow.

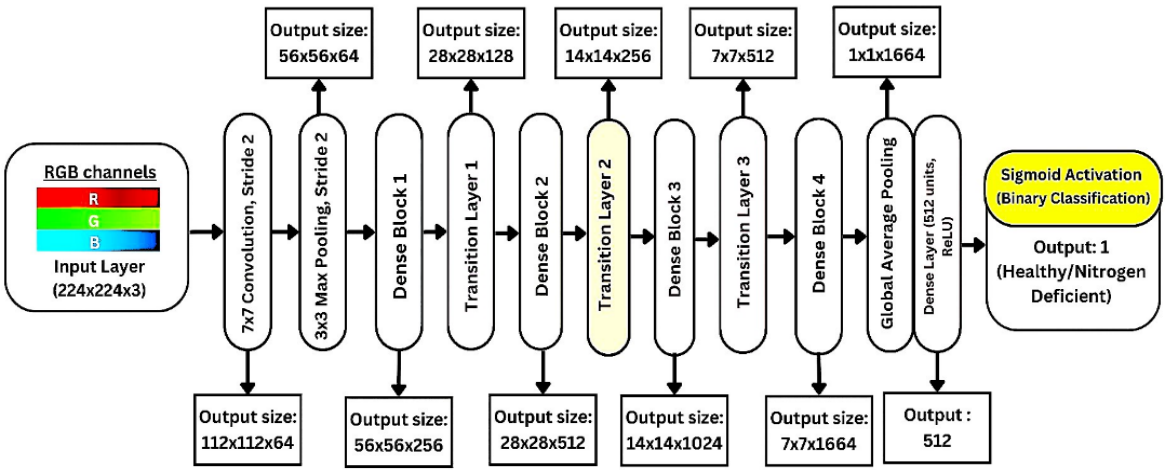


Fig. 3. System Architecture

Initially the pre-processed images of size 224x224 pixels, with 3 channels (Red, Green and Blue) are applied at the input. Each image can be seen as a tensor with dimension.

$$I \in R^{224X224X3} \tag{6}$$

The first layer is a 7x7 convolution with a stride of 2 Convolution applies filters to the input image to extract features like edges, textures, etc, that are foundational for more complex feature detection in deeper layers. Stride of 2 reduces the feature map size, while maintaining the most crucial information, saving computation and memory resources for later layers. For a filter size of F=7, stride S=2, and assuming padding P=3.The output feature map size is:

$$\frac{\text{Output width}}{\text{height}} = \frac{(224-7+2\times6)}{2} + 1 = 112 \tag{7}$$

Thus, the output size after the first convolution is 112 x 112 x 64, assuming 64 filters. Mathematically, each element of the output can be written as:

$$O(i,j,k) = \sum_{c=1}^{3}\sum_{m=1}^{3}\sum_{n=1}^{7} I(i+m, j+n, c).W(m,n,c,k) + b_k \tag{8}$$

Where: I is the input tensor,
W is the filter,
bk is the bias term for the kth filter.

A 3x3 max-pooling layer with a stride of 2 reduces the spatial dimension further, down sampling the output from the previous convolution. The max pooling operation reduces the spatial dimensions from 112x112 to 56x56, summarizing the most important information from each region of the feature map. This helps the model become spatially invariant, meaning it can still recognize features even if they shift slightly within the image. By reducing the spatial dimensions, it also reduces the computational load and helps avoid overfitting.

$$\frac{\text{Output width}}{\text{height}} = \frac{(112-3)}{2} + 1 = 56 \tag{9}$$

So, after max pooling, the output is 56 x 56 x 64.

In this model, each layer in a dense block receives input from all previous layers within that block. This dense connectivity allows the model to re-use features learned earlier in the block and combine them in new ways, which strengthens feature propagation and reduces the risk of vanishing gradients during training. The number of additional feature maps added at each layer is determined by the growth rate "k". A higher growth rate means more complex features are being captured, while a lower rate encourages feature sharing across layers. In essence, each block generates a richer, hierarchical understanding of the input. Each dense block layer's output H_l is computed as:

$$H_l = F[H_0, H_1, \ldots H_{l-1}] \quad (10)$$

Where:
F is the function applied by the convolution layers and $[H_0, H_1, \ldots H_{l-1}]$ is the concatenated output of all preceding layers.
Each layer applies Batch Normalization (BN), ReLU Activation, 1x1 Convolution to reduce dimensionality, 3x3 Convolution helps to extract the features.The output size keeps increasing in the channel dimension but remains spatially 56 x 56. After the entire dense block, if we have, say, 6 layers and growth rate k=32k, then output would be,

$$56x56\times (64+6\times32) = 56\times56\times256 \quad (11)$$

Transition Layers connects two dense blocks with 1x1 convolution. It reduces the number of feature maps output from the dense block, which keeps the model computationally efficient by controlling the number of parameters. It acts as a dimensionality reduction technique before further processing. Here 2x2 Average Pooling reduces the spatial size of the feature maps, which further simplifies the data without losing much important information. Average pooling retains a smoother summary of the image compared to max pooling, making it useful for tasks where fine detail may not be critical. The reduction of 50% in depth is obtained to produce output width to height ratio of 28. Hence the data obtained will be of size 28x28x128.Same purpose is served by all transition layers.
In dense block 2, more advanced, mid-level features are extracted. Since it's deeper in the network, the layers can focus on more complex patterns, such as detecting shapes, textures, or higher-level patterns specific to cotton leaves (e.g., edges of healthy vs. nitrogen-deficient leaves). The dense connections help propagate these mid-level features while avoiding the loss of earlier low-level features, ensuring feature re-usability. The output produced will be of size 14x14x512.
By this point in the network, the model is focusing on high-level features that are more abstract and more related to the overall structure of the object such as patterns of spots, textures, or shapes characteristic of nitrogen deficiency in the cotton leaves. Dense block 3 helps to develop a rich, hierarchical understanding of the image that captures fine details like the texture, which is often key to diagnosing leaf health. The dense block and transition layer reduces the special size down to 7x7x1024.
The final dense block continues to build on the previous block's' features, combining all the learned knowledge of low, mid, and high level features. By the time the image passes through this block, the model has extracted a rich set of hierarchical features that capture all essential aspects of leaf health or nitrogen deficiency. The concatenation of all previous layers' outputs ensures the model can access all the learned features, which makes it more powerful and efficient at classification tasks. This block operates at 7x7x1024. The growth rate continues to add more channels, considering six layers the output will be 7x7x1664.
The global average pooling layer reduces 7x7x1664 tensor in to 1x1x1664 tensor by averaging each feature map over all spatial locations.

$$O_k = \frac{1}{7x7}\sum_{i=1}^{7}\sum_{j=1}^{7} I(i,j,k) \quad (12)$$

Dense Layer is fully connected layer introduces a non-linear transformation with ReLU activation, this allows the network to learn complex, non-linear relationships across the extracted features. It also compresses the feature vector from 1664 to 512 units, retaining the most critical information for classification. This layer serves as the final integration point for all the learned features before making the final classification. A fully connected layer with 512 units is applied to 1664 dimensional output of global average pooling Each neuron computes weighted sum of its input followed by ReLU activation.

$$f(Z_i) = max(0, Z_i) \quad (13)$$

Where

$$Z_i = \sum_{j=1}^{1664} w_{ij} \cdot O_j + b_i$$

It produces the output of size 1x512.

The output layer has a sigmoid activation function and is a single unit that classifies healthy and nitrogen-deficient leaf in binary. Sigmoid defined by Eq. (13) ensures that the output is interpretable as a probability, and it's suitable for binary classification problems. The model is trained to achieve the results depicted in the next section.

5. RESULTS AND DISCUSSION

An exact number of test samples are processed through Open CV to estimate the percentage deficiency of nitrogen in the cotton leaf using color thresholding, As per the symptoms every leaf produces measurable pale yellow color. The percentage of yellow pixels is measured using yellow mask. This data is then processed for validation through DenseNet-121 in terms of accuracy. The results for analysis of percentage nitrogen deficiency are shown in Fig.4. The percentage yellow colour present in the sample tasted is 36.07%.

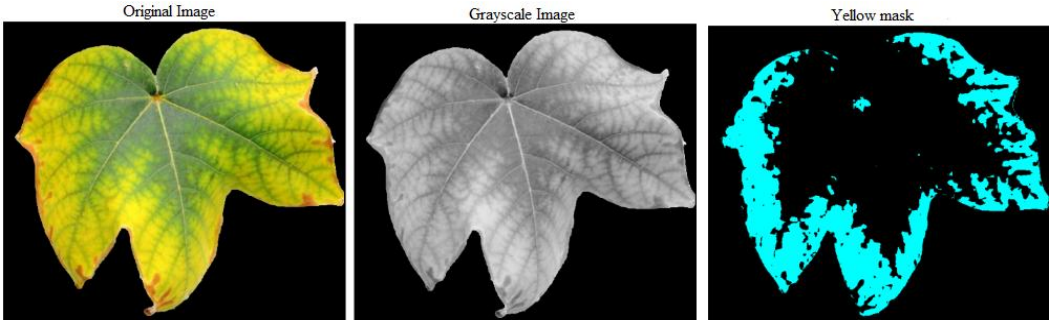
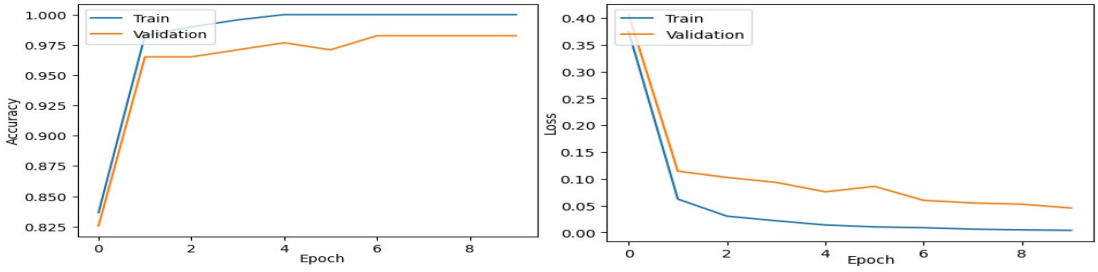
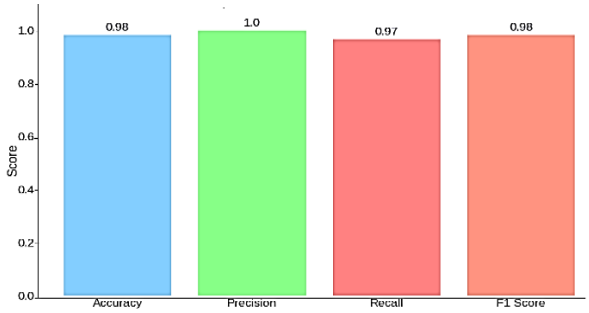


Fig. 4. Results of yellow color analysis using OpenCV

To estimate the model efficiency for nitrogen deficiency analysis of cotton leaf from images using DenseNet-121, this research uses different performance metrics. The accuracy of the model was computed based on correct prediction made from total test samples. These metrics are computed by training and evaluating the suggested model. It results in the classification of two classes as nitrogen deficient or healthy leaf. This study uses more than 400 test images to evaluate the model's recognition accuracy. Fig. 5 (a), (b) and (c) shows the findings. The proposed model obtained the best average accuracy of 98.26%. In the proposed model, the validation loss is 0.0455, the precision is 100%, the recall value is 96.67%, and an F1 score of 0.98 is observed. Similarly, one can use the different mask to analyze the other nutrients deficiencies in various crops.



(a) Model loss vs epochs (b)) Model Accuracy vs epochs.



(c) Model Performance

Fig. 5. (a) Model loss vs epochs. (b) Model Accuracy vs epochs. (c) Model Performance metrics.

6. CONCLUSION AND FUTURE WORKS

The nutrient analysis of any leaf starts with the visual symptoms appearing on the leaf or on the part of the crop. In this study symptoms corresponding to Nitrogen deficiency is considered. This model gives accuracy of 98.26%. A similar approach can be used for the micronutrient analysis, considering the water and soil conditions to achieve more accurate results. Deep learning models are robust, accurate and efficient. Excellent results are observed in several computer vision and image analysis tasks using deep learning models. The outcomes demonstrated in the suggested method uses data augmentation and a DenseNet-121 model, maximized the prediction rate using limited training examples. Subsequent efforts will streamline the DenseNet-121 design by incorporating edge devices with improved features.

REFERENCES:

1. J. A. Silva and R. Uchida. (2000). Essential nutrients for plant growth: Nutrient functions and deficiency symptoms. *Plant Nutrient Management in Hawaii's Soils*, College of Tropical Agriculture and Human Resources, University of Hawaii at Manoa (pp.35-55).
2. Prakash A. H., and Khader S. E. S. A., (2007): Nutritional and physiological disorders of cotton. In *Model Training Course on Cultivation of Long-Staple Cotton (ELS)*, Central Institute for Cotton Research, Regional Station, Coimbatore, (pp.201-206).
3. Yasin Kaya, Ercan Gürsoy., (2023): A novel multi-head CNN design to identify plant diseases using the fusion of RGB images. Ecological Informatics, Volume 75.
4. A. Kaur, V. Kukreja, M. Kumar, A. Choudhary and R. Sharma.,(2024), A Fine-Tuned DenseNet Model for an Efficient Maize Leaf Disease Classification, IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, pp. 1-5.
5. G. Wu et al., (2020): Nitrogen Status Assessment for Multiple Cultivars of Strawberries Using Portable FT-NIR Spectrometers Combined with Cultivar Recognition and Multivariate Analysis. *IEEE Access, vol. 8, pp. 126039–126050.*
6. A. B. Naeem, B. Senapati, A. S. Chauhan, J. C. Orosco, Gavilan, and W. M. F. Abdel-Rehim, (2023): Deep Learning Models for Cotton Leaf Disease Detection with VGG-16. *International Journal of Intelligent Systems and Applications in Engineering, vol.11(2), pp. 550–556.*
7. X. Liang, (2021): Few-shot cotton leaf spots disease classification based on metric learning. *Plant Methods, vol. 17(1),1-13.*
8. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov,(2012): Improving neural networks by preventing co-adaptation of feature detectors.*https://arxiv.org/pdf/1207.0580.*
9. K. P. Ferentinos, (2018): Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture, vol. 145, pp. 311–318.*
10. Sathian Dananjayan, Yu Tang, Jiajun Zhuang, Chaojun Hou, Shaoming Luo, (2022): Assessment of state-of-the-art deep learning based citrus disease detection techniques using annotated optical leaf images. *Computers and Electronics in. Agriculture, vol.193.*
11. D. Shah, V. Trivedi, V. Sheth, A. Shah, and U. Chauhan, (2022): ResTS: Residual Deep interpretable architecture for plant disease detection. *Information Processing in Agriculture, vol. 9(2), pp. 212–223.*
12. N. Ganatra and A. Patel, (2020): A Multiclass Plant Leaf Disease Detection using Image Processing and Machine Learning Techniques.*International Journal on Emerging Technologies vol.11(2),1082-1086.*
13. N. Shah, S. Jain, (2019): Detection of Disease in Cotton Leaf using Artificial Neural Network. *IEEE,473-476.* journal-article, 2019.
14. N. Khan, (2019): Role of proper management of nitrogen in cotton growth and development. *International Journal of Biosciences, pp. 483–496.*
15. P. Udawant and P. Srinath, (2022): Cotton Leaf Disease Detection Using Instance Segmentation. *Journal of Cases on Information Technology, vol. 24(4),pp. 1–10.*