Crops Leaf Disease Recognition From Digital and RS Imaging Using Fusion of Multi Self-Attention RBNet Deep Architectures and Modified Dragonfly Optimization

SOMASHEKHAR NAIK K
P25UV23T113011
WEB TECHNOLOGIES
II sem M.Tech

Under the Guidance of

Dr.PUSHPA C N
Associate Professor
Dept. of Computer Science and Engineering
University of Visvesvaraya College of Engineering

Table of Contents

1. Abstract
2. Introduction
3. Problem Statement
4. Literature Survey
5. Methodology
6. RBnet
7. MDO
8. Result
9. Conclusions
10. References

Abstract

- Globally, pests and plant diseases severely threaten forestry and agriculture. Plant protection could be substantially enhanced by using non-contact, extremely effective, and reasonably priced techniques for identifying and tracking pests and plant diseases across large geographic areas.
- The proposed method is evaluated on two publicly available crop disease images, such as the cotton, wheat, and EuroSAT datasets. For both crops, the proposed method obtained a maximum accuracy of 98.60the proposed method obtained an accuracy of 83.10to the results with recent techniques, the proposed method shows improved accuracy and precision rate.

Introduction

- Food shortage is observed due to the attack of different pests and diseases on crops, bad weather conditions, and the timely detection and eradication of disease in plants
- The core objective of this research is to build a lightweight CNN-based model that can be utilized for the training of crops disease as well as remotely acquired datasets. In addition, to overcome the problems faced by using the existing CNN architectures

Problem Statement

- Early and accurate detection of crop leaf diseases is crucial for preventing crop yield loss and ensuring food security. Traditional disease recognition methods rely on manual inspection, which is time-consuming and prone to errors. This paper proposes an automated deep learning approach using Multi-Self-Attention RBNet architectures combined with Modified Dragonfly Optimization to enhance classification accuracy.
- Main Aim is to build a lightweight CNN model that gives great accuracy than existing one

Literature Survey

Author(s)	Study Title	Key Findings	Year
Shafi et al. [48]	Wheat crop disease classification using RS devices and ML models	Proposed an ML architecture combining RS data (satellite, UAV) and handheld devices for wheat disease detection. Demonstrated that the performance depends on the dataset quality and image acquisition method.	2022
Xu et al. [55]	Wheat leaf disease identification based on deep learning algorithms	Introduced a deep learning model combining residual channel attention and CNN to classify healthy and diseased wheat leaves. Achieved 98.83% accuracy.	2023
Magsi et al. [57]	Disease severity level identification of cotton plants using ML techniques	Focused on detecting severity levels of cotton leaf diseases with ML. Achieved 89.40% accuracy using texture and color features.	2021

Figure: Litrature Survey

Methodology

Deep Learning Model :

The paper introduces RBNet, a deep network enhanced with multi-self-attention mechanisms to capture complex patterns in diseased and healthy leaf images.

Optimization Technique :

A modified version of Dragonfly Optimization (MDO) is used to fine-tune the network's hyperparameters, improving convergence speed and classification performance.

Data Fusion :

The model integrates information from both digital images and remote sensing (RS) images to improve disease recognition accuracy

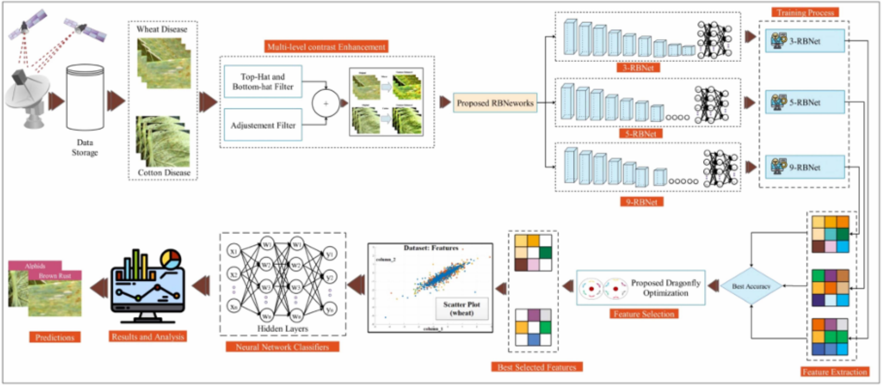
Cont...

Feature Extraction:

High-dimensional features are extracted using self-attention mechanisms, ensuring that critical patterns are retained.

Training Testing:

The proposed model is trained on a large dataset containing various leaf diseases, followed by performance evaluation.



Proposed framework for the classification of crops leaf diseases using deep learning and optimization.

Figure: Archietecture

Cont...

Architecture	Layer Details	Number of Layers	Working Mechanism
ResNet-3	1. Convolutional Layer (3x3, stride 1)	3	- A simple network with residual connections. This model uses a small number of layers, each with convolutional filters and a residual connection to prevent degradation.
	2. Residual Block (2 Convolutional Layers)		- Useful for very simple tasks or as a proof of concept.
	3. Fully Connected (FC) Layer		
ResNet-5	1. Convolutional Layer (7x7, stride 2)	5	- Similar to ResNet-3 but with more layers, allowing for more complex features to be learned. Includes an additional convolutional layer before the residual block.
	2. Residual Block (2 Convolutional Layers)		- Residual connections help avoid vanishing gradients by learning the identity function in deeper layers.
	3. Fully Connected (FC) Layer		
ResNet-9	1. Convolutional Layer (7x7, stride 2)	9	- More complex than ResNet-3 and ResNet-5, with additional residual blocks. Suitable for more complex datasets, learning deeper representations.
	2. 2 Residual Blocks (2 Conv. Layers each)		- Involves more layers and allows the model to extract richer features through deeper residual blocks.
	3. Fully Connected (FC) Layer		

Figure: Layers in Residual network

Algorithm

text Copy code

```
1. Initialize network parameters (number of layers, filters, input size)

2. Input: image X (shape = (H, W, C))

3. Apply initial convolution:
   - Conv2D(7x7, stride=2, padding=3) on X
   - BatchNorm
   - ReLU
   - MaxPool(3x3, stride=2, padding=1)

4. For each residual block:
   For each layer in block:
       - Conv2D(3x3)
       - BatchNorm
       - ReLU
       - Conv2D(3x3)
       - BatchNorm
       - If input size != output size:
           - Apply 1x1 convolution to input
       - Add skip connection (input + output)
       - ReLU on combined output

5. After residual blocks:
   - Apply Global Average Pooling

6. Apply Fully Connected Layer:
   - FC(output size = number of classes)

7. Apply Softmax activation for output class probabilities

8. Return final output
```

Rbnet

RBNet is built on the idea of residual learning, which allows the network to learn modifications (residuals) to the input rather than the entire transformation. The core mathematical representation for a single residual block in RBNet is:

$$\phi_{output} = \Phi_{act}(\phi_{Conv}(i) + i)$$

where:

- i is the input to the residual block,
- $\phi_{Conv}(i)$ represents the convolutional operation (possibly including multiple convolution layers, batch normalization, etc.), and
- Φ_{act} is a non-linear activation function (like Re

- This equation shows that the block adds the input i to the convolution output ( ) Conv (i) before applying the activation, allowing the network to learn residual functions. Stacking multiple such residual blocks in RBNet helps in building very deep networks while mitigating the vanishing gradient problem, and when combined with a self-attention mechanism, it further enhances the ability to capture important spatial dependencies in the data.

Rbnet

- RBNet is built on the idea of residual learning, which allows the network to learn modifications (residuals) to the input rather than the entire transformation. The core mathematical representation for a single residual block in RBNet is:

  $$\phi_{output} = \Phi_{act}(\phi_{Conv}(i) + i)$$

  where:

  - i is the input to the residual block,
  - $\phi_{Conv}(i)$ represents the convolutional operation (possibly including multiple convolution layers, batch normalization, etc.), and
  - Φ_{act} is a non-linear activation function (like Re'

- This equation shows that the block adds the input i to the convolution output ( ) Conv (i) before applying the activation, allowing the network to learn residual functions. Stacking multiple such residual blocks in RBNet helps in building very deep networks while mitigating the vanishing gradient problem, and when combined with a self-attention mechanism, it further enhances the ability to capture important spatial dependencies in the data.

MDO

- The MDO algorithm is inspired by the swarming behavior of dragonflies and is used to optimize feature selection and hyperparameters in the model. Its operation is based on mimicking two main behaviors: exploration (searching for promising regions in the solution space) and exploitation (refining solutions).

1. **Separation (Avoiding overcrowding):**

$$E_i = -\frac{1}{Z}\sum_{b=1}^{Z}(Y - Y_b)$$

- Y is the current position of a dragonfly (solution),
- Y_b is the position of the b-th neighbor,
- Z is the total number of neighboring dragonflies.

3. **Cohesion (Moving toward the center of mass of neighbors):**

$$O_i = \frac{1}{Z}\sum_{b=1}^{Z} Y_b - Y$$

- This encourages the dragonfly to move toward the average position of its neighbors.

4. **Attraction Toward Food (Guiding toward the optimal solution):**

$$V_i = Y + (Y^{+} - Y)$$

5. **Distraction from Enemy (Avoiding undesirable solutions):**

$$D_i = Y - Y^{-}$$

- Y^{-} is the position of the enemy (a solution to be avoided).

A. Results of the Proposed 3-RBNet Self

The classification results of the proposed 3-RBNet on wheat dataset have been presented.The proposed 3- RB-Net is implemented on wheat and cotton data sets to validate its performance. The table shows the maximum accuracy 96.40% on wide NN. Furthermore, few other measures are also computed such as a sensitivity rate of 93.38, precision rate of 94.46, kappa value of 0.8868,and MCC value of 0.9292,respectively

TABLE VII
RESULTS OF THE PROPOSED 3-RBNET ON WHEAT DATASET

Method	Sensitivity (%)	Precision (%)	FNR (%)	Accuracy (%)	Time (s)	Kappa	MCC
Narrow NN	92.92	94.20	7.08	95.80	13.632	0.8699	0.9239
Medium NN	92.58	92.84	7.42	95.70	10.196	0.8669	0.9155
Wide NN	**93.38**	**94.46**	**6.62**	**96.40**	**12.419**	**0.8868**	**0.9292**
Bi-layered NN	92.42	92.38	7.58	95.80	15.309	0.8699	0.9128
Tri-layered NN	93.08	92.58	6.92	95.80	27.711	0.8684	0.9165

The bold values denote the best results.

B. Results of the Proposed 5-RBNet Self Table IX shows the results of 5-RBNet Self architecture on the wheat dataset. The table shows that the narrow NN achieved anaccuracy of95.90%,the medium NN obtained an accuracy of 96.30%, the wide NN obtained an accuracy of 96.70%, 96.70% accuracy obtained by bilayered NN, and 96.20% accuracy is achieved by TNN classifier, respectively

TABLE XIII
RESULTS OF THE PROPOSED DRAGONFLY OPTIMIZATION ON WHEAT DATASET

Classifiers	Sensitivity (%)	Precision (%)	FNR (%)	Accuracy (%)	Time (s)	Kappa	MCC
Narrow NN	98.42	98.44	1.52	**98.60**	4.7342	0.955	0.980
Medium NN	98.04	97.36	1.96	98.20	3.4632	0.944	0.972
Wide NN	98.28	97.58	1.72	98.30	2.9939	0.948	0.974
Bi-layered NN	96.92	96.84	3.08	98.10	3.2534	0.941	0.963
Tri-layered NN	97.34	97.46	2.66	98.10	3.6398	0.941	0.969

The bold value denote the best results.

Figure: Caption

C. Results of the Proposed 9-RBNet Self The classification results of the 9-RBNet Self architecture on the wheat dataset are presented in Table XI. Deep features are extracted from the self-attention layer of the trained model, and results are obtained. In this table, the obtained classification accuracy is 98.10% by narrow NN, 98.40% by medium NN, 98.40% by wide NN, 98.00% by bilayered NN, and 97.70% by trilayered NN. The medium NN accuracy is higher than that of other listed classifiers in this table. In addition, the sensitivity rate of this classifier is 98.32, with a precision rate of 98.14, Kappa value of 0.9510, and MCC value of 0.9780, respectively

TABLE XIV
RESULTS OF THE PROPOSED DRAGONFLY OPTIMIZATION ON COTTON DATASET

Methods	Sensitivity (%)	Precision (%)	FNR (%)	Accuracy (%)	Time (s)	Kappa	MCC
Narrow NN	90.60	90.88	9.40	91.20	3.231	0.5990	0.8946
Medium NN	92.46	92.52	7.54	92.90	4.583	0.7193	0.9262
Wide NN	**93.48**	**93.51**	**6.52**	**93.90**	**3.068**	**0.6770**	**0.9146**
Bi-layered NN	92.16	92.18	7.84	92.50	3.959	0.6591	0.9109
Tri-layered NN	90.68	91.02	9.32	91.30	4.680	0.6012	0.8958

The bold values denote the best results.

Conclusions

- we proposed a novel self-attention and optimization architecture for crop leaf disease classification The strength of the proposed architecture is designing two self attention 5-RBNET and 9-RBNET architectures for cotton and Visual illustration of proposed labelled images.
- wheat disease recognition The contrast enhancement technique has been proposed based on the fusion of two filter mathematical formulations and passed resultant images to proposed architectures for the training. Based on the initial accuracy, the binary dragonfly optimization algorithm was applied on 5-RBNET and 9-RBNET architectures, and the best-selected features for the classification were obtained.

References I

S. K. Upadhyay and A. Kumar, *"A novel approach for rice plant diseases classification with deep convolutional neural network,"*, Journal of Computer Science, vol. 34, no. 4, pp. 451–465, 2024.

X. Chao, *"Toward sustainability: Trade-off between data quality and quantity in crop pest recognition,"*, MIT Press, 2022.

A. Arshaghi, M. Ashourian,, *"Potato diseases detection and classification using deep learning methods,"*, vol. 22, no. 1, pp. 12–25, 2023.

S. Chakraborty and A. C. Newton, *"Climate change, plant diseases and food security: An overview,"*, Journal of Information Security, vol. 18, no. 5, pp. 98–112, 2019.

Thank You