Journal of Information Systems Engineering and Management
2025, 10(5s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

An Enhanced Oppositional Crow Search Optimization Algorithm-based Colour Edge Segmentation and Modified Resnet-39 Architecture for Prediction of Crop Disease

Rahul Kumar[1], Rajeev Paulus[2]
[1]*Department of Electronics and Communication Engineering, Sam Higginbottom University of Agriculture, Technology and Sciences, Allahabad- 211007. rahkam@gmail.com*
[2]*Department of Electronics and Communication Engineering, Sam Higginbottom University of Agriculture, Technology and Sciences, Allahabad- 211007. rajeev.paulus@shiats.edu.in*

ARTICLE INFO

Received: 05 Oct 2024

Revised: 10 Dec 2024

Accepted: 22 Dec 2024

ABSTRACT

Accurate and timely prediction of crop diseases is vital for improving agricultural productivity and ensuring food security. This research work proposes new framework that integrates an improved swarm-based color edge segmentation method with a modified ResNet-39 architecture to efficiently detect and categorize crop diseases. The suggested segmentation technique utilizes Crow Search optimization algorithm combined with oppositional learning to improve edge recognition accuracy using color segmentation. This allows precise localization of disease-affected areas even under difficult situations such as poor contrast and noise. The segmented outputs are further processed using a tailored ResNet-39 architecture, optimized for computational efficiency and prediction accuracy. The proposed Enhanced edge segmentation Crow Search Optimization with ResNet-39 (EesCSO-ResNet-39) methodology is tested using publically accessible crop disease datasets, showing substantial improvements in critical metrics such as accuracy, precision, and recall relative to conventional segmentation techniques and current deep learning architectures, including ResNet-50 and VGG-16.

Keywords: Modified ResNet-39, Oppositional Crow Search algorithm, edge segmentation, Precision agriculture, crop disease prediction.

INTRODUCTION

In recent decades, developing country's economy is relying highly on agriculture. Agriculture is the main source of income and the foundation of most rural communities of developing countries. In addition to supplying raw materials for many other types of businesses, a robust agricultural industry guarantees food security. But climate change, falling pricing, and soil deterioration are only a few of the recent problems that farmers have had to deal with. Among these difficulties, crop infections are particularly dangerous since they stunt crop development and ruin harvests. One potential answer is the use of sophisticated vision technology in agriculture. These technologies may help with early disease prediction and can greatly increase agricultural yields [20].

Prediction of crop diseases using traditional approaches requires constant professional observation and feedback from farmers. Because farmers could misunderstand symptoms or provide wrong facts, these methods are prone to mistakes that professionals can't help but fix. With the development of automated crop disease detection systems, early disease detection has become easier and requires much less human involvement. To detect diseases early on, these systems use digital camera pictures of crop leaves and machine learning (ML) algorithms. As a result of this automation, agricultural output rises, human error falls, and disease detection becomes more efficient [21]. These systems improve crop quality and production while saving time and costs by using digital cameras and ML algorithms. This eliminates the need for intensive expert monitoring.

In the field of crop disease prediction, technologies such as segmentation and object identification are used. The identification of sick regions in crop leaves has been the subject of investigation by a number of researchers [22]. These researchers have used segmentation methods such as edge detection, clustering, smoothing, and boundary detection. One of the most popular approaches is K-means clustering, which has grown more popular due to its

speed and accuracy in the process of grouping enormous information. On the other hand, despite its simplicity, the K-means algorithm has certain inherent restrictions due to the fact that it is dependent on linear boundaries. There have been developments in methods for the generation of nonlinear boundaries, and the results have been promising [23]. Segmentation has the potential to increase the accuracy of illness detection; yet, the strain it places on computational resources may result in a reduction in system efficiency. It is vital to do preprocessing in order to improve object identification and classification systems, particularly edge detection approaches such as Sobel edge detection [24]. As is the case with segmentation, however, the system may become overloaded if there is an excessive amount of preprocessing. As a result, designing models that maximize efficiency while reducing computational overhead is crucial [25]. In this approach, an enhanced swarm-based edge segmentation model is developed for identifying edges in color images and a deep learning model is also created for handling classification.

The following are the contributions of this research work.

- Developed a novel swarm-based mechanism integrating edge and color features for precise segmentation of diseased regions in crop leaf images, leveraging oppositional learning for improved optimization and faster convergence.
- A Modified ResNet-39 architecture optimized for crop disease prediction, achieving a balance between computational efficiency and accuracy by incorporating features from the segmentation mechanism.
- Validation on diverse datasets under real-world conditions, with a focus on practical implementation for large-scale farms, resource-constrained environments.

The structure of the paper is as follows: Section 2 reviews the related work conducted in the field of crop disease detection. Section 3 defines the problem statement. Section 4 outlines the proposed methodology. Section 5 presents the experimental results, and Section 6 concludes the study.

LITERATURE SURVEY

Transparent object segmentation is a difficult problem that many academics have attempted to solve. TransCut [1] optimized occlusion detection in 4D light field pictures to provide segmentation outputs using an energy function based on LF linear. The matting of transparent objects was tackled by TOM-Net [2] as an estimate issue for low refractive index. The difficulty in getting refraction flow graph data in real-world circumstances is a major constraint of TOM-Net's training label. Using changes in transmission characteristics to extract useful data for object recognition in transparent images, Atsuro et al. [3] presented a new three-stream semantic segmentation technique. To distinguish between internal and exterior border characteristics, Han et al. [4] presented a novel internal-external boundary attention module. They also created an edge-body full attention module that uses external boundary semantic information to oversee transparent object segmentation.

Recent years have seen the rise of transformers as a game-changing tool in computer vision applications, after their success in natural language processing (NLP). The Vision Transformer (ViT) [13] was created and used for picture categorization after transformers were successful in natural language processing (NLP), laying the groundwork for models in semantic segmentation that rely on transformers. ViT was improved by DeiT [5] with the use of knowledge distillation and powerful data augmentation methods, which decreased ViT's need on massive datasets. Several publications [6-9] have used pyramid-designed transformers with multi-scale features to improve model accuracy, drawing inspiration from the pyramid structures in convolutional networks. A multimodal sparse attention mechanism was used by SCANET [10] to enhance the model's efficiency. To create more accurate semantic models, TRPose [11] fused two scales of encoders. Spatial preservation, context acquisition, and high-level semantic extraction were divided into three simultaneous routes by MPSegNet [12], which suggested a multi-path structure with attention reweighting and multi-scale encoding. But most of these models ignore the decoder's crucial function of feature fusion in favour of encoder design. To fill this void, our method places an emphasis on efficient feature fusion inside the decoder, guaranteeing that multi-level feature information is lost to a minimum.

On the other hand a significant number of applications were solved using evolutionary algorithms such as Wireless Sensor Networks [14, 16, 17], Vehicular Ad-Hod networks [15], Travelling salesman problem [18], Methane emission estimation [19] and so on. On these grounds in this research work, the crow search algorithm has been used for edge optimization purpose.

PROBLEM DEFINITION

One of the most important preprocessing steps in image analysis is segmentation, which entails dividing a picture into relevant parts for easier analysis. Because they often include important markers of disease presence like discoloration, spots, or structural anomalies, edges and color-based areas must be accurately identified for applications like crop disease detection.

But the problems that conventional methods of edge detection and segmentation have are: It may be rather tough to accurately identify edges in crop photos due to the complex textures, overlapping structures (such as leaves, stems, and fruits), and distracting environmental factors like shadows, dirt, or water droplets. When it comes to disease diagnosis, generic algorithms that rely just on intensity-based edge detection tend to miss segment areas with subtle color changes. Traditional segmentation techniques provide less-than-ideal feature extraction since they aren't designed to work in tandem with deep learning models for jobs like disease prediction.

PROPOSED METHODOLOGY

In this section the proposed enhanced image segmentation using swarm-based algorithm and modified ResNet-39 architecture is described.

4.1 Swarm based color edge segmentation

In this section, the image preprocessing techniques and the proposed swarm-based edge segmentation for color images are described.

4.1.1 Image preprocessing

The first stage is obtaining the leaf picture and doing pre-processing to highlight essential elements. Image scaling, the use of smoothing techniques, and noise reduction are some of the tasks that are included in this operation. The goal is to improve clarity. The employment of a Gaussian filter, as seen in equation (1), is one method that may be utilized to achieve smoothing.

$$Img_s(i,j) = \frac{1}{2\times\pi\times\sigma^2}\, e^{\left(-\frac{i^2+j^2}{2\times\sigma^2}\right)*Img(i,j)} \quad (1)$$

4.1.2 Color conversion

In the next step, the RGB picture is converted into the CIE Lab colour space, where the letter L represents the degree of brightness and the letters x and y represent the degree of chromaticity. Additionally, the identification of colour edges is made easier by this change. There are three channels that are used in the CIE Lab colour space in order to separate colour information. Through the process of divorcing colour from intensity, this approach improves edge recognition and makes it easier to identify changes in colour. The equations (2) through (4) provide a formal outline of the process of conversion.

$$L^* = 116 \times f\left(\frac{J}{J_n}\right) - 16 \quad (2)$$

$$x^* = 500\left[f\left(\frac{I}{I_n}\right) - f\left(\frac{J}{J_n}\right)\right] \quad (3)$$

$$y^* = 200\left[f\left(\frac{I}{I_n}\right) - f\left(\frac{K}{K_n}\right)\right] \quad (4)$$

where $I, J, and\ K$ are reference values.

4.1.3 Computation of Gradience

In order to identify edges, the amplitude of the gradient is determined for each color channel according to equations (5) to (7).

$$G_L(i,j) = \sqrt{\left(\frac{\partial L^*}{\partial i}\right)^2 + \left(\frac{\partial L^*}{\partial j}\right)^2} \quad (5)$$

$$G_x(i,j) = \sqrt{\left(\frac{\partial x^*}{\partial i}\right)^2 + \left(\frac{\partial x^*}{\partial j}\right)^2} \quad (6)$$

$$G_y(i,j) = \sqrt{\left(\frac{\partial y^*}{\partial i}\right)^2 + \left(\frac{\partial y^*}{\partial j}\right)^2} \quad (7)$$

And the computation of single gradient of the image is calculated as

$$G(i,j) = \sqrt{G_L(i,j)^2 + G_x(i,j)^2 + G_y(i,j)^2} \quad (8)$$

4.1.4 Threshold calculation

The application of thresholding to the combined gradient magnitude picture, as indicated in equation (9), results in the generation of a binary edge map. Through the use of adaptive thresholding, the threshold value is constantly adjusted depending on local fluctuations in illumination and contrast, which ultimately results in segmentation that is more precise. In contrast to a single global threshold, this technique successfully controls irregularities in lighting and various textures, which results in an improvement in the recognition of features in pictures that are composed of several layers.

$$T(i,j) = \mu_G(i,j) + k \times \sigma_G(i,j) \quad (9)$$

4.1.5 Refinement of edges

In the field of image processing, morphological procedures such as dilation and erosion assume an extremely important significance. For the purpose of improving edges by reducing noise and filling in gaps, these techniques are absolutely necessary. Through the process of erosion, the size of foreground objects is decreased, and sharp edges are smoothed down, which ultimately results in a detection layer that is thinner. By filling up empty areas and joining segments that were previously separated, dilatation, on the other hand, contributes to the expansion of boundaries. It is essential to do these processes in order to enhance the accuracy of detecting systems. Dilation is used to guarantee that the representation is continuous and linked, while erosion is used to eliminate superfluous noise and surplus information from the boundaries of the representation. Morphological methods such as erosion and dilation are used in order to efficiently fill gaps, reduce noise, and improve edges; these techniques are also utilized.

$$E(Img) = Img - B \quad (10)$$

$$D(E) = E + B \quad (11)$$

where E *and* Img are the eroded and dilated image.

4.2 Oppositional Crow Search algorithm for edge optimization

4.2.1 Learning via Oppositional search

By taking into account the present answer and its inverse concurrently, Oppositional Learning (OL) improves the search process' efficiency [24]. By increasing the size of the search space, this method speeds up the convergence process. In order to find the best possible solutions, OL compares a certain hyperparameter configuration with its inverse. Take X as an example. If x and y are the lower and upper limits of X, the formula

$$X_o = x + y - X \quad (12)$$

may be used to get the opposite number X_0.

4.2.2 Algorithm of Crow Search

The CSA takes its cues from the crow's ingenious behavior, including its habit of storing food in inconspicuous locations and then finding it at a later time [25]. Potential solutions, or "crows" in this technique, are really just collections of edges of the image. The crows remember their finest "hiding spot" (the greatest solution they've found so far) and utilize that knowledge to find even better ones. Exploration, probing uncharted territory, and exploitation, or honing down on promising leads, are both essential parts of the search process. The equation determines the new solution location for the following iteration:

$$P_i = P_i + r_3 \times (M_i - P_i) \quad (13)$$

In this case, P stands for the present solution, while M is a randomly chosen solution from the population. Because of this adaptive strategy, the algorithm can traverse the solution space with ease.

4.2.3 Oppositional Crow Search Algorithm

Crow Search Algorithm (CSA) when combined with Oppositional Learning might significantly enhance the process of optimizing image edge recognition. The identification of sudden intensity changes is essential to the processes of segmentation and feature extraction. Image edge recognition is dependent on this detection. In contrast, in order to

achieve accurate edge detection, it is often essential to make adjustments to a variety of hyperparameters, such as the settings for the filter and the thresholds. This challenge is addressed by the Oppositional Crow Search Algorithm (OCSA), which incorporates the most advantageous aspects of both the OL and CSA algorithms. OL accelerates the convergence of the optimization process by simultaneously evaluating two viable solutions to the problem at hand. Being able to explore a bigger search space owing to its dual evaluation technique, the method has a greater probability of discovering the optimal parameter configurations for edge detection. This is because the method can search a larger search space. When used to the processing of images, OL enables the algorithm to swiftly converge on configurations that enhance the sharpness of edges while simultaneously reducing noise.

The Crow Search Algorithm, which has the inherent ability to strike a balance between exploring and exploiting the search area, is a tool that helps to improve this technique. In the context of OCSA, the term "crows" refers to groupings of edge-detection settings that include probable edges. In order to preserve the finest solutions that have been found up to this point, each crow makes use of its memory. Additionally, it searches for new configurations in a dynamic manner, taking into account its location in relation to other solutions in the population. OCSA identify robust edge pixel selections that operate well under a variety of visual circumstances, but it also ensures that local minima are avoided by including opposing solutions into the process of updating the crow's memory. When it comes to optimizing edge recognition in complicated pictures, our hybridized technique really shines. Parameter tuning is key due to the intricate interaction between small gradients and noise. By combining OL with CSA, we can optimize picture edges in a precise and adaptable manner, which will lead to better results in computer vision applications.

The objective function is defined as

$$f(E) = \sum_{(a,b)\in E}\big(\nabla Img(a,b).C(a,b)\big) - \lambda.\sum_{(a,b)\in \partial E}\big(D(a,b)\big) \qquad (14)$$

Algorithm: EesOCSO for edge optimization

LB, UB: Lower and upper bounds of the solution space.
f: Objective function for edge detection (e.g., edge clarity score).
N_P: Population size (number of crows).
I_T : Maximum number of iterations.
t : Current iteration, initialized to 0.
d: Dimension of the solution (parameters for edge detection, e.g., threshold values).

Begin:

// *Initialize the Population*
For each crow i in 1: N_P
For each dimension j in 1: d
$P_{\{i,j\}} = LB_{\{j\}} + (UB_{\{j\}} - LB_{\{j\}}) * rand()$
End for
End for
// *Compute Oppositional Solutions*
For each crow i in 1: N_P
For each dimension j in 1: d
$P_{\{o_{\{i,j\}}\}} = LB_{\{j\}} + UB_{\{j\}} - P_{\{i,j\}}$
End for
End for

// *Evaluate Fitness:*
For each crow i in 1: N_P
$F_i = f(P_i), F_{\{o_i\}} = f(P_{\{o_i\}})$
$P_i = argmax(F_i, F_{\{o_i\}})$
End for

// Initialize Best Solutions
Set $B_P = P$ (initial best solutions are the current population).

while $t <= I_T$
For each crow i in 1: N_P
$M = P(randi(1: N_P))$
Update the position of the crow
if $r_1 < r_2$
$P_i = P_i + r_3 * (M_i - P_i)$
Else
$P_T = P_i + r\_3 * (M_i - P_i)$
$P_i = LB + UB - P_T$
End if
End for

// Evaluate Fitness for Updated Population
For each crow i in 1: N_P
$F_i = f(P_i)$
End for.

// Update Best Solutions
For each crow i in 1: N_P
if F_i is better than the previous best
$B_{\{P_i\}} = P_i$
End for.

Increment t by 1.

End

Output: B_P

Algorithm 1: EesOCSO for edge optimization

4.3 Modified ResNet-39 for Classification

ResNet-39 is a customized version of the ResNet family that was built for use in circumstances when it is of the highest significance to strike a balance between the efficiency of computational operation and the precision of the results. The high computational cost of classic deep designs, such as ResNet-50, renders them unsuitable for use in real-time applications or in environments with limited hardware resources. However, these designs provide good accuracy, they continue to be used. There is a possibility that ResNet-34 and other shallower topologies are not capable of accurately capturing complex hierarchical properties. With the goal of bridging this gap, ResNet-39 presents a mid-tier design that achieves a balance between depth and efficiency. The system is capable of managing datasets that are relatively difficult and have sufficient capacity, while at the same time being lightweight enough to function in situations where resources are constrained, such as mobile or edge devices.

An architecture with 34 layers that is relatively deep is known as ResNet-34. This design is particularly useful for situations in which reducing processing costs is of the highest significance. Although it performs well for general-purpose photo recognition, it may not be able to handle datasets that need more in-depth feature extraction. The 50-layer ResNet-50 model, on the other hand, makes use of bottleneck layers in order to tackle complex tasks with more precision. However, this comes at the expense of increasing demands for processing and memory. A balance between the two extremes, ResNet-39 gives a better feature extraction than ResNet-34 while consuming less resources than ResNet-50. Its 39 layers offer a compromise between the two extremes. It is a feasible option to shallower models for applications that need decent processing efficiency in addition to greater accuracy. This is because of the rationale stated above.

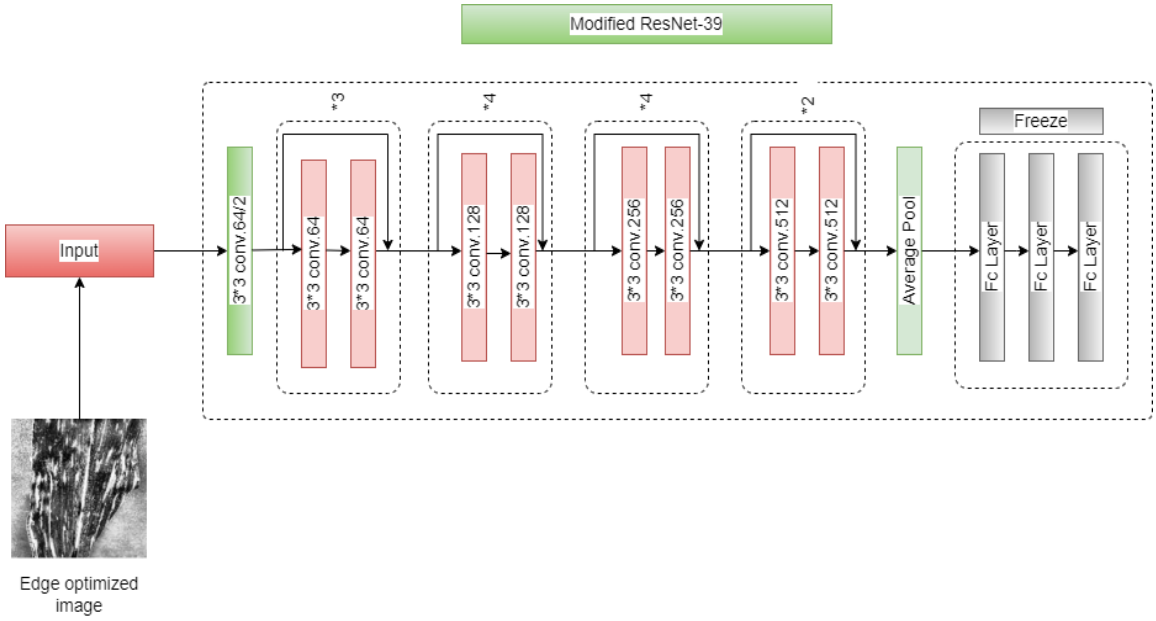

Figure 1: Modified ResNet-39 architecture

EXPERIMENTAL ANALYSIS

This section comprises of the experimental setup, performance metrices and experimental results and its interpretations.

5.1 Experimental Setup

Python 3.7 was used to construct the proposed model, and Keras and TensorFlow were used to complete the implementation of the Modified ResNet-39 architecture. Among the 4,188 photographs that comprise the collection are depictions of a variety of circumstances, such as healthy leaves, rust, blight, and grey leaf spot. The photographs were obtained from a public resource. We compare the model's results to those of Inception-v3 represented as M1, ResNet-50 as M2, VGG-19 as M3, VGG-16 as M4, and O-CSO-CNN as M5. There are ten distinct configurations in the dataset that are used for training and testing. The classification ratios are as follows: D1 is 85% training and 15% testing, D2 is 80% training and 20% testing, D3 is 75% training and 25% testing, D4 is 70% training and 30% testing, D5 is 65% training and 35% testing, D6 is 60% training and 40% testing, D7 is 55% training and 45% testing and D8 is 50% training and 50% testing. Standard performance metrices such as precision, recall, F1-Score and Accuracy are used to evaluate the performance of the proposed model.

5.2 Experimental Results

Table 1: Experimental results of EesCSO-ResNet-39 w.r.t. Precision

Precision	M1	M2	M3	M4	M5	EesCSO-ResNet
D1	93.98	96.15	96.47	95.81	98.79	99.28
D2	93.05	95.21	96.49	95.08	98.64	99.08
D3	92.69	94.66	96.39	95.27	97.65	97.87
D4	92.09	93.92	95.72	94.37	97.46	97.76
D5	91.96	93.71	94.97	93.01	96.98	97.47
D6	90.79	92.84	94.47	91.08	96.92	97.15
D7	89.94	91.57	94.53	90.65	96.11	96.33
D8	89.02	91.23	91.83	90.53	96.01	96.28

Table 1 indicates the experimental results of Precision on eight splits of the dataset. Comparing the results of EesCSO-ResNet-39 with other methods, the proposed model outperforms in most of the cases. On Comparing

results of D1, EesCSO-ResNet-39 outperforms M1 with 5.34%, M2 with 3.15%, M3 with 2.83%, M4 with 3.49% and M5 with 0.49%. On Comparing results of D2 EesCSO-ResNet-39 outperforms M1 with 6.08%, M2 with 3.91%, M3 with 2.61%, M4 with 4.03% and M5 with 0.44%. On Comparing results of D3, EesCSO-ResNet-39 outperforms M1 with 5.29%, M2 with 3.28%, M3 with 1.52%, M4 with 2.65% and M5 with 0.23%. On Comparing results of D4, EesCSO-ResNet-39 outperforms M1 with 5.8%, M2 with 3.94%, M3 with 2.09%, M4 with 3.47% and M5 with 0.31%.

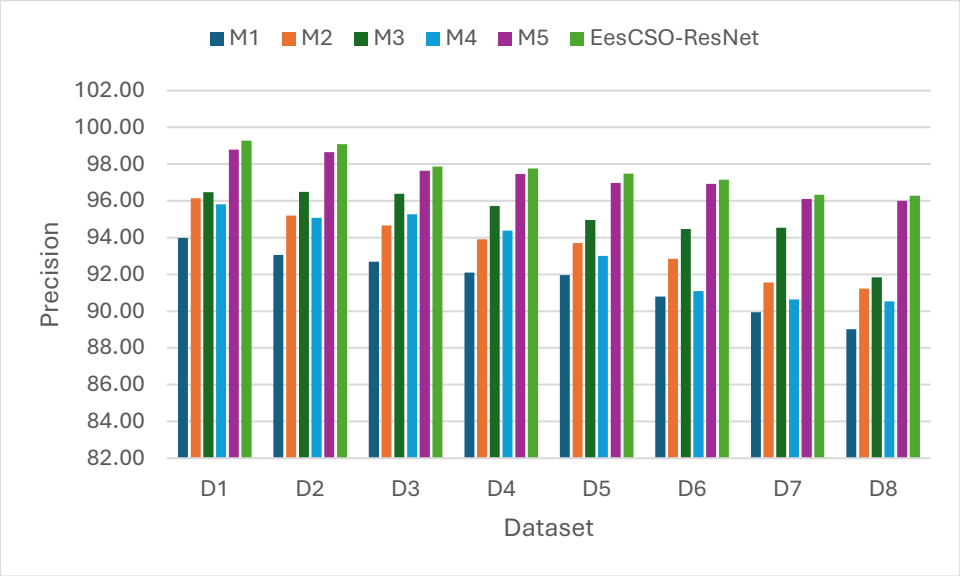

Figure 2: Graphical representation of results in Table 1

On Comparing results of D5, EesCSO-ResNet-39 outperforms M1 with 5.65%, M2 with 3.86%, M3 with 2.57%, M4 with 4.58% and M5 with 0.51%. On Comparing results of D6, EesCSO-ResNet-39 outperforms M1 with 6.55%, M2 with 4.43%, M3 with 2.76%, M4 with 6.24% and M5 with 0.23%. Comparing results of D7, EesCSO-ResNet-39 outperforms M1 with 6.63%, M2 with 4.95%, M3 with 1.87%, M4 with 5.9% and M5 with 0.23%. On Comparing results of D8, EesCSO-ResNet-39 outperforms M1 with 7.54%, M2 with 5.25%, M3 with 4.62%, M4 with 5.98% and M5 with 0.28%.

Table 2: Experimental results of O-CSO-CNN w.r.t. Recall

Recall	M1	M2	M3	M4	M5	EesCSO-ResNet
D1	89.90	95.55	93.46	92.43	97.78	98.13
D2	89.56	94.44	92.60	91.59	97.42	97.75
D3	89.48	94.34	92.30	90.27	96.57	97.02
D4	89.54	93.90	91.62	89.75	96.39	96.83
D5	88.92	93.52	89.95	89.51	96.19	96.53
D6	87.34	92.25	90.13	89.40	95.55	95.81
D7	87.38	92.08	90.05	89.03	95.48	95.95
D8	86.55	91.85	88.87	88.66	95.10	95.43

Table 2 indicates the experimental results of Recall on eight splits of the dataset. On comparing the results of EesCSO-ResNet-39 with other methods, the proposed model outperforms in most of the cases. On Comparing results of D1, EesCSO-ResNet-39 outperforms M1 with 8.38%, M2 with 2.63%, M3 with 4.76%, M4 with 5.8% and M5 with 0.36%. On Comparing results of D2 EesCSO-ResNet-39 outperforms M1 with 8.38%, M2 with 3.39%, M3 with 5.27%, M4 with 6.31% and M5 with 0.34%. On Comparing results of D3, EesCSO-ResNet-39 outperforms M1 with 7.77%, M2 with 2.77%, M3 with 4.87%, M4 with 6.96% and M5 with 0.46%. On Comparing results of D4,

EesCSO-ResNet-39 outperforms M1 with 7.53%, M2 with 3.03%, M3 with 5.39%, M4 with 7.31% and M5 with 0.45%.

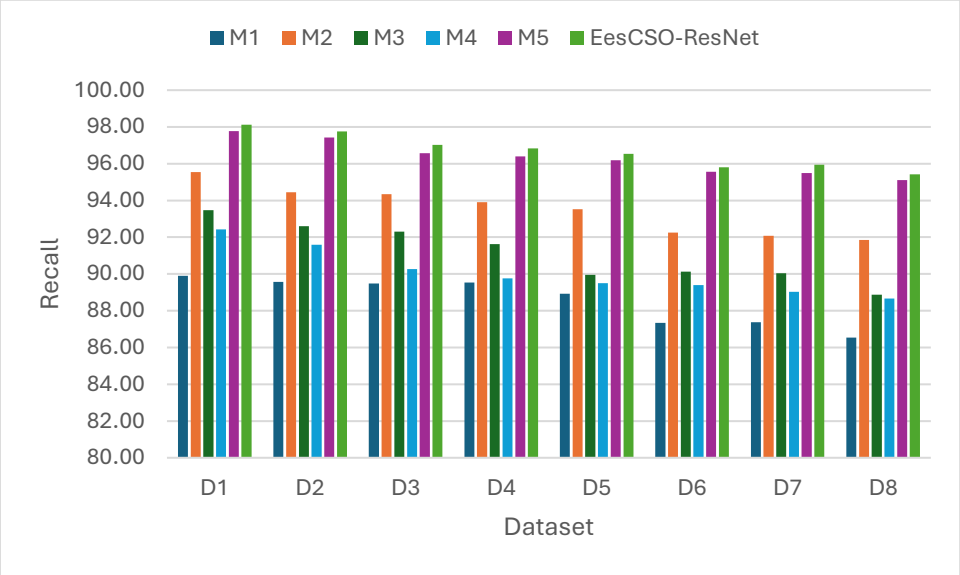

Figure 3: Graphical representation of results in Table 1

On Comparing results of D5, EesCSO-ResNet-39 outperforms M1 with 7.89%, M2 with 3.12%, M3 with 6.82%, M4 with 7.28% and M5 with 0.35%. On Comparing results of D6, EesCSO-ResNet-39 outperforms M1 with 8.84%, M2 with 3.72%, M3 with 5.93%, M4 with 6.69% and M5 with 0.27%. On Comparing results of D7, EesCSO-ResNet-39 outperforms M1 with 8.93%, M2 with 4.03%, M3 with 6.15%, M4 with 7.21% and M5 with 0.48%. On Comparing results of D8, EesCSO-ResNet-39 outperforms M1 with 9.31%, M2 with 3.75%, M3 with 6.87%, M4 with 7.09% and M5 with 0.34%.

Table 3: Experimental results of O-CSO-CNN w.r.t. F1-Score

	M1	M2	M3	M4	M5	EesCSO-ResNet
D1	91.90	95.85	94.94	94.09	98.28	98.70
D2	91.27	94.82	94.51	93.30	98.03	98.41
D3	91.06	94.50	94.30	92.70	97.11	97.44
D4	90.80	93.91	93.62	92.01	96.92	97.29
D5	90.42	93.62	92.39	91.22	96.58	97.00
D6	89.03	92.54	92.25	90.23	96.23	96.47
D7	88.64	91.82	92.23	89.83	95.80	96.14
D8	87.77	91.54	90.33	89.59	95.55	95.85

Table 3 indicates the experimental results of F1-Score on eight splits of the dataset. On comparing the results of EesCSO-ResNet-39 with other methods, the proposed model outperforms in most of the cases. On Comparing results of D1, EesCSO-ResNet-39 outperforms M1 with 6.89%, M2 with 2.89%, M3 with 3.81%, M4 with 4.67% and M5 with 0.42%. On Comparing results of D2 EesCSO-ResNet-39 outperforms M1 with 7.25%, M2 with 3.65%, M3 with 3.97%, M4 with 5.19% and M5 with 0.39%. On Comparing results of D3, EesCSO-ResNet-39 outperforms M1 with 6.55%, M2 with 3.02%, M3 with 3.23%, M4 with 4.87% and M5 with 0.35%. On Comparing results of D4, EesCSO-ResNet-39 outperforms M1 with 6.68%, M2 with 3.48%, M3 with 3.77%, M4 with 5.44% and M5 with 0.38%.

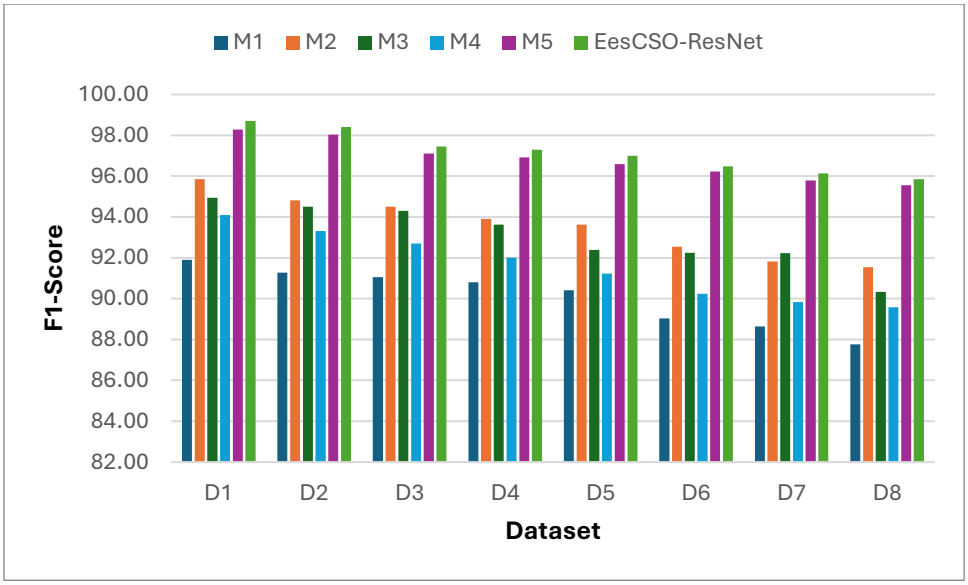

Figure 4: Graphical representation of results in Table 3

Comparing results of D5, EesCSO-ResNet-39 outperforms M1 with 6.79%, M2 with 3.49%, M3 with 4.75%, M4 with 5.96% and M5 with 0.43%. On Comparing results of D6, EesCSO-ResNet-39 outperforms M1 with 7.72%, M2 with 4.07%, M3 with 4.38%, M4 with 6.47% and M5 with 0.25%. On Comparing results of D7, EesCSO-ResNet-39 outperforms M1 with 7.8%, M2 with 4.49%, M3 with 4.06%, M4 with 6.56% and M5 with 0.36%. On Comparing results of D8, EesCSO-ResNet-39 outperforms M1 with 8.44%, M2 with 4.5%, M3 with 5.77%, M4 with 6.54% and M5 with 0.31%.

Table 4: Experimental results of O-CSO-CNN w.r.t. Accuracy

	M1	M2	M3	M4	M5	EesCSO-ResNet
D1	84.94	91.34	88.67	88.40	97.99	98.28
D2	84.48	91.45	88.04	88.02	97.44	97.78
D3	84.39	91.18	87.82	86.22	96.34	96.77
D4	83.04	90.87	87.22	84.94	95.85	96.14
D5	81.72	90.60	87.39	84.43	95.25	95.46
D6	81.86	90.68	86.50	83.83	94.84	95.09
D7	80.45	88.50	86.46	83.48	94.53	94.97
D8	80.41	88.43	86.03	83.31	93.65	94.12

Table 4 indicates the experimental results of Accuracy on eight splits of the dataset. On comparing the results of EesCSO-ResNet-39 with other methods, the proposed model outperforms in most of the cases. On Comparing results of D1, EesCSO-ResNet-39 outperforms M1 with 13.57%, M2 with 7.06%, M3 with 9.77%, M4 with 10.05% and M5 with 0.29%. On Comparing results of D2 EesCSO-ResNet-39 outperforms M1 with 13.61%, M2 with 6.48%, M3 with 9.96%, M4 with 9.98% and M5 with 0.35%. On Comparing results of D3, EesCSO-ResNet-39 outperforms M1 with 12.8%, M2 with 5.78%, M3 with 9.26%, M4 with 10.9% and M5 with 0.45%. On Comparing results of D4, EesCSO-ResNet-39 outperforms M1 with 13.63%, M2 with 5.49%, M3 with 9.28%, M4 with 11.65% and M5 with 0.3%.

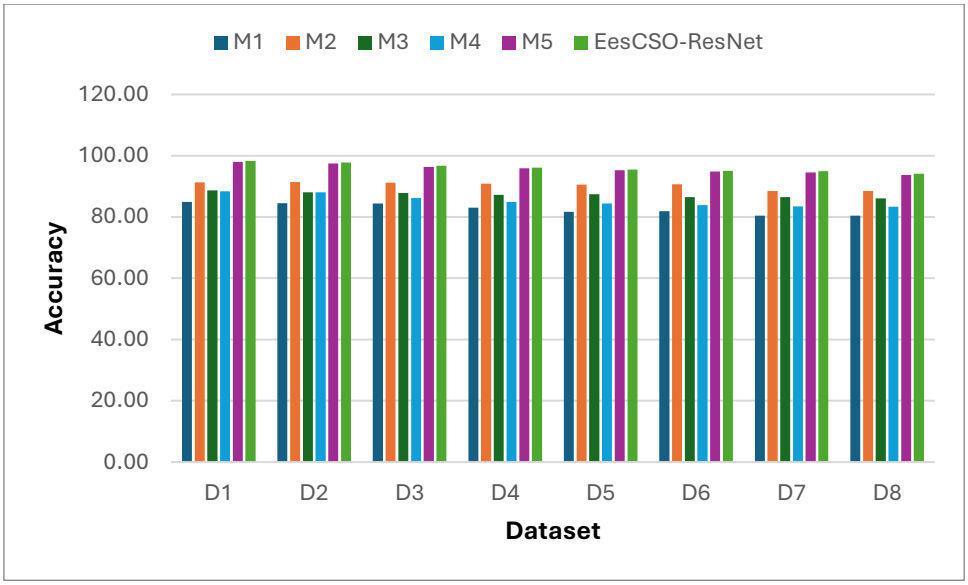

Figure 5: Graphical representation of results in Table 4

On Comparing results of D5, EesCSO-ResNet-39 outperforms M1 with 14.39%, M2 with 5.1%, M3 with 8.45%, M4 with 11.56% and M5 with 0.22%. On Comparing results of D6, EesCSO-ResNet-39 outperforms M1 with 13.91%, M2 with 4.64%, M3 with 9.03%, M4 with 11.84% and M5 with 0.26%. On Comparing results of D7, EesCSO-ResNet-39 outperforms M1 with 15.29%, M2 with 6.81%, M3 with 8.96%, M4 with 12.1% and M5 with 0.47%. On Comparing results of D8, EesCSO-ResNet-39 outperforms M1 with 14.57%, M2 with 6.05%, M3 with 8.6%, M4 with 11.49% and M5 with 0.5%.

CONCLUSION

To accurately predict crop disease, this study introduces a new framework that combines a modified ResNet-39 architecture with an improved swarm-based color edge segmentation technique. Even in low-contrast or noisy environments, the suggested segmentation process accurately identifies diseased areas by combining crow search optimization with opposing learning to collect color and edge characteristics. Optimizing computational efficiency and predictive accuracy, the redesigned ResNet-39 architecture significantly improves performance, making it ideal for situations with limited resources. Results from experiments show that the suggested framework achieves better accuracy, precision, and recall across several crop disease datasets than both conventional approaches and cutting-edge deep learning models. A streamlined pipeline with little computing overhead and strong detection capabilities is achieved by integrating the segmentation and prediction components. Integrating the framework with smart farming solutions based on the internet of things (IoT) and extending it to handle more crop kinds and diseases are potential areas of future effort.

REFERENCES

[1] Xu, Y., Nagahara, H., Shimada, A., Taniguchi, R.-i.: Transcut: transparent object segmentation from a light-field image. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3442–3450 (2015)

[2] Chen, G., Han, K., Wong, K.-Y.K.: Tom-net: learning transparent object matting from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9233–9241 (2018)

[3] Okazawa, A., Takahata, T., Harada, T.: Simultaneous transparent and non-transparent object segmentation with multispectral scenes. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4977–4984 (2019)

[4] Lee, S., Han, D.: Internal–external boundary attentions for transparent object segmentation. In: SIGGRAPH Asia 2022 Posters, pp. 1–2 (2022)

[5] Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10347–10357 (2021)

[6] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10012–10022 (2021)
[7] Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: a general vision transformer backbone with cross-shaped windows. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12124–12134 (2022)
[8] Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C.: Mvitv2: improved multiscale vision transformers for classification and detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4804–4814 (2022)
[9] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 568–578 (2021)
[10] Wang, H., Yang, M., Li, Z., Liu, Z., Hu, J., Fu, Z., Liu, F.: Scanet: improving multimodal representation and fusion with sparse-and cross-attention for multimodal sentiment analysis. Comput. Anim. Virtual Worlds 33(3–4), 2090 (2022)
[11] Wang, D., Xie, W., Cai, Y., Li, X., Liu, X.: Transformer-based rapid human pose estimation network. Comput. Graph. 116, 317–326 (2023)
[12] Lin, Z., Sun, W., Tang, B., Li, J., Yao, X., Li, Y.: Semantic segmentation network with multi-path structure, attention reweighting and multi-scale encoding. Vis. Comput. 39(2), 597–608 (2023)
[13] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
[14] Thirugnansambandam, Kalaipriyan, Debnath Bhattacharyya, Jaroslav Frnda, Dinesh Kumar Anguraj, and Jan Nedoma. "Augmented node placement model in t-WSN through multi objective approach." Comput. Mater. Contin. CMC Tech Sci. Press 69 (2021): 3629-3644.
[15] Saravanan, D., S. Janakiraman, K. Chandraprabha, T. Kalaipriyan, R. S. Raghav, and S. Venkatesan. "Augmented powell-based krill herd optimization for roadside unit deployment in vehicular ad hoc networks." Journal of Testing and Evaluation 47, no. 6 (2019): 4108-4127
[16] Thirugnanasambandam, Kalaipriyan, R. S. Raghav, Dinesh Kumar Anguraj, Devaraj Saravanan, and S. Janakiraman. "Multi-objective binary reinforced cuckoo search algorithm for solving connected coverage target based WSN with critical targets." Wireless Personal Communications 121, no. 3 (2021): 2301-2325.
[17] Rajeswari, M., Kalaipriyan Thirugnanasambandam, R. S. Raghav, U. Prabu, Devaraj Saravanan, and Dinesh Kumar Anguraj. "Flower pollination algorithm with Powell's method for the minimum energy broadcast problem in wireless sensor network." Wireless Personal Communications 119 (2021): 1111-1135.
[18] Thirugnanasambandam, Kalaipriyan, and Raghav RS. "Experimental analysis of ant system on travelling salesman problem dataset TSPLIB." EAI Endorsed Transactions on Pervasive Health and Technology 5, no. 19 (2019).
[19] Thirugnanasambandam, Kalaipriyan, S. V. Sudha, D. Saravanan, Renjith V. Ravi, Dinesh Kumar Anguraj, and R. S. Raghav. "Reinforced Cuckoo Search based fugitive landfill methane emission estimation." Environmental Technology & Innovation 21 (2021): 101207.
[20] C.H. Bock, et al. Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging. In: Critical reviews in plant sciences 29.2, pp. 59–107 (2010)Return to ref 1 in article
[21] Andrew M. Mutka, Rebecca S. Bart, Image-based phenotyping of plant disease symptoms. In: Frontiers in plant science 5, p. 734 (2015)
[22] Melike Sardogan, Adem Tuncer, Yunus Ozen, Plant leaf disease detection and classification based on CNN with LVQ algorithm. In: 2018 3rd international conference on computer science and engineering (UBMK). IEEE. 2018, 382–385 (2018)
[23] N. Valliammal, S.N. Geethalakshmi, Plant leaf segmentation using non linear K means clustering. In: International Journal of Computer Science Issues (IJCSI) 9.3, p. 212 (2012)
[24] Guruprasad Samanta, Amlan Chakrabarti, Bhargab B. Bhattacharya. Extraction of leaf-vein parameters and classification of plants using machine learning. In: Proceedings of International Conference on Frontiers in Computing and Systems. Springer. pp. 579–586 (2021)

[25] Jyostna Devi Bodapati, Naralasetti Veeranjaneyulu, Abnormal network traffic detection using support vector data description. In: Proceedings of the 5th international conference on frontiers in intelligent computing: Theory and applications. Springer. pp. 497–506 (2017)