IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 18, 2020, accepted October 12, 2020, date of publication October 19, 2020, date of current version October 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3031914

Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection

TAN NHAT PHAM, LY VAN TRAN, AND SON VU TRUONG DAO
School of Industrial Engineering and Management, International University, Vietnam National University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Son Vu Truong Dao (dvtson@hcmiu.edu.vn)

ABSTRACT Plant disease, especially crop plants, is a major threat to global food security since many diseases directly affect the quality of the fruits, grains, and so on, leading to a decrease in agricultural productivity. Farmers have to observe and determine whether a leaf was infected by naked eyes. This process is unreliable, inconsistent, and error prone. Several works on deep learning techniques for detecting leaf diseases had been proposed. Most of them built their models based on limited resolution images using convolutional neural networks (CNNs). In this research, we aim at detecting early disease on plant leaves with small disease blobs, which can only be detected with higher resolution images, by an artificial neural network (ANN) approach. After a pre-processing step using a contrast enhancement method, all the infested blobs are segmented for the whole dataset. A list of several measurement-based features that represents the blobs are chosen and then selected based on their influences on the model's performance using a wrapper-based feature selection algorithm, which is built based on a hybrid metaheuristic. The chosen features are used as inputs for an ANN. We compare the results obtained using our methods with another approach using popular CNN models (AlexNet, VGG16, ResNet-50) enhanced with transfer learning. The ANN's results are better than those of CNNs using a simpler network structure (89.41% vs 78.64%, 79.92%, and 84.88%, respectively). This shows that our approach can be implemented on low-end devices such as smartphones, which will be of great assistance to farmers on the field.

INDEX TERMS Neural network, image classification, plant disease, feature selection, precision agriculture.

I. INTRODUCTION

Plant diseases are the main cause of quantity and quality losses in agricultural production. These losses negatively impact the production cost as well as the profit of the stakeholders in agriculture. However, tools for quick and accurate recognition remains scarce. The welfare and livelihoods of farmers as well as the food supply and the nutrition security of a nation are severely threatened should any kinds of disease outbreaks happen.

Traditionally, farmers and plant pathologists use their eyes to detect diseases and make decisions based on their experiences, which is often not accurate and sometimes biased since in the early stage many types of diseases appear to be the same. Also, their experiences need to be passed down generations by generations. This approach leads to the unnecessary use of pesticides, which in turn results in higher production cost. Based on these pieces of evidence, the need for an accurate disease detector associated with a reliable database to help farmers is necessary, especially for the case of young and inexperienced ones. Advances in computer vision pave the way for this with the state-of-the-art Deep learning (DL) or machine learning (ML) algorithms. There is also a need for an early disease detection system to protect the crop in time.

There are many previous research conducted for this purpose. Most of them make use of the so-called "Plant Village" [1] dataset, a widely known dataset that is available online, with CNNs being by far the most popular models. However, the CNNs require a large amount of data for their training [2]. In this work, we proposed two approaches: CNN models enhanced with transfer learning (TL) and Artificial Neural Network (ANN) with Feature Selection (FS) to solve the multi-class classification for three types of diseases,

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

namely Anthracnose, Gall Midge, and Powdery Mildew. The proposed frameworks aim at increasing the models' accuracy when the data is limited. The rest of this work is structured as follows: section II discusses related works using CNN and

ANN, section III presents our approaches, followed by results and discussion in section IV, and culminating in conclusion in section V.

II. LITERATURE REVIEW

A. PLANT DISEASES RECOGNITION USING CNN

CNNs are a class of hierarchical model where an object's features are learned by training through many examples. They consist of multiple layers with later ones built on top of previously learned features [3]. Saleem *et al.* [4] conducted a review of plant disease detection and classification by Deep Learning techniques. He concluded that the ''PlantVillage'' dataset has a simple or plain background and more practical scenarios should be considered. Also, hyperspectral or multispectral imaging should be used together with DL models to develop early disease detectors, and a much wider variety of training data should be collected, from several sources of different geographic areas, cultivation conditions, and image capturing modes. Konstantinos *et al.* [5] implemented a Visual Geometry Group (VGG) model for plant disease detection, in which the network achieved 99.53% of accuracy over the data given in [1]. Rangarajan *et al.* [6] used AlexNet and VGG16 to classify tomato leaf diseases, in which VGG16 reached 97.29% and AlexNet reached 97.49% of accuracy. Mohanty *et al.* [7] implemented transfer learning (TL) approach with a pre-trained AlexNet to classify diseases in crops. The model can classify 26 distinct diseases in 14 crop species with a sample size of 54,306 images and 99.35% accuracy. Too *et al.* [8] conducted a review of the performances of different deep learning models namely Visual Geometry Group (VGG), Inception V4, ResNet, and DenseNet in disease classification using the ''Plant Village'' dataset, in which DenseNet was the most efficient with 99.75% of accuracy. VGG16 was also been used in Shijie *et al.* [9] to classify tomato diseases, which achieved 88% accuracy. Many other applications of CNN are also discussed in [10].

B. PLANT DISEASES RECOGNITION USING ARTIFICIAL NEURAL NETWORK (ANN)

Khirade *et al.* [11] conducted a review of various techniques to segment the infested part of the plant. This research also reviewed some feature extraction and classification techniques to extract the features from infected leaves and the classification of plant diseases. Many approaches are using ANN methods for the classification of disease in plants such as self-organizing feature map, backpropagation algorithm, support vector machines (SVMs), etc. Singh *et al.* [12] used ANN together with image segmentation to detect diseases on various types of plants, namely banana, beans, jackfruit, lemon, mango, potato, tomato, and sabota. Firstly, the Minimum Distance Criterion with K-Mean Clustering is used, then the classification is done by SVM. The proposed method reached an average accuracy of 97.6%. Kulkarni *et al.* [13] proposed a method using ANN together with Gabor filter for feature extraction for early plant diseases detection, which gives a recognition rate up to 91%. The ANN used the combination of texture and color features for classification.

In summary, previous research results are significant. However, there are still gaps in them. Firstly, ''Plant Village'' is the most widely used dataset in previous research, this dataset has 54.303 healthy and infected leaf images belonging to 38 types of plant and diseases, but most of them are in the later stage of disease spreading in which the infected area is relatively large compared to the area of the leaf, so it is easy to spot the type of disease. Secondly, when testing with images taken under conditions that are different from the training images, the CNN models' performance decreased significantly, as stated by Mohanty *et al.* [7]. He also suggested that more training images in more practical scenarios should be considered. Another worth noting point is that ''Plant Village'' has no sample of mango leaves, which is our main concern in this research. A new dataset [14] was used in [10] to detect mango leaves' diseases, however, they only focus on healthy leaves and those infected with ''Anthracnose'' only. For the research involving ANN, the authors implemented many processing techniques to extract desired features from the images then put them to the ANN. However, there was a lack of a feature selection step to select the most useful features.

C. FEATURE SELECTION (FS)

FS aims at selecting a number of features and ignoring the irrelevant, noisy features from a set of features for easier subsequent analysis. The selection criteria are the redundancy and relevance of the features. Based on these two characteristics, Yu *et al.* [15] have classified the feature subset into four types: noisy & irrelevant, redundant & weakly relevant, weakly relevant and non-redundant, and strongly relevant. An irrelevant feature is one that has no significant influence on the prediction accuracy, thus should be discarded. Models, search strategies, feature quality measures, and feature evaluation are common techniques that can be implemented to wrapper or filter methods. The number of features is directly proportional to the size of the hypothesis space, i.e., as the number of features increases, the size of the search space is also increased. One such case is that if there are M features with the binary class label in a dataset, then it has 2^{2^M} combination in the search space.

There are three types of FS methods, namely Filter, Wrapper, and Embedded Methods. The Filter method selects statistics-based features. It can be conducted independently out of the learning algorithm and therefore is less computationally expensive. Information gain, chi-square test [16], Fisher score, correlation coefficient, and variance threshold are common measures used to understand the importance

of the features. In contrast, the Wrapper method's performance highly depends on the classifier. Best features are chosen according to the performance of the classifier. Wrapper methods are much more computationally expensive than filter methods since it needs to run simultaneously with the classifier many times, but they are more accurate than the filter ones. Some examples are Recursive feature elimination [17], Sequential feature selection algorithms [18], and Genetic algorithms [19]. The third approach is called Embedded method, which selects features based on a combination of ensemble learning and hybrid learning methods. This method has a collective decision, therefore its performance is better than the previous ones. One example is Random forest which is less computationally intensive than wrapper methods. One drawback of the embedded method is that it is specific to a learning model. Many evolutionary metaheuristics-based feature selection methods are also proposed, many of them are wrapper-type since it has been proven that wrapper provides better performance [20]. Too *et al.* [21] proposed a competitive binary Grey Wolf Optimizer (CBGWO), which is based on the Grey Wolf Optimizer proposed by Mirjalili *et al.* [22], for feature selection problem in EMG signal classification. The results showed that CBGWO outranked other algorithms in terms of performance for that case study. Many other wrapper-based feature selection algorithms were also introduced in many previous works to select a subset of features, including binary grey wolf optimization (BGWO) [23], binary particle swarm optimization (BPSO) [24], ant colony optimization (ACO) [25], and binary differential evolution (BDE) [26].

FIGURE 1. Four classes of leaf diseases in this study: Anthracnose (1), Gall Midge (2), Healthy (3), and Powdery Mildew (4).

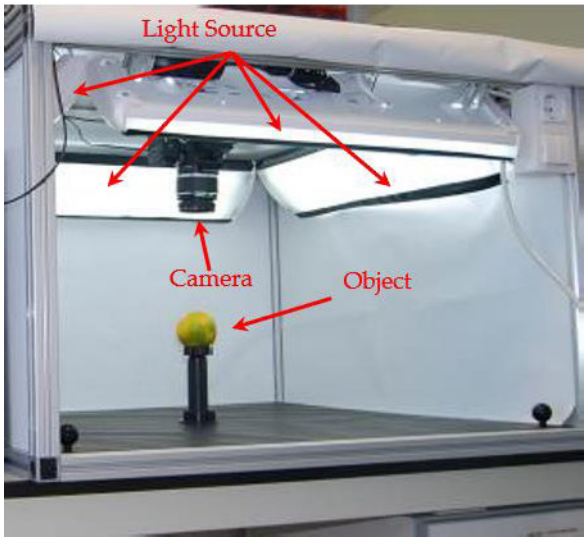

FIGURE 2. Chamber for taking images.

III. PROPOSED APPROACH

A. DATASET

Our data set contains 450 images of mango leaves, which belong to four different types (three diseases and one healthy): Anthracnose, Gall Midge, Powdery Mildew, and Healthy. These are also four classes in our classification as in Fig. 1. The samples are collected from various places in An Giang province, which is known as one of the places with the largest productions of mango in Vietnam. The leaves were collected when blobs started to appear, they were taken from the tree and their images were taken in the same day. The images are captured using a camera in the resolution of 3096×3096 pixels with no background.

The images are taken under different lighting conditions in a chamber shown in Fig. 2. The chamber consists of an aluminum frame and one camera model CANON 60D is mounted on top so that it points straight to the frame at the bottom. Besides, light sources are placed around the camera to eliminate the shadow of an object. The lighting intensity can be adjusted to simulate real lighting conditions.

The flowchart of this work is shown in Fig. 3. To begin with, Mango images are pre-processed by rescaling into a lower resolution, compared with the original size. Then, the centre alignment step is responsible for guaranteeing the region of a leaf to be in the centre of the image fitting exactly the top and bottom of the image. Since there are various contrasts in leaf images, we apply the contrast enhancement method to adjust pixel intensities which benefit in case of providing more information in some areas of an image. The image values are then normalized within the [0,1] range so that the loss function would reach the global optimal easily. Normalization also speeds up the convergence of the backpropagation algorithm. The image dataset is divided into Training, Validation and Testing set: First, 20% of the total dataset is set as the testing set, then the remaining part of the dataset is split again in the 80/20 fashion, with the 20% being the validation set, and the 80% being the training set. The training and validation set is used for training the model, while the testing set is used to evaluate the performance of the trained model. The data is split randomly using Python's ''train_test_split'' built-in function of the ''scikit-learn'' library [27]. Python's ''shuffle'' method was used together with ''train_test_split'' to produce random subsets. This method is based on a random number generator and it reorders the images so that the sampling process is unbiased. We also split the data in the ''PlantVillage'' dataset in the same manner to create pre-trained CNN models.

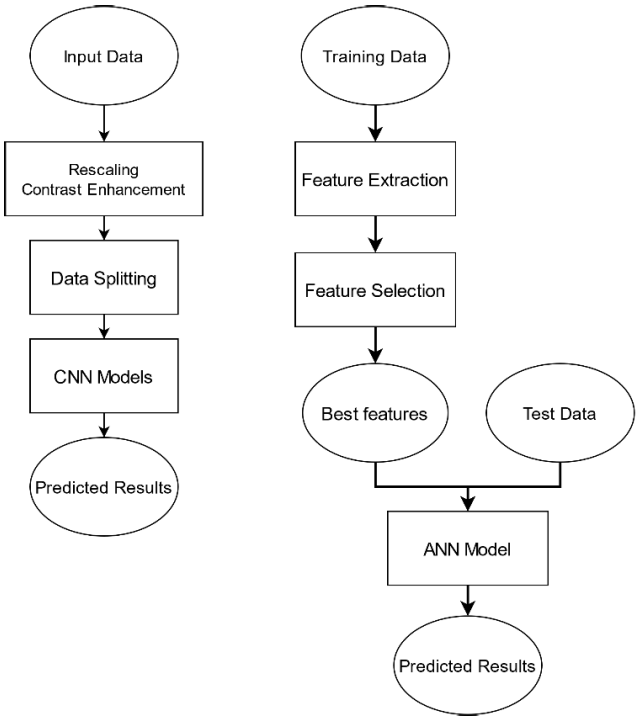

FIGURE 3. Flowchart of the two approaches in this study.

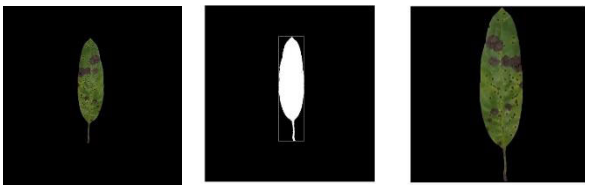

FIGURE 4. An example of rescaling and center alignment for a leaf image: Original image; Corresponding image with bounding box; Final rescaled and center-aligned image.

B. IMAGE PRE-PROCESSING

Since the leaves have different sizes, it is necessary to perform rescaling to ensure the training and testing image have the same dimension. Rescaling is performed to compress the original images to lower resolution ones, 256 × 256 pixels to be exact. First, the original image is segmented and converted to binary one to find the minimum bounding box. The vertical size of the bounding box was used to rescale to 256 pixels to ensure the top and bottom leaf fit exactly to the top and bottom of the scaled image. The horizontal size of the bounding box will be used to shift the leaf image into the exact center of the scaled image. The results can be found in Fig. 4.

Due to various contrasts in the leaf region, the contrast enhancement method is used to change pixel intensities which benefit in case of providing more information in some areas of an image.

Many contrast enhancement methods have been widely applied to improve the quality of the image [28]. In this paper, to ameliorate features which are low contrast to achieve improvement in term of contrast quality, we use a contrast enhancement approach based on the one in [29] before further

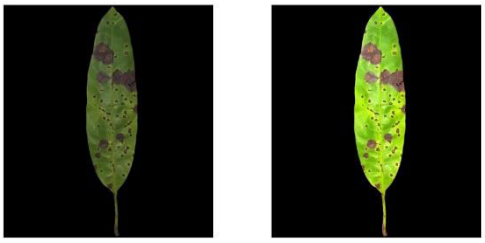

FIGURE 5. Contrast enhancement effect: before and after.

TABLE 1. Comparison of CNN models considered.

Model	AlexNet	VGG16	ResNet-50
Number of layers	8	16	50
Number of parameters (in million)	60	138	23

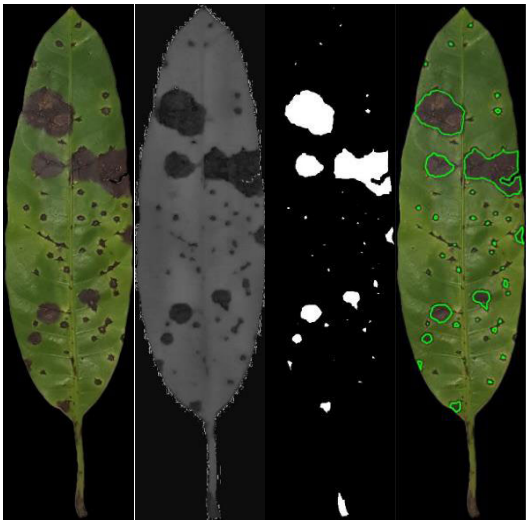

FIGURE 6. Feature extraction: Original Image, CLAHE is applied on H channel, Extracted defective regions, Final result.

analysis. The main idea of this method is to preserve the mean brightness of an input image during contrast adjustment in local regions. Firstly, the input image in RGB color channels is converted into HSI ones. This approach only focuses on the intensity parameter and preserves other hue and saturation values. Afterward, the intensity is divided by separator into two sub-parameters which are high and low groups. This is done by the search method shown in the following equation.

$$\gamma_{hi} = \{\gamma\,(i)\,|\,i > \gamma_m\}, \quad \gamma_{lo} = \{\gamma\,(j)\,|\,j \leq \gamma_m\} \tag{1}$$

where γ_{hi} and γ_{lo} are intensity high and low groups respectively, γ_m is a trial threshold intensity value which is defined to divide the image into two sub-images. After obtaining estimates of the two sub-parameters of intensity, a combination of them is performed to achieve the enhanced intensity. The enhanced intensity is calculated by the following equation as follows:

$$\gamma_{enhance}\,(i) = \gamma_{lo} + (\gamma_{hi} - \gamma_{lo}) \times \chi\,(i) \tag{2}$$

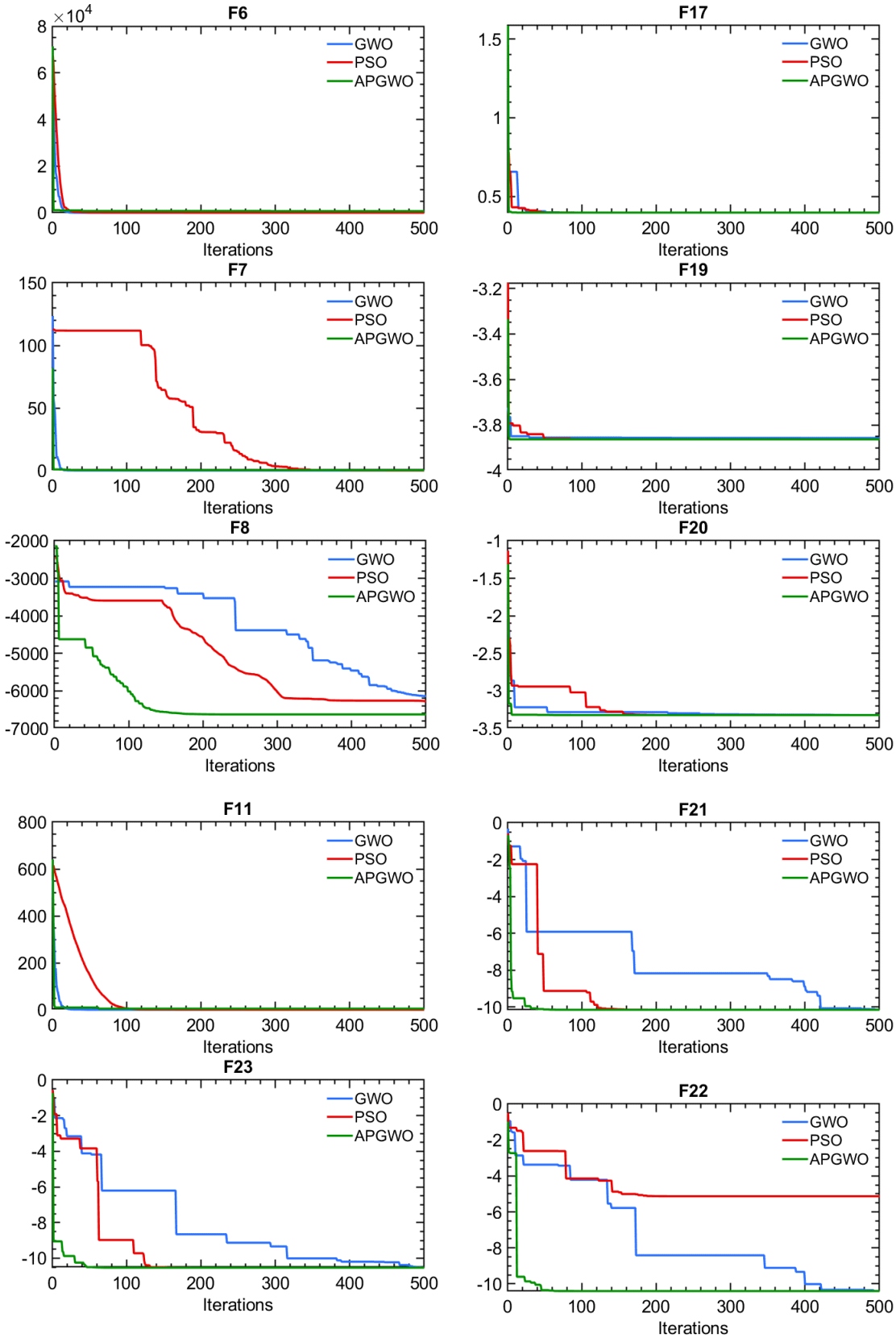

FIGURE 7. Average benchmarking results of the proposed APGWO with 5 runs. Vertical axis is the function's fitness.

TABLE 2. Details of benchmark functions.

Function	Dimension	Range	f_{min}
Unimodal			
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100,100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10,10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j-1}^{i} x_j)^2$	30	[-100,100]	0
$F_4(x) = max_i\{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100,100]	0
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_i - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	[-30,30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100,100]	0
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1]$	30	[-1.28,1.28]	0
Multimodal			
$F_8(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	[-500,500]	-418.9289×5
$F_9(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[-32,32]	0
$F_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[-600,600]	0
$F_{12}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10sin^2(3\pi y_{i+1})] + (y_n - 1)^2 + \sum_{i=1}^{n} u(x_i, 10,100,4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30	[-50,50]	0
$F_{13}(x) = 0.1\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5,100,4)$	30	[-50,50]	0
Fixed-dimension Multimodal			
$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[-65,65]	1
$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[-5,5]	0.00030
$F_{16}(x) = 4x_i^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[-5,5]	-1.0316
$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	[-5,5]	0.398
$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2\left(19 - 14x_1 + 3x_i^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[-2,2]	3
$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{4} a_{ij}\left(x_j - p_{ij}\right)^2)$	3	[-1,3]	-3.86
$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	6	[0,1]	-3.32
$F_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.1352
$F_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.4028
$F_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0,10]	-10.5363

where $\chi(i)$ is the cumulative density calculated from the histogram. Mean brightness and input brightness are calculated and compared to minimize the error. The iteration of this process is performed until getting an optimal value of enhanced intensity. Eventually, enhanced intensity and other initial hue and saturation values are combined and converted back to RGB color channel to give the output image. The contrast enhancement effect is illustrated in Fig. 5.

C. APPROACH WITH CNNS

After performing contrast enhancement step, we investigate four popular CNN models, namely AlexNet [30], VGG16 [31], ResNet [32] (ResNet-50 variant). A comparison between these models is available in Table 1 and their architecture are shown in Fig. 15-Fig. 18 in the Appendix.

In this work, we kept the original architecture of these CNN and only modified the last fully-connected layer of them to four nodes, according to the four classes in our problem. We also perform transfer learning to fine-tune the models to enhance their performances. Transfer learning (TL) is a solution for the lack of training data in deep learning [33]. TL means using the knowledge from a specific task to solve another correlated task. In deep learning, TL helps the model learn the features from a large dataset so that it performs

TABLE 3. Result of FS.

Index	Feature	Notation	Status
1	Area		Selected
2	Major axis length		
3	Minor axis length		
4	Eccentricity		Selected
5	Orientation		Selected
6	ConvexArea		Selected
7	EquivalentDiameter		Selected
8	Solidity		
9	Ratio		
10	Perimeter		
11	R_mean		Selected
12	G_mean		Selected
13	B_mean		
14	H_mean		Selected
15	S_mean		Selected
16	V_mean		Selected
17	R_StdDeV		Selected
18	G_StdDev		Selected
19	B_StdDev		
20	H_StdDev		Selected
21	S_StdDev		Selected
22	V_StdDev		Selected
23	R_Skewness		Selected
24	G_Skewness		Selected
25	B_Skewness		
26	H_Skewness		
27	S_Skewness		
28	V_Skewness		
29	R_Kurtosis		Selected
30	G_Kurtosis		
31	B_Kurtosis		Selected
32	H_Kurtosis		Selected
33	S_Kurtosis		
34	V_Kurtosis		Selected
35	Contrast_0		Selected
36	Contrast_45		Selected
37	Contrast_90		
38	Contrast_135		Selected
39	Corr_0		
40	Corr_45		
41	Corr_90		
42	Corr_135		Selected
43	SumSq_0		Selected
44	SumSq_45		Selected
45	SumSq_90		Selected
46	SumSq_135		Selected
47	Homo_0		Selected
48	Homo_45		Selected
49	Homo_90		Selected
50	Homo_135		Selected
51	$H_{1/16}S_{1/2}V_{1/2}$	bin1	Selected
52	$H_{1/16}S_{1/2}V_{2/2}$	bin2	Selected
53	$H_{1/16}S_{2/2}V_{1/2}$	bin3	Selected
54	$H_{1/16}S_{2/2}V_{2/2}$	bin4	
55	$H_{2/16}S_{1/2}V_{1/2}$	bin5	Selected
56	$H_{2/16}S_{1/2}V_{2/2}$	bin6	Selected
57	$H_{2/16}S_{2/2}V_{1/2}$	bin7	Selected
58	$H_{2/16}S_{2/2}V_{2/2}$	bin8	Selected
59	$H_{3/16}S_{1/2}V_{1/2}$	bin9	
60	$H_{3/16}S_{1/2}V_{2/2}$	bin10	Selected
61	$H_{3/16}S_{2/2}V_{1/2}$	bin11	Selected
62	$H_{3/16}S_{2/2}V_{2/2}$	bin12	
63	$H_{4/16}S_{1/2}V_{1/2}$	bin13	Selected
64	$H_{4/16}S_{1/2}V_{2/2}$	bin14	Selected
65	$H_{4/16}S_{2/2}V_{1/2}$	bin15	
66	$H_{4/16}S_{2/2}V_{2/2}$	bin16	Selected
67	$H_{5/16}S_{1/2}V_{1/2}$	bin17	
68	$H_{5/16}S_{1/2}V_{2/2}$	bin18	Selected
69	$H_{5/16}S_{2/2}V_{1/2}$	bin19	Selected
70	$H_{5/16}S_{2/2}V_{2/2}$	bin20	
71	$H_{6/16}S_{1/2}V_{1/2}$	bin21	Selected
72	$H_{6/16}S_{1/2}V_{2/2}$	bin22	Selected
73	$H_{6/16}S_{2/2}V_{1/2}$	bin23	Selected
74	$H_{6/16}S_{2/2}V_{2/2}$	bin24	Selected
75	$H_{7/16}S_{1/2}V_{1/2}$	bin25	Selected
76	$H_{7/16}S_{1/2}V_{2/2}$	bin26	Selected
77	$H_{7/16}S_{2/2}V_{1/2}$	bin27	Selected
78	$H_{7/16}S_{2/2}V_{2/2}$	bin28	Selected
79	$H_{8/16}S_{1/2}V_{1/2}$	bin29	Selected
80	$H_{8/16}S_{1/2}V_{2/2}$	bin30	Selected
81	$H_{8/16}S_{2/2}V_{1/2}$	bin31	
82	$H_{8/16}S_{2/2}V_{2/2}$	bin32	Selected
83	$H_{9/16}S_{1/2}V_{1/2}$	bin33	Selected
84	$H_{9/16}S_{1/2}V_{2/2}$	bin34	Selected
85	$H_{9/16}S_{2/2}V_{1/2}$	bin35	Selected
86	$H_{9/16}S_{2/2}V_{2/2}$	bin36	Selected
87	$H_{10/16}S_{1/2}V_{1/2}$	bin37	
88	$H_{10/16}S_{1/2}V_{2/2}$	bin38	Selected
89	$H_{10/16}S_{2/2}V_{1/2}$	bin39	Selected
90	$H_{10/16}S_{2/2}V_{2/2}$	bin40	
91	$H_{11/16}S_{1/2}V_{1/2}$	bin41	Selected
92	$H_{11/16}S_{1/2}V_{2/2}$	bin42	Selected
93	$H_{11/16}S_{2/2}V_{1/2}$	bin43	Selected
94	$H_{11/16}S_{2/2}V_{2/2}$	bin44	Selected
95	$H_{12/16}S_{1/2}V_{1/2}$	bin45	Selected
96	$H_{12/16}S_{1/2}V_{2/2}$	bin46	Selected
97	$H_{12/16}S_{2/2}V_{1/2}$	bin47	
98	$H_{12/16}S_{2/2}V_{2/2}$	bin48	Selected
99	$H_{13/16}S_{1/2}V_{1/2}$	bin49	
100	$H_{13/16}S_{1/2}V_{2/2}$	bin50	
101	$H_{13/16}S_{2/2}V_{1/2}$	bin51	
102	$H_{13/16}S_{2/2}V_{2/2}$	bin52	
103	$H_{14/16}S_{1/2}V_{1/2}$	bin53	Selected
104	$H_{14/16}S_{1/2}V_{2/2}$	bin54	Selected
105	$H_{14/16}S_{2/2}V_{1/2}$	bin55	Selected

TABLE 3. *(Continued.)* Result of FS.

TABLE 3. *(Continued.)* **Result of FS.**

106	$H_{14/16}S_{2/2}V_{2/2}$	bin56	Selected
107	$H_{15/16}S_{1/2}V_{1/2}$	bin57	Selected
108	$H_{15/16}S_{1/2}V_{2/2}$	bin58	Selected
109	$H_{15/16}S_{2/2}V_{1/2}$	bin59	
110	$H_{15/16}S_{2/2}V_{2/2}$	bin60	Selected
111	$H_{16/16}S_{1/2}V_{1/2}$	bin61	Selected
112	$H_{16/16}S_{1/2}V_{2/2}$	bin62	Selected
113	$H_{16/16}S_{2/2}V_{1/2}$	bin63	Selected
114	$H_{16/16}S_{2/2}V_{2/2}$	bin64	Selected

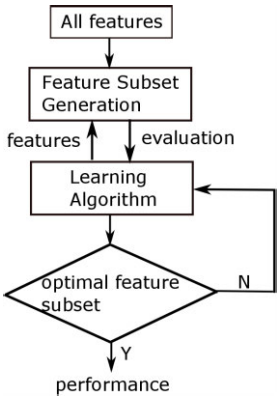

FIGURE 8. Process of APGWO- wrapper based feture selection.

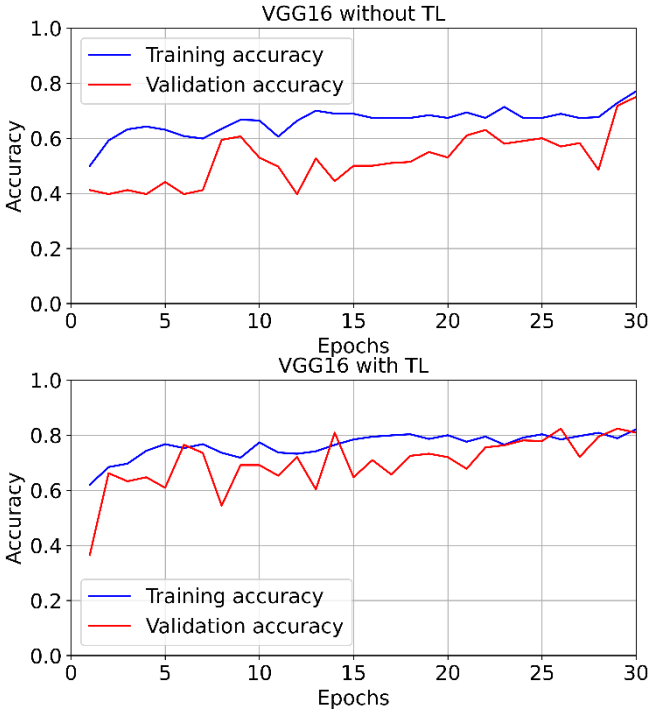

FIGURE 9. Performance of VGG16 without TL, and with TL.

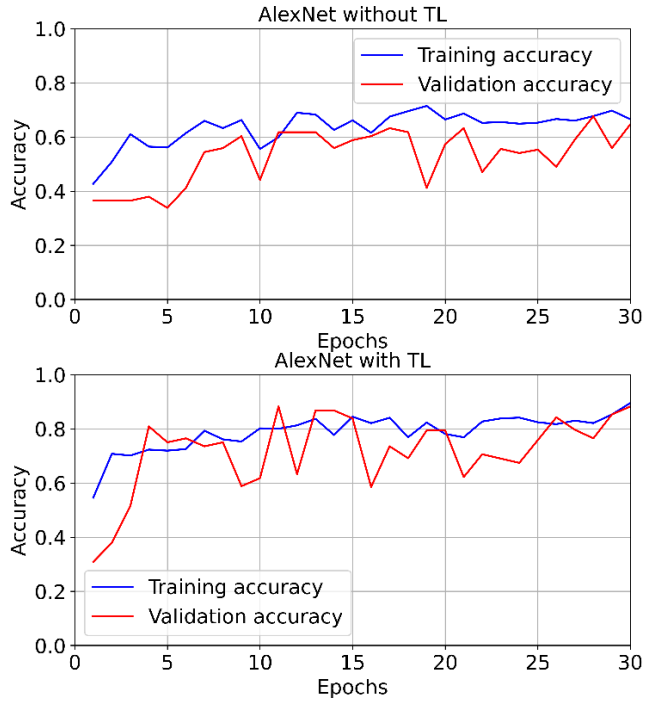

FIGURE 10. Performance of AlexNet without TL, and with TL.

better on a relevant dataset but may be smaller in size, and this method has shown effectiveness in image classification task [7], [34], [35]. In our work, the models are first trained on the Plant Village dataset. This dataset has a huge amount of data and allows the convolutional layers of the models to learn similar features effectively. Based on the Plant Village dataset, pre-trained models are created, then the models are trained one more time on our dataset to calibrate the models. The cross-entropy is applied as a loss function to estimate the error prediction after the classification layer. Moreover, the optimization algorithm for the training process is done by Adam optimizer[36]. The maximum number of epochs to train the proposed model is 30 with an initial learning rate of 0.0005.

D. FEATURE EXTRACTION FOR ANN APPROACH

Contrast Limited Adaptive Histogram Equalization (CLAHE) was proposed by K. Zuiderveld in 1994 [37]. The method examines a histogram of intensities in a contextual region centered at each pixel and sets the displayed intensity at the pixel as the rank of that pixel's intensity in its histogram. That histogram is a modified form of the ordinary histogram in which the contrast enhancement induced by the method at each intensity level is limited to a user-selectable maximum. In this study, CLAHE is utilized to perform the thresholding of the image. The original image is converted to HSV format, and CLAHE is applied to the H channel to enhance the contrast of the defective regions. The defective regions are then separated and mapped back to the original image. The process is illustrated in Fig. 6. In this approach, instead of recognizing which types of diseases that a leaf is infested with like in the approach of CNNs, we extract, learn the features, and classify the types of infested blobs. In other words, this approach is more accurate in case a leaf is infested with more than one type of disease.

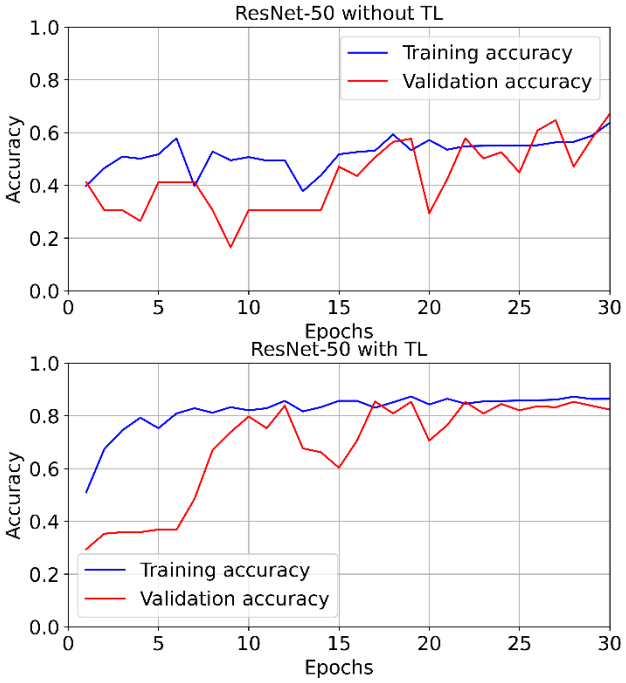

FIGURE 11. Performance of ResNet-50 without TL, and with TL.

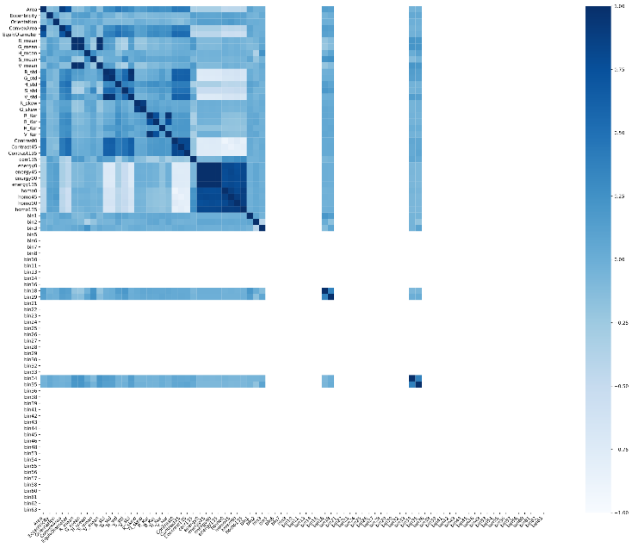

FIGURE 12. Correlation between chosen features.

In this study, we extract the following features:

- Statistics-based: mean, standard deviation, skewness, and kurtosis of color channels R, G, B, H, S, V (feature 11-34, Table 2).
- Geometry-based (feature 1-10, Table 3):
  - Area
  - Perimeter
  - Major/Minor axis length: Lengths (in pixels) of the major/minor axis of an ellipse having the same normalized second central moments as the separated blob.
  - Eccentricity: distance between the foci divided by the major axis length.

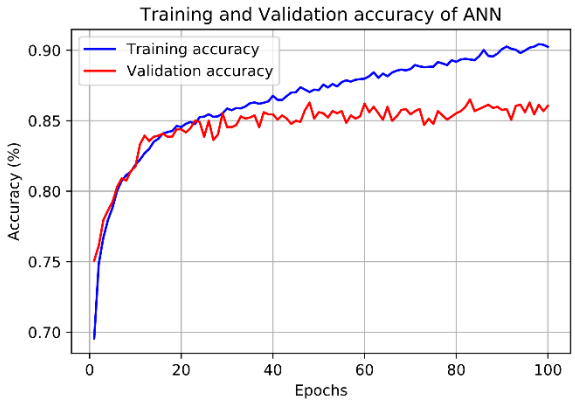

FIGURE 13. Performance of ANN approach.

  - Orientation: angle between the x-axis and the major axis of the above-mentioned ellipse, from -90^0 to 90^0.
  - Convex area: area generated by convex hull of the blob.
  - Equivalent diameter: diameter of a circle having the same area as the blob, calculated by $\sqrt{\frac{4\times Area}{\pi}}$.
  - Solidity: Ratio of the blob area over the convex area, calculated by $\frac{Area}{Convexarea}$
  - Ratio of pixels in the region with respect to the total bounding box.
- Textural features: GLCM features with four distinct angle offsets, 0^0, 45^0, 90^0, 135^0 [38]: Contrast, Homogeneity, Sum of squared elements, Correlation (feature 35-50, Table 3).
- Compressed HSV density: Originally, each H, S, and V layer has 256 bins which total 768 features. To limit the feature dimension, we reduce the number of bins in H layer to 16 and those in S and V layers to 2 since in our case, H channel provides more useful information. Instead of having 768 features, we only have $16\times2\times2 = 64$. Each defect slob is rescaled to a standard size of 20×20 pixels. The relative frequencies of the 64-feature-set are computed and to be used for our FS algorithm (feature 51-114, Table 3).

E. PROPOSED FS ALGORITHM

In this study, we implement a wrapper-based method with a hybrid meta-heuristic called Adaptive Particle – Grey Wolf Optimization (APGWO)[39], which is combined from Particle Swarm Optimization (PSO)[40] and Grey Wolf Optimization (GWO) [22]. In PSO, there are two controlling coefficients, namely c_1 and c_2, which is called "acceleration coefficients" and influence the exploration and exploitation capabilities of the algorithm. These two coefficients are usually set to 1 or 2 by empirical studies to get a balance between the exploration and exploitation capability. We modified these two coefficients by the following formulas:

$$c_1^t = 1.2 - \frac{f\left(x_k^t\right)}{f\left(gBest\right)} \tag{3}$$

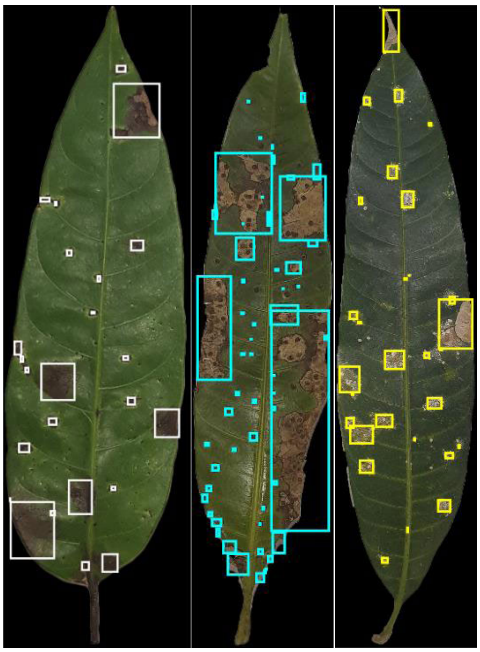

FIGURE 14. Segmentation results of leaves infected with Anthracnose, Gall Midge, and Powdery Mildew (from left to right).

$$c_2^t = 0.5 + \frac{f\left(x_k^t\right)}{f\left(gBest\right)} \tag{4}$$

where c_1^t and c_2^t stand for the coefficients at iteration t; $f\left(x_k^t\right)$ is the fitness of particle k at iteration t, and $f\left(gBest\right)$ is the swarm's global best fitness. The values of 1.2 and 0.5 are also found by empirical studies. This modification allows the search agents to focus on exploration in the early phase, then encourages them to converge to a good optimum near the end of the process. We also modify the formula for inertia as followed:

$$w_t = (maxIter - t) \times \frac{wMax - wMin}{maxIter} + wMin \tag{5}$$

This formula would reduce the inertia of search agents gradually during the iteration process, thus control the particles' velocity so they would not jump out of good solutions randomly. This strategy also enhances the local search capability of the algorithm. Finally, we update the velocity and position of particles by the following equations:

$$\begin{aligned} v_k^{t+1} &= w \times v_k^t + c_1^t \times rand \times \left(pbest_k^t - x_k^t\right) \\ &\quad + c_2^t \times rand \times \left(gBest - x_k^t\right) \end{aligned} \tag{6}$$

$$x_k^{t+1} = x_k^t + v_k^t \tag{7}$$

In this study, we introduce a probability of mutation, which will trigger a small number of iterations of GWO within the PSO main loop, and the best solution of the nested loop would be added to the swarm. The pseudocode for this algorithm is given in Algorithm 1.

The proposed algorithm is tested on 23 benchmark functions given in [41], which the author in [22] also used to evaluate the GWO. Details of the functions can be found in Table 2. These functions have many local optima. The problem becomes worse with higher dimension cases and is usually used to benchmark meta-heuristic algorithms. "Range" indicates the boundary of the search space, and f_{min} is the optimum. The proposed APGWO yields competitive results compared with the standard PSO and GWO algorithm. Fig. 7 shows some representative cases.

Algorithm 1 Proposed APGWO

```
Initialize the particle population
Initialize parameters
while (t < Max number of iteration)
for each particle with position x_p
        calculate fitness value f(x_p)
        if f(x_p) is better than pbest_p then
                pbest_p ← x_p
        endif
        if f(pbest_p) is better than gbest then
                gbest ← pbest_p
        endif
end for
update w according to equation (5)
for each particle with position x_p
        update c1, c2 according to equation (3), (4)
        calculate velocity of each particle by equation (6)
        update position of each particle by equation (7)
end for
if rand (0,1) < prob
        run GWO
        x_p = position of the best wolf
endif
t = t + 1
end while
return gbest
```

The solution for the APGWO-wrapper is a binary array, with a dimension of $1 \times n$, where n is the total number of features. Selected features will take a value of 1, and 0 otherwise. The parameters set for the algorithms are as follows: 20 search agents (for PSO main loop), 20 search agents (for nested GWO loop), 20 iterations for main PSO loop, 5 iterations for nested GWO, wMax = 0.9, wMin = 0.2. The fitness function is defined as:

$$\textit{Minimize}\ w \times E_t + (1 - w) \times E_v \tag{8}$$

where $w = 0.8$, E_t is the error rate on the training set, E_v is the error rate on the validation set. The model considered is a multi-layer perceptron (MLP) with 3 hidden layers, with the number of neurons being 30, 20, and 10, respectively. Activation functions of the hidden layers are ReLU. The optimizer for the MLP is Stochastic Gradient Descend (GSD) with a learning rate of 0.01. The output layer contains 3 neurons with Softmax activation function, corresponding to 3 types of infested blobs, ignoring the healthy ones.

For the ANN approach, we extracted 8246 infested blobs and split them in the same manner as described in

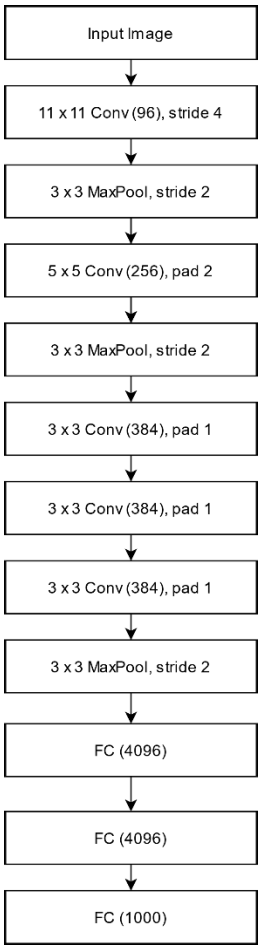

FIGURE 15. Architecture of AlexNet.

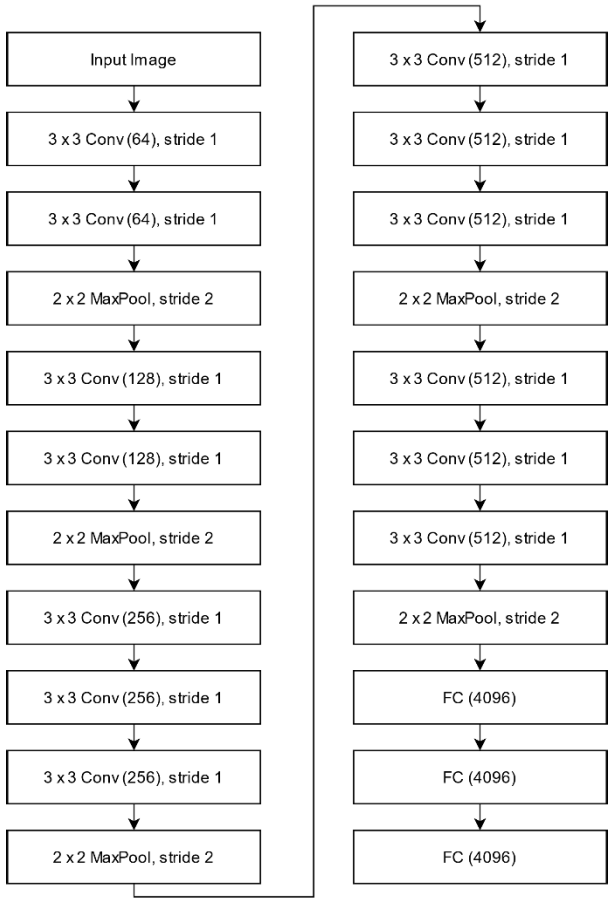

FIGURE 16. Architecture of VGG16.

Section III.A. All the models are programmed using Python language and implemented on a Desktop PC having GPU GTX 1070 which has 1920 CUDA cores with processor Intel(R) Core(TM) i7-7700 at 3.6 GHz, 32 GB of DDR4 Random Access Memory (RAM), and a Solid State Drive (SSD) of 128 GB.

The metrics used in this work are:

$$Accuracy = \frac{Number\ of\ correctly\ classified\ samples}{Total\ number\ of\ samples} \quad (9)$$

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (12)$$

Precision determines how accurate a model is based on the true predicted value, how many of them are actually ''true''. Recall is the ratio of correctly predicted label to all the sample of in the same class. F1 score is defined as the weighted average or harmonic mean of Precision and Recall.

IV. RESULTS

Without TL, VGG16 achieves 77.16% training accuracy, and 76.74% of testing accuracy, while AlexNet achieves

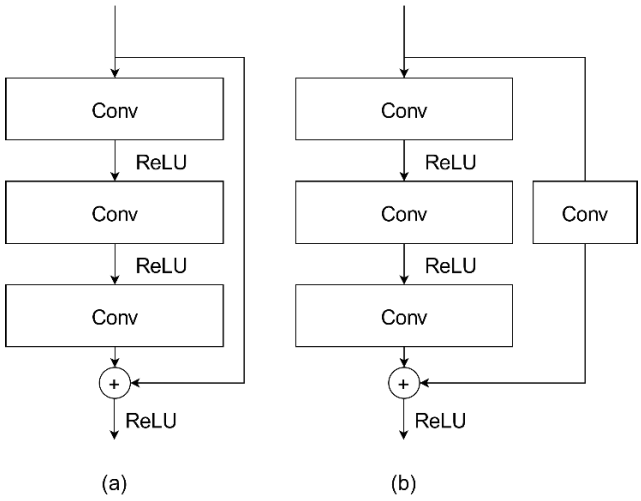

FIGURE 17. Main components of ResNet-50: (a) Identity block, (b) Convolutional block.

69.73% of training accuracy and 70.31% testing accuracy. As expected, with more convolutional layers, the VGG16 model shows a better ability to remove unwanted information. This helps with the prediction accuracy as shown in Fig. 9 and Fig. 10. With TL, VGG16's performance is

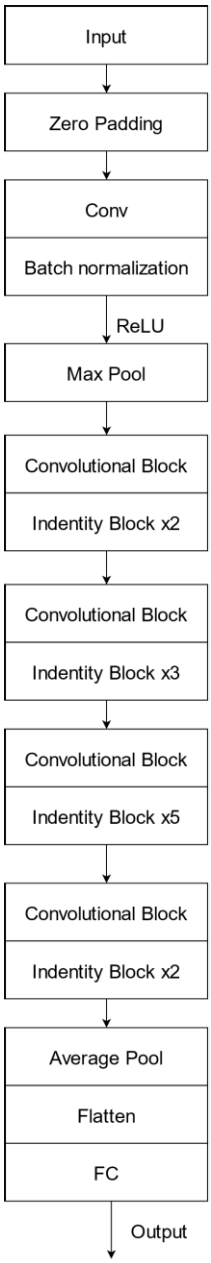

FIGURE 18. Architecture of ResNet-50.

TABLE 4. Performances of ANN In recognizing infected blobs.

Class	Anthracnose blob	Gall Midge blob	Powdery Mildew blob
Recall	0.81	0.92	0.75
Precision	0.90	0.79	0.93
F1- score	0.85	0.85	0.83

TABLE 5. Testing performances of ANN.

Class	Anthracnose	Gall Midge	Healthy	Powdery Mildew
Recall	1.00	0.62	0.97	0.67
Precision	0.90	0.73	1.00	0.67
F1- score	0.95	0.67	0.99	0.67

TABLE 6. Performances comparison of all models.

Models	Class	1	2	3	4
AlexNet	Recall	0.93	0.36	0.96	0.11
w/o TL	Precision	0.58	0.57	0.96	1.00
	F1- score	0.71	0.44	0.96	0.20
VGG16	Recall	1.00	0.44	0.92	0.10
w/o TL	Precision	0.77	0.54	0.96	0.25
	F1- score	0.87	0.48	0.94	0.14
ResNet50	Recall	0.65	0.08	1.00	0.44
w/o TL	Precision	0.63	1.00	0.69	1.00
	F1- score	0.63	0.15	0.81	0.62
AlexNet	Recall	0.94	0.75	1.00	0.23
with TL	Precision	0.78	0.71	0.90	1.00
	F1- score	0.85	0.73	0.95	0.38
VGG 16	Recall	0.93	0.50	0.97	0.42
with TL	Precision	0.74	0.80	1.00	0.42
	F1- score	0.83	0.62	0.98	0.42
ResNet 50					
with TL	Recall	1.00	0.38	1.00	0.56
	Precision	0.82	1.00	1.00	0.63
	F1- score	0.90	0.48	1.00	0.59
ANN	Recall	1.00	0.62	0.97	0.67
	Precision	0.90	0.73	1.00	0.67
	F1- score	0.95	0.67	0.99	0.67

1: Anthracnose, 2: Gall Midge, 3: Healthy, 4: Powdery Mildew

increased to 82.2% training and 78.64% testing accuracy, while those of AlexNet are 85.54%, and 79.92%, respectively. The training process is much more stable with TL.

Before applying TL, ResNet-50 reaches 64.13% training accuracy and 68.73% testing accuracy. After TL is applied, the performances of both models are improved to 86.58% and 84.88%. The training process can be seen in Fig. 11.

After running the Wrapper algorithm, we obtained the selected features as shown in Table 3.

Based on the features obtained by our algorithm, we have made several observations as follows:

- The majority of features related to the Blue channel are not selected at all, except for Kurtosis.

- Most texture features (GLCM) are used since the textures of the infected leaves are quite different from those of the healthy ones.

- Several geometrical features are not used since all blobs are normalized to a standard size of 20 $\times$ 20, therefore, differences between features like axis lengths are not important.

- Most compressed HSV density features are used. In the near future, we plan to increase the number of features in this category by using more bins for H, S, or V channel, also for R, G, and B ones. For example, if we use 16, 4, and 4 bins for H, S, and V channels respectively, the total compressed HSV density features will increase to 16*4*4 = 256 and we

TABLE 7. Glossary.

ACRONYMS	
DL	Deep Learning
CNN	Convolutional Neural Network
ML	Machine Learning
ANN	Artificial Neural Network
FS	Feature Selection
GWO	Grey Wolf Optimizer
PSO	Particle Swarm Optimization
APGWO	Adaptive Particle-Grey Wolf Optimizer
TL	Transfer Learning
ResNet	Residual Network
MLP	Multilayer Perceptron
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descend
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
pbest	Personal best solution of particle
gbest	Global best solution
Conv	Convolutional layer
FC	Fully-connected layer

can increase the prediction capability of our model regarding these features. Fig. 12 shows the heatmap indicating the correlation between the 81 selected features.

For the ANN with FS approach, the model attains 91.32% training accuracy and 85.45% testing accuracy for recognizing type of infected blobs, as shown in Fig. 13. Other metrics can be found in Table 4.

For the ANN approach, the program takes in a leaf image and classify the infected blobs in that one. If more than 50% of the infected blobs on a leaf belong to one type of disease, we conclude that the leaf is infested with that disease. If there is no blob detected, then the leaf is healthy. An example of segmentation results of the ANN can be seen in Fig. 14. We took out the same testing sample which is used for ResNet-50, the CNN that performed best in this study, to compare the two models. The ANN reached 89.41% of testing accuracy. Recall, Precision and F1-score can be found in Table 5. Table 6 summarizes the performances of all models. It can be seen that the ANN's metrics are very competitive to those of the CNNs for the case of infected leaves, although it still misclassifies some healthy leaves.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a multi-class mango leaf disease classification using deep neural networks. At first, a wrapper-based feature selection approach using an Adaptive Particle-Grey Wolf metaheuristic (APGWO) was performed to select 81 features out of the originally proposed 114 features. These features are selected as inputs for the MLP for the classification task. The approach developed outperformed deep learning models such as VGG, AlexNet, ResNet-50, which are already enhanced with transfer learning (89.41% vs 78.64%, 79.92%, and 84.88%, respectively). Furthermore, the MLP network is much smaller, therefore, leads to faster performance. This is preferable since we want to implement this algorithm on resource-constrained devices such as smartphones. Some of our future works are given as follows:

1. Further fine-tuning the parameters of the MLP model, such as the number of layers, the number of hidden nodes, as well as the activation function.
2. Working with plantations to obtain a more diverse dataset.
3. Fine-tuning the feature selection algorithm.
4. Build a complete disease monitoring system that can be deployed on several platforms.

APPENDIX

See Table 7.

REFERENCES

[1] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, *arXiv:1511.08060*. [Online]. Available: http://arxiv.org/abs/1511.08060

[2] J. G. A. Barbedo, "Factors influencing the use of deep learning for plant disease recognition," *Biosyst. Eng.*, vol. 172, pp. 84–91, Aug. 2018, doi: 10.1016/j.biosystemseng.2018.05.013.

[3] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[4] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant disease detection and classification by deep learning," *Plants*, vol. 8, no. 11, p. 468, Oct. 2019, doi: 10.3390/plants8110468.

[5] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Comput. Electron. Agricult.*, vol. 145, pp. 311–318, Feb. 2018, doi: 10.1016/j.compag.2018.01.009.

[6] A. K. Rangarajan, R. Purushothaman, and A. Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Comput. Sci.*, vol. 133, pp. 1040–1047, Jan. 2018, doi: 10.1016/j.procs.2018.07.070.

[7] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers Plant Sci.*, vol. 7, p. 1419, Sep. 2016, doi: 10.3389/fpls.2016.01419.

[8] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agricult.*, vol. 161, pp. 272–279, Jun. 2019, doi: 10.1016/j.compag.2018.03.032.

[9] J. Shijie, J. Peiyi, H. Siping, and S. Haibo, "Automatic detection of tomato diseases and pests based on leaf images," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 2510–2537, doi: 10.1109/CAC.2017.8243388.

[10] U. P. Singh, S. S. Chouhan, S. Jain, and S. Jain, "Multilayer convolution neural network for the classification of mango leaves infected by anthracnose disease," *IEEE Access*, vol. 7, pp. 43721–43729, 2019, doi: 10.1109/ACCESS.2019.2907383.

[11] S. D. Khirade and A. B. Patil, "Plant disease detection using image processing," in *Proc. Int. Conf. Comput. Commun. Control Autom.*, Feb. 2015, pp. 768–771, doi: 10.1109/ICCUBEA.2015.153.

[12] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agricult.*, vol. 4, no. 1, pp. 41–49, Mar. 2017, doi: 10.1016/j.inpa.2016.10.005.

[13] A. H. Kulkarni and A. Patil, "Applying image processing technique to detect plant diseases," *Int. J. Mod. Eng. Res.*, vol. 2, no. 5, pp. 3661–3664, 2012.

[14] S. S. Chouhan, A. Kaul, U. P. Singh, and Madhav Institute of Technology Science, "A database of leaf images: Practice towards plant conservation with plant pathology," Mendeley Data, V1, 2019, doi: 10.17632/hb74ynkjcn.1.

[15] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. 10, pp. 1205–1224, 2004.

[16] J. O. Pedersen and Y. Yang, "A comparative study on feature selection in text categorization," in *Proc. ICML. 14th Int. Conf. Mach. Learn.*, 1997, pp. 412–420. [Online]. Available: https://pdfs.semanticscholar.org/f87c/96a5f22a741fa1015089f15beb78cda95aa2.pdf

[17] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015, doi: 10.1016/j.snb.2015.02.025.

[18] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997, doi: 10.1109/34.574797.

[19] C. De Stefano, F. Fontanella, C. Marrocco, and A. S. D. Freca, "A GA-based feature selection approach with an application to handwritten character recognition," *Pattern Recognit. Lett.*, vol. 35, pp. 130–141, Jan. 2014, doi: 10.1016/j.patrec.2013.01.026.

[20] E. Zorarpacı and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016, doi: 10.1016/j.eswa.2016.06.004.

[21] J. Too, A. Abdullah, N. M. Saad, N. M. Ali, and W. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," *Computers*, vol. 7, no. 4, p. 58, Nov. 2018, doi: 10.3390/computers7040058.

[22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[23] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016, doi: 10.1016/j.neucom.2015.06.083.

[24] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–38, Feb. 2008, doi: 10.1016/j.compbiolchem.2007.09.005.

[25] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, "Text feature selection using ant colony optimization," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6843–6853, Apr. 2009, doi: 10.1016/j.eswa.2008.08.022.

[26] X. He, Q. Zhang, N. Sun, and Y. Dong, "Feature selection with discrete binary differential evolution," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, 2009, pp. 327–330, doi: 10.1109/AICI.2009.438.

[27] F. Pedregosa, G. Varoquaux, and A. Gramfort, "Scikit-learn: Machine learning in Python title," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: https://jmlr.org/papers/v12/pedregosa11a.html

[28] G. Maragatham and S. Mansoor Roomi, "A review of image contrast enhancement methods and techniques," *Res. J. Appl. Sci., Eng. Technol.*, vol. 9, no. 5, pp. 309–326, Feb. 2015, doi: 10.19026/rjaset.9.1409.

[29] M. A. Rahman, S. Liu, C. Y. Wong, G. Jiang, and N. Kwok, "Image contrast enhancement for brightness preservation based on dynamic stretching," *Int. J. Image Process.*, vol. 9, no. 4, pp. 241–253, 2015.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105, doi: 10.1145/3065386.

[31] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–14. [Online]. Available: http://arxiv.org/abs/1409.1556.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[33] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, J. Santamaría, Y. Duan, and S. R. Oleiwi, "Towards a better understanding of transfer learning for medical imaging: A case study," *Appl. Sci.*, vol. 10, no. 13, p. 4523, Aug. 2020, doi: 10.3390/app10134523.

[34] L. Alzubaidi, M. A. Fadhel, O. Al-Shamma, J. Zhang, and Y. Duan, "Deep learning models for classification of red blood cells in microscopy images to aid in sickle cell anemia diagnosis," *Electronics*, vol. 9, no. 3, p. 427, Mar. 2020, doi: 10.3390/electronics9030427.

[35] L. Alzubaidi, O. Al-Shamma, M. A. Fadhel, L. Farhan, J. Zhang, and Y. Duan, "Optimizing the performance of breast cancer classification by employing the same domain transfer learning from hybrid deep convolutional neural network model," *Electronics*, vol. 9, no. 3, p. 445, Mar. 2020, doi: 10.3390/electronics9030445.

[36] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[37] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems*. Cambridge, MA, USA: Academic, 1994, ch. VIII.5, pp. 474–485.

[38] F. Albregtsen, "Statistical texture measures computed from gray level coocurrence matrices," Lab. Dept. Informat., Univ. Oslo, Oslo, Norway, Tech. Rep., 2008. [Online]. Available: https://www.uio.no/studier/emner/matnat/ifi/INF4300/h08/undervisningsmateriale/glcm.pdf, doi: 10.2307/302397.

[39] D. Son and P. N. Tan, "Capacitated vehicle routing problem—A new clustering approach based on hybridization of adaptive particle swarm optimization and grey Wolf optimization," in *Evolutionary Data Clustering: Algorithms, and Applications*. Singapore: Springer, 2019.

[40] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN-Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[41] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999, doi: 10.1109/4235.771163.

TAN NHAT PHAM is currently a Student majoring in industrial engineering with the School of Industrial Engineering and Management, International University, Vietnam National University, Vietnam. His research interests include metaheuristic algorithms and applied artificial intelligence.

LY VAN TRAN received the B.Eng. degree in mechanical engineering in 1996, and the M.Eng. degree in manufacturing systems engineering in 2000. From 2000 to 2019, he was in various industries with a wide range of roles from operations to senior management. His research interests include operation research and product development.

SON VU TRUONG DAO received the B.Eng. degree in aeronautical engineering, the M.Sc. degree in manufacturing systems and technology, and the Ph.D. degree in sensors technology, in 2004, 2005, and 2010, respectively. From 2006 to 2008, he was with Hylax Ltd., Singapore, designing high power laser systems. From 2010 to 2012, he was a Postdoctoral Fellow with Kindai University, Japan. From 2012 to 2015, he worked as a Senior Scientist with Ritsumeikan University, Japan. His research interests include development of advanced imaging sensors and systems and applied artificial intelligence.