

Article

A Feature Selection Based on Improved Artificial Hummingbird Algorithm Using Random Opposition-Based Learning for Solving Waste Classification Problem

Mona A. S. Ali ^{1,2,*}, Fathimathul Rajeeana P. P. ^{1,†} and Diao Salama Abd Elminaam ^{2,3,4,*}

¹ Computer Science Department, College of Computer Science and Information Technology, King Faisal University, Al Ahsa 400, Saudi Arabia; fatimah.rajeena@kfu.edu.sa

² Faculty of Computers and Artificial Intelligence, Benha University, Benha 12311, Egypt

³ Computer Science Department, Faculty of Computer Science, Misr International University, Cairo 12585, Egypt

⁴ Faculty of Information Technology, Middle East University, Amman 11831, Jordan

* Correspondence: m.ali@kfu.edu.sa (M.A.S.A.); diaa.salama@fci.bu.edu.eg or diaa.salama@miuegypt.edu.eg or ds_desert@yahoo.com (D.S.A.E.)

† These authors contributed equally to this work.

Abstract: Recycling tasks are the most effective method for reducing waste generation, protecting the environment, and boosting the overall national economy. The productivity and effectiveness of the recycling process are strongly dependent on the cleanliness and precision of processed primary sources. However, recycling operations are often labor intensive, and computer vision and deep learning (DL) techniques aid in automatically detecting and classifying trash types during recycling chores. Due to the dimensional challenge posed by pre-trained CNN networks, the scientific community has developed numerous techniques inspired by biology, swarm intelligence theory, physics, and mathematical rules. This research applies a new meta-heuristic algorithm called the artificial hummingbird algorithm (AHA) to solving the waste classification problem based on feature selection. However, the performance of the AHA is barely satisfactory; it may be stuck in optimal local regions or have a slow convergence. To overcome these limitations, this paper develops two improved versions of the AHA called the AHA-ROBL and the AHA-OBL. These two versions enhance the exploitation stage by using random opposition-based learning (ROBL) and opposition-based learning (OBL) to prevent local optima and accelerate the convergence. The main purpose of this paper is to apply the AHA-ROBL and AHA-OBL to select the relevant deep features provided by two pre-trained models of CNN (VGG19 & ResNet20) to recognize a waste classification. The TrashNet dataset is used to verify the performance of the two proposed approaches (the AHA-ROBL and AHA-OBL). The effectiveness of the suggested methods (the AHA-ROBL and AHA-OBL) is compared with that of 12 modern and competitive optimizers, namely the artificial hummingbird algorithm (AHA), Harris hawks optimizer (HHO), Salp swarm algorithm (SSA), aquila optimizer (AO), Henry gas solubility optimizer (HGSO), particle swarm optimizer (PSO), grey wolf optimizer (GWO), Archimedes optimization algorithm (AOA), manta ray foraging optimizer (MRFO), sine cosine algorithm (SCA), marine predators algorithm (MPA), and rescue optimization algorithm (SAR). A fair evaluation of the proposed algorithms' performance is achieved using the same dataset. The performance analysis of the two proposed algorithms is applied in terms of different measures. The experimental results confirm the two proposed algorithms' superiority over other comparative algorithms. The AHA-ROBL and AHA-OBL produce the optimal number of selected features with the highest degree of precision.

Keywords: waste classification; deep features; artificial hummingbird algorithm; swarm intelligence; wrapper feature selection

MSC: 2020



Citation: Ali, M.A.S.; P. P., F.R.; Salama Abd Elminaam, D. A Feature Selection Based on Improved Artificial Hummingbird Algorithm Using Random Opposition-Based Learning for Solving Waste Classification Problem. *Mathematics* **2022**, *10*, 2675. <https://doi.org/10.3390/math10152675>

Academic Editors: Adrian Deaconu, Petru Adrian Cotfas and Daniel Tudor Cotfas

Received: 7 June 2022

Accepted: 21 July 2022

Published: 29 July 2022

Corrected: 6 May 2023

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Modern consumption and manufacturing have made trash a global problem. The world manufactured 8.3 billion metric tons of plastic between 1950 and 2015, and 6.3 billion tons of plastic was dumped. 9% of garbage is recycled, 14% is burned, and 79% is correctly disposed of, according to research [1]. Plastic decomposes in 400 years. A million-year-old glass bottle will decompose. Recycling garbage (plastic, glass, and metal) harms the environment too much to be taken lightly. Each trash item dumped in oceans, farms, or other essential regions endangers all life. The economy suffers [2].

Recycling and reusing garbage is common. Many countries have launched legal recycling studies. This has affected each country's social-economic culture [3]. Recycling is key to saving resources. Waste categorization is critical and time-consuming for calculating recyclable waste. Historically, trash was sorted manually. With more urban trash, untrained sorters are engaged [4]. Because of these difficulties, recycling trash became necessary [5]. Municipalities, ministries, and non-profits require a fast, straightforward trash classification system.

CNNs enhance rubbish classification accuracy and resource recycling efficiency [6–8]. Deeper representations yield more semantically rich characteristics. Linking network layers has shown tremendous features; Convolution kernels or filters extract features. Therefore, a receptive field should cover the accuracy and efficiency. However, multiple studies revealed that this technique lacks context. CNNs are feed-forward artificial neural networks inspired by the animal visual cortex. Deep learning algorithms are considered the most dependable. Their implementation is suitable for modern real-time applications [9].

Real-world problems contain a lot of data, making processing tough. Datasets have attributes/features. Data extraction doesn't require all features. Redundant features might reduce a model's performance. Feature reduction reduces each dataset's size while maintaining accuracy. Feature reduction involves feature selection. Feature extraction adds new features from existing datasets, whereas feature selection selects needed features.

Optimization and meta-heuristic algorithms are currently two of the hottest topics in computer science due to their presence in several domains, such as feature selection problems [10–12], facial recognition [13,14], opinion mining [15,16], the identification of parameters in photovoltaic applications [17,18], economic load dispatch problems [19,20], bin packing problems [21,22], software cost estimations [23], traveling salesman problems [24], constrained engineering problems [25], and continuous optimization problems [26,27]. According to the no free lunch (NFL) theory [28], no algorithm can discover the optimal solution to all problems; hence, numerous optimization approaches exist in the literature. In other words, if an algorithm can determine the optimal answer for a particular problem, it will fail for other types. Most of previous theorem permits researchers to build and enhance current methods.

Based on their sources of inspiration, meta-heuristic algorithms can be separated into three subcategories, namely swarm-based (biogeography-based [29], social network [30], and biology-based), physics-based, and differential-evolution-based [31] optimizers.

Swarm-inspired meta-heuristics include algorithms that replicate the social and biological characteristics of organisms, such as mating, labor division, foraging, navigation, and self-organization. Examples of social network optimizers include the multi-swarm whale optimization algorithm [32], genetic algorithm [33], multitracker optimization algorithm [34], and parallel multiobjective evolutionary algorithm [35].

Moreover, examples of biogeography-based optimizers include the evolutionary optimization algorithms (EOAs) [36], cuckoo search algorithm (CSA) [37], krill herd algorithm (KHA) [38], hybrid PSO-GA algorithm [39], shuffled frog leaping algorithm (SFLA) [40], swarm intelligence optimization algorithms (SIOAs) [41], Laplacian biogeography-based optimization algorithm (LBOA) [42], biogeography-based optimization algorithm (BOA) [43], and population-based algorithms (PBAs) [44].

There are also different modified versions of BBOs, such as modified versions of BBOs using migration-based modifications [45–47], mutation-based modifications [29,48,49], and others [50,51]. Furthermore, there are different modified versions of BBO-based hybridization, such as hybridizations with local search algorithms [52–54] and hybridizations with other population-based algorithms [55–57].

Examples of biology-based optimizers include the grey wolf optimizer (GWO) [58], whale optimization algorithm (WOA) [59,60], firefly algorithm (FA) [61], Salp swarm algorithm (SSA) [62–64], emperor penguin colony optimizer [65], squirrel search algorithm [66,67], slime mold algorithm (SMA) [68], barnacles mating optimizer (BMO) algorithm [69], tunicate swarm algorithm (TSA), and artificial hummingbird algorithm (AHA) [70].

The second category, physics-inspired meta-heuristics, contains algorithms influenced by scientific facts or principles. Examples include simulated annealing [71], big bang BigCrunch [72], and the gravitational search algorithm (GSA) [73], lightning search algorithm [74], black hole algorithm [75], sine cosine algorithm (SCA) [76], artificial electric field algorithm [77], arithmetic optimization algorithm (AOA) [78], multi-verse optimizer (MVO), and Henry gas solubility optimizer [79].

The third category, differential evolution algorithms, is motivated by concepts of biological evolution, such as the genetic algorithm (GA) [80], evolutionary programming [81], biogeography-based optimization [82], memetic algorithm [83], multipopulation differential evolution algorithm (MPDE) [84], self-adaptive mutation differential evolution (SaMDE) [85], fitness-adaptive differential evolution (FiADE) [86], modified differential evolution (MDE) and MDE with a pbest crossover (MDE-pBX) [87], teaching-and-learning-based self-adaptive differential evolution (TLBSaDE) [88], modified differential evolution algorithm (MDE) [89], adaptive differential evolution [90], and differential evolution with a crossover rate repair [91].

Moreover, hybrid differential evolution algorithms use recent swarm intelligence algorithms, such as hybrids of differential evolution and the artificial bee colony (ABC) algorithm [92], differential evolution and the ant colony optimizer (ACO) [93], differential evolution and the bacterial foraging-based optimizer (BFO) [94], differential evolution and the gravitational search algorithm (GSA) [95], differential evolution and the invasive weed optimizer (IWO) [96], differential evolution and the firefly algorithm (FFA) [97], and differential evolution and the fireworks algorithm (FWA) [98].

The previously proposed meta-heuristics for the feature selection (FS) problem all suffer from a slow convergence, poor scalability [99,100], and lack of precision and consistency. These limitations motivated the current study, which suggests a novel AHA-based algorithm for FS tasks.

Therefore, in this study, we propose a modification to the current optimization method known as the AHA, which is an innovative bioinspired meta-heuristic algorithm. The AHA simulates wild hummingbirds' incredible flight capabilities and cunning feeding strategies. An adaptive opposition strategy is proposed to enable the original algorithm to achieve more precise results with more hard challenges using two operators (ROBL and OBL).

The main contributions of this paper can be summarized as follows:

- In this study on solving the feature selection problem, the AHA is enhanced for the first time.
- An enhanced version of the AHA is proposed based on two operators: random opposition-based learning (ROBL) and opposition-based learning (OBL).
- The two proposed models are compared with the original algorithm and 12 different algorithms.
- The study applies the modified algorithms AHA-ROBL and AHA-OBL to the TrashNet database by using two pre-trained networks: VGG19 and ResNet.
- The two proposed algorithms each demonstrate a greater robustness and stability than other recent algorithms.

Our paper is structured as follows: Section 2 conducts a literature review, while Section 3 discusses the fundamentals of the AHA-ROBL optimization technique for pre-trained neural networks. Section 4 summarizes the acquired results regarding fitness, accuracy, and feature selection.

2. Literature Review

In recent years, considerable research has been conducted on garbage image classification. This paper will present the work of domestic and international scholars in the fields of image recognition and waste classification

2.1. Waste Recycling Using Traditional Machine-Learning Algorithms

Different machine-learning algorithms have been applied to the TrashNet data. Yang et al. achieved an accuracy rate of 63% [101] using the SVM algorithm and Costa et al. achieved an accuracy of 88% using the kNN algorithm [102]. Satvilkar classified garbage images from the TrashNet dataset with a 62.61% accuracy [103] using the RF algorithm and classified garbage images from the TrashNet dataset using the XGBoost algorithm with an accuracy of 70.1%.

2.2. Waste Recycling Using Deep-Learning Algorithms

Deep- and machine-learning models have been combined to classify trash types. Researchers in [103] conducted an experiment in which they examined solely recyclable waste material classified into five distinct categories. The CNN, k-nearest neighbor (kNN), random forest (RF), and SVM models were all used, with the CNN model achieving the highest classification accuracy of 89.91%. [104] evaluated the kNN, RF, SVM and VGG16 models in combination. A processed dataset was created using photos of four distinct recycling materials with a success rate of 93%. Zhu et al. [105] established an identification approach for plastic solid waste (PSW) chemicals classified into six types based on near-infrared (NIR) reflectance spectroscopy, principal component analysis (PCA), and the support vector machine (SVM) model with a 97.5% classification accuracy. Zkan et al. [106] classified garbage into plastic and non-plastic categories.

2.3. Waste Recycling Using Deep-Transfer Learning

In the following section, detailed descriptions of this dataset are provided. Several studies utilizing the TrashNet dataset to evaluate proposed solutions to the trash classification problem are summarized in [3,8,107].

First, Aral et al. classified trash from the TrashNet dataset using different transfer learning models. According to the experimental findings, the DenseNet121 model had the highest accuracy, achieving 95% [107].

Then, Ruiz et al. used different CNN models and achieved an average accuracy of 88.66% for the TrashNet dataset, producing the best performance results. This method, which ResNet Ruiz denotes, was reimplemented in our experiments [8].

Several well-known CNN models for image classification, such as ResNext [108], ImageNet [109], VGG [110], ResNet [111], and DenseNet [112], can also be used as base models for trash classification. This study determined that among the CNN models listed above, ResNext is the best model for transfer learning to classify trash.

AHA demonstrates an extremely competitive performance. It demonstrates an effectiveness in optimization issues. Moreover, this algorithm has an advantage over other algorithms. Its straightforward procedure, low computational cost, significant convergence speed, relatively close solutions, independence from problems, and gradient-free nature make it a desirable algorithm [113–116]. In the present paper, enhanced AHA algorithms are used to select the most optimal features in the waste classification problem. We propose two new enhanced approaches based on AHA for FS, namely the AHA-ROBL and AHA-OBL, based on the kNN classifier.

3. Procedure and Methodology

Figure 1 illustrates the proposed framework for an improved artificial hummingbird algorithm using random opposition-based learning for solving waste classification problems based on feature selection, which contains seven significant steps:

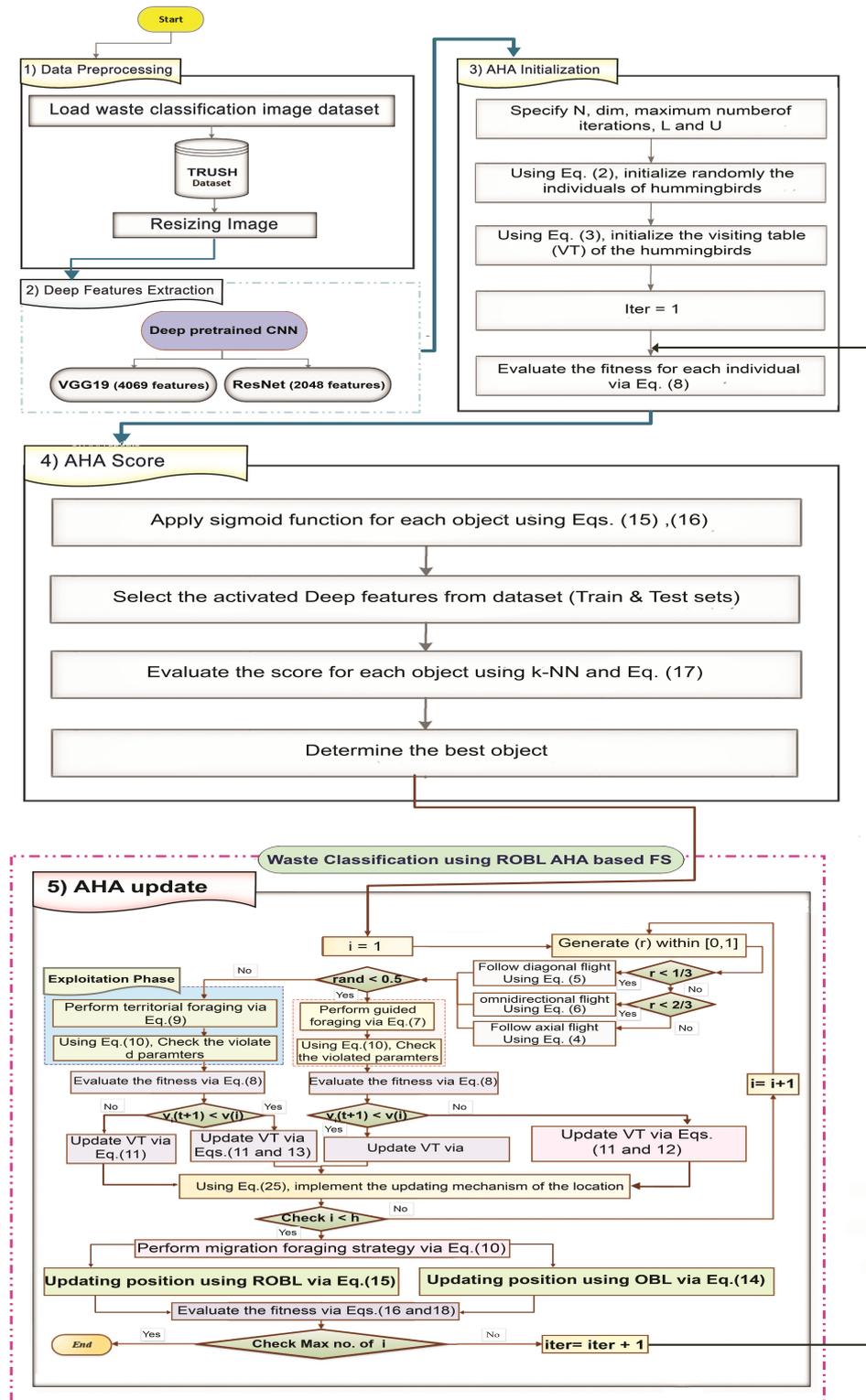


Figure 1. The design framework of the AHA-ROBL based FS for waste classification.

- Data collection
- Data pre-processing
- Feature extraction techniques using pre-trained deep-learning models (VGG19 and Resnet20)
- Waste classification with the AHA-ROBL using AHA initialization followed by AHA scoring and AHA updating using an exploration mode and the AHA and an exploitation mode using ROBL
- Prediction and evaluation metrics

3.1. Dataset Description

The dataset used and implemented in this research is the TrashNet dataset. The TrashNet dataset includes 2527 images classified into six categories: cardboard, glass, metal, paper, plastic, and rubbish. This study supplemented the original dataset to build a huge dataset. The dataset augmentation resulted in 2527 images of horizontal flipping, 2527 images of vertical flipping, and 2527 random 25° rotations, resulting in 10,108 waste images. Additionally, this study compared the outcomes using 2527 photos and 10,108 photographs. The dataset was partitioned, with 90% and 10% of each class randomly assigned to training and testing sets, respectively [117]. Figure 2 shows examples of each category.



Figure 2. TrashNet dataset with one example of each class: (a) metal, (b) glass, (c) cardboard, (d) paper, (e) plastic, and (f) trash.

3.2. Feature Extraction Using Pre-Trained CNN

The process of feature extraction using a pre-trained CNN is introduced in this section. CNNs are composed of three layers: convolutional, pooling, and fully connected layers. The most critical layers are the convolutional and pooling layers. By convolving an image area with numerous filters, a convolution layer is utilized to extract features. Due to the higher layer count, a CNN can interpret the features in its input image more precisely. The pooling layer compresses the output mapping of the convolution. Four pre-trained networks can be used in computer vision tasks, such as image generation, image classification, image captioning, and many others. The four types are VGG19, Inception V3, GoogLeNet, ResNet50, and AlexNet.

In this research, two of these pre-trained networks were used: VGG19 and ResNet50. Their benefits contributed to an improved prediction performance while avoiding over-fitting traditional ANN models. The following section will explain the two pre-trained models used in this paper.

3.2.1. VGG19

The VGG neural network is a 19-layer convolutional neural network. Simonyan and Zisserman developed and trained it in 2014 at the University of Oxford. The details can be found in their 2015 paper “Very Deep Convolutional Networks for Large-Scale Image Recognition”. Additionally, the VGG19 network was trained using images from the ImageNet collection totaling more than 1 million images. Naturally, you may import the model with training weights from ImageNet. Up to 1000 items can be classified using this pre-trained network. The network was trained using colorful images with a resolution of 224×224 pixels (Figure 3 [118]).

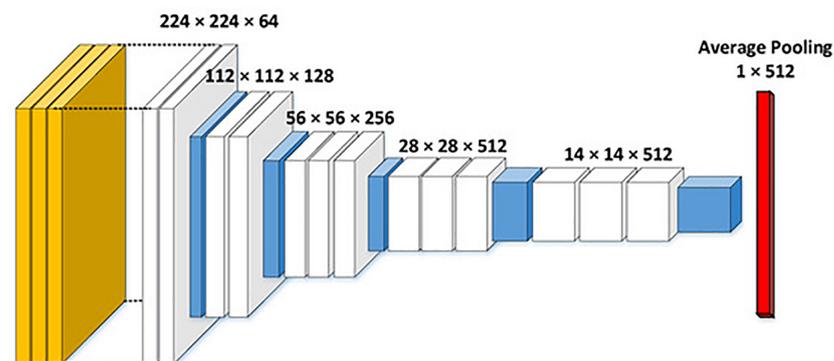


Figure 3. VGG19 architecture.

3.2.2. ResNet50

ResNet50 is a 50-layer convolutional neural network. As with VGG19, it can classify up to 1000 objects and was trained on 224×224 pixel colored images. Additionally, this model was trained on over 1 million photos from the ImageNet collection. Microsoft developed and trained the model in 2015, and the model’s performance results are available in their publication titled “Deep Residual Learning for Image Recognition”.

Figure 4 illustrates a ResNet residual block. As illustrated in the figure, stacked layers execute residual network mapping by establishing shortcut connections that perform the identity mapping (x). Their outputs are added to the residual function F of the stacked layers (x).

An error gradient was determined and propagated to the shallow layers during the deep network’s backpropagation training. This inaccuracy became smaller and smaller as one progressed deeper into the levels until it eventually vanished. This phenomenon is referred to as the gradient vanishing problem in really deep networks. As illustrated in Figures 4 and 5, the problem can be handled via residual learning [119].

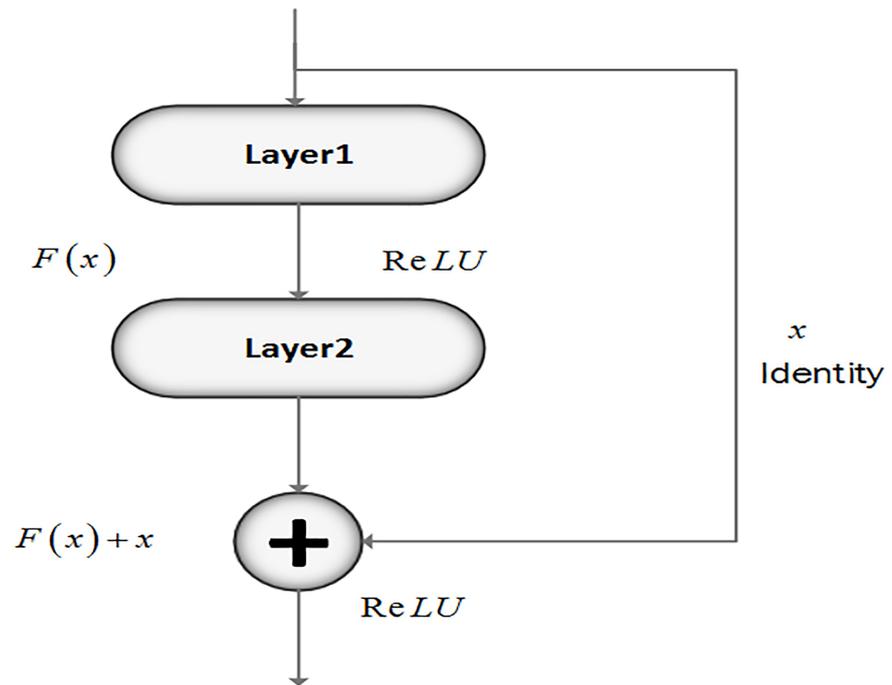


Figure 4. ResNet residual block.

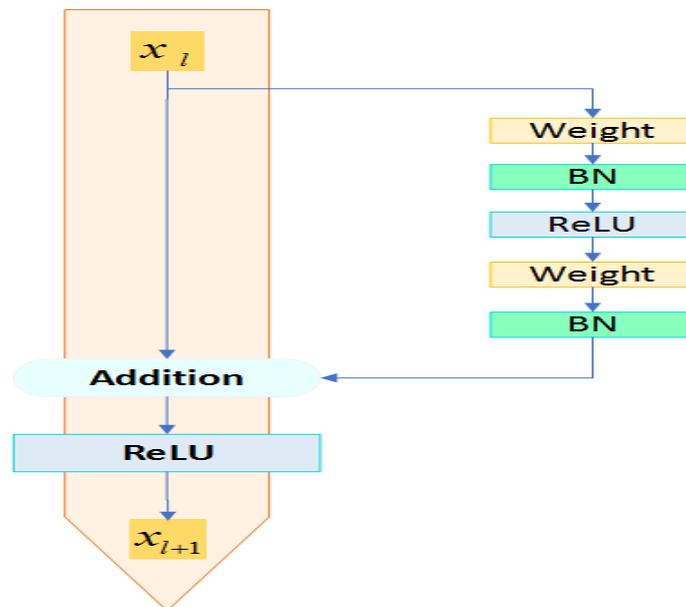


Figure 5. Original residual unit.

The initial residual branch, or unit l , is depicted in Figure 5 within the residual network. Weights, batch normalization (BN), and a corrected linear unit are depicted in the figure ($ReLU$). The following equations were used to determine the input and output of a residual unit (Equation (1)):

$$\begin{aligned}
 y_l &= h(x_l) + F(x_l + W_l) \\
 x_{l+1} &= f(y_l)
 \end{aligned}
 \tag{1}$$

where $h(x_l)$ represents the identity mapping, F represents the residual function, x_l represents the input, and W_l represents the weight coefficient. The identity mapping, which is denoted by $h(x_l) = x_l$, is the foundation of the ResNet architecture. The residual networks

were created for networks with layer counts of 34, 50, 101, and 152. ResNet50 was employed in this investigation. The network is made up of 50 layers.

3.3. Artificial Hummingbird Algorithm (AHA)

The AHA is a brand new bioinspired meta-heuristic algorithm. The AHA simulates the amazing flying abilities and intelligent feeding methods of hummingbirds in the wild. The technique uses three flight-capability foraging strategies: axial, diagonal, and omnidirectional. In addition, foraging strategies (directed, territorial, and migratory) and a visiting table are employed to simulate the memory function of hummingbirds concerning food sources. The technique is straightforward and has few pre-defined parameters that can be modified. Each hummingbird in the AHA is assigned a unique food source from which it can be nourished. A hummingbird can memorize the location and rate of nectar replenishment of this particular food source. It can also recall the time between visits to each food source. These exceptional skills afford the AHA an exceptional capability for locating ideal solutions.

This section describes the steps of the AHA Algorithm 1, which simulate the behavior of hummingbirds. There are three types of flight skills referred to as axial, diagonal, and omnidirectional flights; these skills are employed in foraging strategies [120]. In addition, there are various types of search strategies, such as guided foraging, territorial foraging, and migratory foraging; a visiting table is also created to simulate the memory function of hummingbirds. As aforementioned, the AHA is a new bioinspired optimizer proposed by Mirjalili for solving optimization problems [70]. This algorithm was inspired by the unique flight capabilities and intelligent foraging strategies of hummingbirds.

Algorithm 1 AHA pseudo-code.

```

Define  $N_{pop} = n$  = The size of the population
Define  $N_{iter,max}$ 
Define upper and lower population limits
Initialize the population using Equation (2)
while ( $t_p \leq N_{iter,max}$ ) do
  for (each population determine direction change vector  $D$ ) do
    if ( $rand \leq 1/3$ ) then
      Implement the diagonal flight Equation (5)
    else
      if ( $rand \leq 2/3$ ) then
        Implement the omnidirectional flight using Equation (6)
      else
        Implement the axial flight using Equation (4)
      end if
    end if
  end for
  for Each population foraging behavior update do
    if  $rand \leq 0.5$  then ▷ Exploration operation
      Implement the guided foraging using Equation (7)
    else ▷ Exploitation operation
      Implement the territorial foraging using Equation (9)
    end if
    if  $t_p = 2n$  then
      Implement the migration foraging using Equation (10)
    end if
  end for
  Update positions
  Return the highest value for fitness
   $t_p = t_p + 1$ 
end while

```

The mathematical formulation of the AHA is illustrated by constructing the initial population of X hummingbirds out of N individuals, as shown in Equation (2)

$$X_i = L + r \times (U - L), i = 1, 2, \dots, N \tag{2}$$

where L and U , respectively, represent the upper and lower bounds for a D dimension. r is a random vector in the range of $[0, 1]$. Additionally, a visited table of food sources is created using Equation (3):

$$VT_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ null & i = j \end{cases}, i = 1, \dots, N, j = 1, \dots, N, \tag{3}$$

where for $i = j$, the value of $VT_{i,j}$ becomes null and stands for the food taken by a hummingbird at its specific food source. Additionally, when $i \neq j$ and $VT_{i,j}$ becomes zero, they stand for hummingbird i visiting food source j .

3.3.1. Guided Foraging

In this stage, three flight skills are utilized during foraging, including omnidirectional, diagonal, and axial flight.

The axial flight is defined using Equation (4):

$$D^{(i)} = \begin{cases} 1, & \text{if } i = \text{randi}([1, d]) \\ 0, & \text{else} \end{cases} \quad i = 1, \dots, d \tag{4}$$

The diagonal flight can be expressed using Equation (5):

$$D^{(i)} = \begin{cases} 1, & \text{if } i = Pp(j), j \in [1, k] \\ & , Pp = \text{randperm}(Kp) \\ & , Kp \in [2, [r_1 \cdot (d - 2)] + 1] \\ 0, & \text{else } i = 1, \dots, d \end{cases} \tag{5}$$

The omnidirectional flight is represented using Equation (6):

$$D^i = 1 \quad i = 1, \dots, d \tag{6}$$

where $\text{randi}([1, d])$ represents a random integer between 1 and d , $\text{randperm}(k)$ represents a random permutation of the integers between 1 and k , and $r_1 \in [0, 1]$ represents a random number. Formulating the guided foraging behavior using using Equation (7):

$$V_i(t + 1) = X_{i,t}(t) + a \times D \times (X_i(t) - X_{i,t}(t)), a \in N(0, 1) \tag{7}$$

where $X_{i,t}(t)$ denotes the source of food i at iteration t . $X_{i,t}(t)$ is the target food source that i th hummingbirds visit. The value of X_i can be updated using Equation (8):

$$x_i(t + 1) = \begin{cases} x_i(t), & \text{if } f(x_i(t)) \leq f(v_i(t + 1)) \\ v_i(t + 1), & \text{if } f(x_i(t)) > f(v_i(t + 1)) \end{cases} \tag{8}$$

where f is the fitness value.

3.3.2. Territorial Foraging

A hummingbird is more likely to search for a new food source after visiting its target food source when flower nectar has been consumed as opposed to visiting other present food sources. Consequently, a hummingbird might readily travel to a nearby location within its area, where a possibly superior food supply could be identified. The modeling is given using Equation (9):

$$V_i(t + 1) = X_{i,t}(t) + b \times D \times X_i(t), b \in N(0, 1) \tag{9}$$

3.3.3. Migration Foraging

In the last phase, the AHA algorithm determines the migration coefficient. If a hummingbird’s preferred feeding location runs out of food, it migrates to a more distant feeding location. This hummingbird will abandon the previous food source in favor of the new one, causing the visit table to be modified. The following is a description of a hummingbird’s migration from a nectar source with the lowest nectar-refilling rate to one with a random rate of nectar production (see Equation (10)):

$$X_w(t + 1) = L + r \times (U - L), \tag{10}$$

Here, x_w represents the food source with the lowest fitness value.

A crucial component of the AHA algorithm is the visiting table. Using Equations (11)–(13), the visiting table is updated for each hummingbird.

$$VT_{i,k} = VT_{i,k} + 1, \quad \text{if } k \neq i \& k \neq \text{target}, k = 1, 2, \dots, h_n \tag{11}$$

$$VT_{i, \text{target}} = 0 \tag{12}$$

$$VT_{i,k} = \max_{L \neq i \& L \in h_n} (VT_{i,L}) + 1, \quad \text{if } k \neq i, k = 1, 2, \dots, h_n \tag{13}$$

This visiting table indicates the length of time since the same hummingbird’s last visit to each food source. A long interval between visits indicates a high frequency of visits.

3.4. Opposition-Based Learning (OBL)

In the first proposed approach (the AHA-OBL), OBL is applied.

OBL is an effective search strategy for avoiding stagnation in possible solutions [121]. OBL, which was proposed by Tizhoosh, improves the exploitation capability of a search mechanism [122]. In meta-heuristic algorithms, a convergence occurs rapidly when initial solutions are relatively close to an optimal location; otherwise, a late convergence is predicted. Here, the OBL strategy generates new solutions by considering search regions that may be nearer to the global optimal solution.

To better understand the OBL, assume the opposite of the real number $x \in [lb, ub]$ can be calculated as $Opp = (ub + lb) - x$, where Opp is the variable opposite var. Consequently, for N-dimensional real numbers, the previous formulation can be generalized as demonstrated by Equation (14):

$$\overrightarrow{Opp}_i = (\overrightarrow{ub}_i + \overrightarrow{lb}_i) - \overrightarrow{X}_i \tag{14}$$

3.5. Random Opposition-Based Learning (ROBL)

In the second approach, ROBL [123] is applied to enhance the exploitation ability of a search mechanism and improve the convergence speed. Different from the original OBL, this paper utilizes this improved OBL strategy [123], which is defined using Equation (15):

$$\hat{x}_j = l_j + u_j - \text{rand} \times x_j, \quad j = 1, 2, \dots, n \tag{15}$$

where x_j is the opposite solution, l_j and u_j are the lower and upper bounds of the problem in the j th dimension, and rand is a random number within (0, 1).

3.6. AHA-ROBL- and AHA-OBL-Based FS for Waste Classification

In this study, we used two improved versions of the AHA to select the most optimal features based on ROBL and OBL based on the kNN classifier for selecting an optimal set of features. To improve the exploitation phase of the original AHA method and avoid a convergence to local minima, we developed the two new approaches. The first approach,

the AHA-ROBL, incorporates ROBL. The second approach, the AHA-OBL, incorporates OBL. These operators ensure a more balanced approach to exploration and exploitation. The incorporation of OBL and ROBL with the AHA provides a good solution to escape from local optima. The design of waste classification based on the AHA-ROBL and AHA-OBL is depicted in Figure 1, and it contains five basic processes, which are detailed as follows:

1. Pre-processing data

This stage consists of loading the TrashNet dataset, which is divided into k -folds. All images must be resized to $224 \times 224 \times 3$ for ResNet and VGG19.

2. Deep feature extraction

In this stage, two pre-trained CNNs are used to extract trainable features, which are more efficient than other descriptors. AlexNet extracts 4096 features while ResNet extracts 2048.

3. Initialization

As is the case for the majority of computational algorithms, the AHA begins by generating an initial population of N objects; each object has the dimension Dim in the search space that is constrained by the higher and lower bounds of a population and the maximum number of iterations, as defined by Equation (2). The process of FS requires converting the real values into binary using a sigmoidal function, defined by the following equations:

$$x_i^{It+1} = \begin{cases} 0, & \text{if } rand < Sig(x_i^{It}) \\ 1, & \text{if } rand \geq Sig(x_i^{It}) \end{cases} \quad (16)$$

where

$$Sig(x_i^{It}) = \frac{1}{1 + e^{(-10*(x_i^{It}-0.5))}}. \quad (17)$$

Any solution is represented as a one-dimensional vector; the number of deep features specifies the length. Any cell may have one of two values, 0 or 1, where 1 indicates that the appropriate feature has been selected and 0 indicates that it has not been selected.

4. Score evaluation

Generally, the feature selection seeks to decrease the number of features and the classification error rate. In other words, classification accuracy is maximized by deleting superfluous and redundant traits and maintaining only the most pertinent ones. The k NN classifier was used in this investigation due to its ease in evaluating the score. Thus, the score for each object was evaluated by using the following:

$$Sc = 0.99 \times (1 - Cr) + 0.01 \times \frac{|Sel_f|}{|Tot_f|} \quad (18)$$

where Cr and Sel_f are the accuracy obtained by using k NN ($k = 5$) and the size of selected deep features, respectively. Tot_f is the total number of trainable features provided by AlexNet/ResNet.

5. Updating process

First, the AHA seeks to update the guided foraging by using the three flight skills, namely omnidirectional, diagonal, and axial flight, using Equations (4)–(6), respectively. In case of $r \leq 1/3$, follow diagonal flight using Equation (5). In case of $r \leq 2/3$, follow omnidirectional flight using Equation (4); otherwise, follow axial flight using Equation (6).

Second, the updating of objects is realized by using the exploration mode (when $r \leq 0.5$), which applies the adjustment of acceleration using Equation (7). Otherwise, follow territorial foraging using Equation (9) (exploitation operation). The migration foraging is applied when $t_p = 2n$ by using Equation (10).

The exploitation mode is realized by the integration of ROBL or OBL, which ensures a good balance between the exploration and exploitation modes using Equations (14) and

(15). This integration deeply enhances the convergence to the global solution. The third step consists of evaluating the score for each object using Equation (18) to find the best candidate. The evaluating and updating stages are repeated indefinitely until a termination condition is satisfied. This condition is utilized in this study to determine the quality of the suggested approach for locating the optimal subset of features within the given number of iterations.

3.7. The Computational Complexity of the Two Proposed Algorithms (the AHA-OBL and AHA-ROBL)

This section explains the time and space costs maintained by the proposed methods.

3.7.1. Time Complexity

First, the two proposed approaches (the AHA-OBL and AHA-ROBL) produce N number of search agents, each with size D so that the initialization complexity can be represented as the $\mathcal{O}(N \times D)$ time complexity. Moreover, the AHA-OBL and AHA-ROBL calculate the fitness of each search agent with the complexity of $\mathcal{O}(t_{max} \times N \times D)$, where t_{max} indicates the cumulative number of iterations. In addition, the AHA-OBL and AHA-ROBL need the $\mathcal{O}(T)$ time complexity to perform T number of its main operations (phase 1, phase 2, and phase 3; memory saving; and OBL or ROBL). Hence, the total time complexity of the two proposed approaches (the AHA-OBL and AHA-ROBL) can be represented by $\mathcal{O}(t_{max} \times T \times N \times D)$.

3.7.2. Space Complexity

Space complexity determines the total amount of space occupied by the two proposed algorithms. The AHA-OBL and AHA-ROBL use the space complexity of $\mathcal{O}(N \times D)$.

4. Experimental Results

In order to conduct a fair analysis, the effectiveness of the AHA-ROBL and AHA-OBL was compared to that of different and recent computational algorithms, namely AHA, HHO, SSA, AO, HGSO, PSO, GWO, AOA, MRFO, SCA, MPA, and SAR. The performance was tested on the TrashNet dataset under identical conditions utilizing two deep descriptors, namely VGG19 and ResNet20. In this section, the comparison between the results of the two developed FS approaches and the other 12 methods is performed. Overall, 90% of the dataset was used for training the classification algorithm and 10% of the dataset was used for the validation. As a classification algorithm, kNN was used.

In this study, we set the maximum number of iterations to 200. Due to the stochastic nature of the computational algorithms, each algorithm was run 30 times separately. The computer's CPU was an Intel Core i7-5500U processor running at 2.40 GHz, and the RAM was 32 GB.

4.1. Parameter Settings for the Comparative Algorithms

This section defines the parameters for each optimizer. To ensure a fair comparison, it is necessary to list the waste recognition algorithms that were implemented. The suggested two methods (the AHA-ROBL and AHA-ROBL) and other 12 computational algorithms are specified in Table 1.

Table 1. Parameter settings of the AHA-ROBL and other computational algorithms.

Algorithm	Parameter Settings
	Common Settings Max no. of iterations: ($It_{Max} = 200$) Number of runs: 30 Size of population: $N = 30$ Problem dimensions: Dim = 3 and $B = 0.1$ Social effect parameter: $SE = 0.05$ Maximal limit: Fixed to 1 Minimal limit: Fixed to 0
AHA [70]	$r = rand, r$ range $[0, 1]$, migration coefficient = $2n$
HHO [124]	$\beta = 1.5$ (default)
SSA [61]	$c_1 = 2 * exp(-4t/It_{Max})^2, c_2 = rand,$ and $c_3 = rand$
AO [125]	$U = 0.00565; r1 = 10; \omega = 0.005; \alpha = 0.1; \delta = 0.1;$ $G1 \in [-1, 1];$ and $G2 = [2, 0]$
HGSO [126]	Clusters number = 2, $M_1 = 0.1, M_2 = 0.2, \alpha = \beta = 1, K = 1,$ $l_1 = 5 \times 10^{-3}, l_2 = 1 \times 10^2,$ and $l_3 = 1 \times 10^{-2}$
PSO [127]	$W_{max} = 0.9, W_{min} = 0.2,$ $c_1 = 2,$ and $c_2 = 2$
GWO [128]	$a \in [2, 0]$
AOA [129]	$c_1 = 2, c_2 = 6,$ $\lambda = 0.9,$ and $\mu = 0.1$
MRFO [130]	$b = 1$ and a decreases linearly from -1 to -2 (default) Maximum count of iterations: 100
SCA [76]	$P_a \in [0, 1]$
MPA [129]	FADs = 0.2 and $P = 0.5$
SAR [131]	$MU = 2 \times D$ for infeasible solutions and $MU = 30 \times D$ for feasible solutions

4.2. Performance Metrics

The following evaluation metrics and measurements were computed for the proposed method (the AHA-ROBL) developed for waste-analysis-based FS. Consequently, the metrics' correct rate of waste classification involves mean accuracy (μ_{Acc}), recall (Re), precision (Pr), F-score (F_{sc}), score, sensitivity, specificity, average execution time, and selection ratio. Consequently, all metrics are expressed in terms of the mean and standard deviation, which are characterized by using the following:

- Mean accuracy (μ_{Acc}): The μ_{Acc} metric is calculated as Equation (19):

$$\mu_{Acc} = \frac{1}{M} \frac{1}{N_s} \sum_{k=1}^M \sum_{r=1}^{N_s} (C_r == L_r) \tag{19}$$

where M represents the number of runs, N_s represents the number of samples in the test dataset, and C_r and L_r represent the classifier output label and the reference label class of sample r , respectively.

- Mean fitness value (μ_{Fit}): The fitness value metric, which evaluates the performance of algorithms, is expressed as in Equation (20):

$$\mu_{Fit} = \frac{1}{M} \sum_{k=1}^M Fit_*^k \tag{20}$$

where M is the number of runs and Fit_*^k is the best fitness value for the k^{th} run.

- Average recall (μ_{Re}): This indicates the percentage of predicted positive patterns that is defined as in Equation (21):

$$Re = \frac{Trp}{Trp + Fan} \tag{21}$$

The μ_{Re} is calculated from the best object (O_{best}) using Equation (22):

$$\mu_{Re} = \frac{1}{30} \sum_{r=1}^{30} Re_r^{best} \tag{22}$$

- Average precision (μ_{Pr}): This indicates the frequency of true expected samples as in Equation (23):

$$Pr = \frac{Trp}{Fap + Trp} \tag{23}$$

The mean precision (μ_{Pr}) can be calculated by using Equation (24):

$$\mu_{Pr} = \frac{1}{30} \sum_{r=1}^{30} Pr_r^{best} \tag{24}$$

- Mean F-score ($\mu_{F_{Score}}$): This metric is already in use for balanced data, which can be calculated using Equation (25):

$$F_{Score} = 2 \times \frac{Re \times Pr}{Re + Pr} \tag{25}$$

The mean F-score can be calculated using Equation (26):

$$\mu_{F_{Score}} = \frac{1}{30} \sum_{r=1}^{30} F_r^{Score-best} \tag{26}$$

- Mean features selection size (μ_{Size}): This indicates the average size of the selected attributes and is expressed as in Equation (27):

$$\mu_{Size} = \frac{1}{M} \sum_{k=1}^M d_*^k \tag{27}$$

4.3. Results and Discussion

- **Fitness:** Table 2 displays the results of comparing the two proposed models (the AHA-ROBL and AHA-OBL) and competing algorithms. Based on the obtained results, it is evident that our AHA-ROBL model provides superior results, followed by the AHA-OBL. Two pre-trained CNN models (VGG19 and ResNet20) and the TrashNet dataset were chosen. The deep analysis of the dataset that was used revealed that the quantitative results obtained by using the proposed AHA-ROBL approach performed better with the two pre-trained CNN models (VGG19 and ResNet20) than the optimization algorithms, namely the basic AHA, HHO, SSA, AO, HGS, PSO, GWO, AOA, MRFO, SCA, MPA, and SAR. The results of the proposed VGG19 method are significantly superior to those of ResNet20. The AHA-OBL followed this with the lowest fitness value. The standard deviation was computed to evaluate the stability of the fitness value for each FS method. According to the standard deviation results, the

traditional AHA-ROBL, AHA, and PSO approaches are more stable than other algorithms. HGS is the worst possible algorithm. It is important to note that the AHA-OBL obtained the second-best position using VGG19. In addition, for the ResNet20 deep features, the AHA-OBL was ranked second compared to the remaining 12 algorithms.

Table 2. Comparison of the performance of deep features models of the AHA-ROBL with other recent optimizers versus avg. fitness.

Fitness Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.0129	0.0219	0.0045	0.0533	0.0720	0.0085
AHA-OBL	0.0148	0.0257	0.007707	0.0643	0.0842	0.01407
AHA	0.0250	0.0391	0.0053	0.0972	0.1120	0.0072
HHO	0.0291	0.0459	0.0051	0.1089	0.1280	0.0065
SSA	0.0475	0.0532	0.0033	0.1067	0.1246	0.0090
AO	0.0401	0.0485	0.0053	0.0968	0.1267	0.0109
HGS	0.0517	0.0612	0.0045	0.1222	0.1407	0.0080
PSO	0.0476	0.0508	0.0020	0.1104	0.1241	0.0043
GWO	0.0522	0.0590	0.0031	0.1288	0.1404	0.0065
AOA	0.0478	0.0568	0.0041	0.1264	0.1393	0.0070
MRFO	0.04079	0.0322	0.0047	0.1142	0.1002	0.0088
SCA	0.00195	0.0553	0.0475	0.0043	0.1302	0.1103
MPA	0.0378	0.0300	0.0049	0.1051	0.0935	0.0082
SAR	0.0513	0.0457	0.0025	0.1220	0.1155	0.0040

- Accuracy:** The following observations can be drawn from the data presented in Table 3. First, the results demonstrated that the two proposed approaches (the AHA-ROBL and AHA-OBL) outperformed the optimization algorithms, namely the basic AHA, HHO, SSA, AO, HGS, PSO, GWO, AOA, MRFO, SCA, MPA, and SAR, in terms of quantitative results using the two pre-trained CNN models (VGG19 and ResNet20). The results of the proposed VGG19 method were significantly superior to those of ResNet. Compared to the 12 optimization algorithms, the MPA achieved the highest accuracy value.

Table 3. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus accuracy.

Accuracy Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.9881	0.9791	0.0045	0.9486	0.9295	0.0085
AHA-OBL	0.9860	0.9770	0.0007	0.9369	0.9180	0.0133
AHA	0.9763	0.9626	0.0052	0.9051	0.8909	0.0070
HHO	0.9723	0.9559	0.0049	0.8933	0.8750	0.0064
SSA	0.9565	0.9510	0.0034	0.8972	0.8791	0.0091
AO	0.9605	0.9536	0.0050	0.9051	0.8755	0.0109
HGS	0.9526	0.9430	0.0045	0.8814	0.8627	0.0080
PSO	0.9565	0.9535	0.0020	0.8933	0.8796	0.0044
GWO	0.9526	0.9445	0.0035	0.8775	0.8631	0.0072
AOA	0.9565	0.9474	0.0042	0.8775	0.8655	0.0064
MRFO	0.9762	0.9695	0.0047	0.9288	0.9025	0.0087
SCA	0.9684	0.9591	0.0045	0.8932	0.8749	0.0078
MPA	0.9802	0.9710	0.0049	0.9288	0.9080	0.0079
SAR	0.9644	0.9582	0.0026	0.8972	0.8880	0.0041

- Recall and precision:** Tables 4 and 5 list the recall and precision of the two proposed methods (the AHA-ROBL and AHA-OBL) with the 12 wrapper FS algorithms employing the two deep descriptors (VGG19 and ResNet20). By examining the average recall

and precision values for the TrashNet dataset, it is evident that the AHA-ROBL outperformed all advanced competitor algorithms based on both deep features (VGG19 and ResNet20). Moreover, the average recall and precision obtained by using the AHA-ROBL based on VGG19 were superior to those obtained by the AHA-ROBL based on ResNet20. It can be seen that the AHA-ROBL based on deep descriptors has a strong stability for the TrashNet dataset due to the lower values of the standard deviation in terms of the precision and recall metrics. In addition, the AHA-OBL based on the deep VGG19 descriptor ranked in the second position in terms of average recall and precision for the TrashNet dataset. In addition, the MPA based on the deep VGG19 descriptor ranked in the third position in terms of average recall and precision for the TrashNet dataset. It can be seen that the AHA-ROBL based on the deep descriptors has a strong stability for the TrashNet dataset due to the lower values of the standard deviation in terms of the precision and recall metrics.

Table 4. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus avg. recall.

Recall Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.9861	0.9774	0.0048	0.9418	0.9022	0.0172
AHA-OBL	0.9853	0.9744	0.0077	0.9365	0.8980	0.0272
AHA	0.9740	0.9595	0.0068	0.8866	0.8588	0.0128
HHO	0.9703	0.9506	0.0081	0.8774	0.8402	0.0150
SSA	0.9546	0.9440	0.0062	0.8682	0.8440	0.0140
AO	0.9613	0.9471	0.0080	0.8623	0.8362	0.0161
HGS	0.9469	0.9344	0.0061	0.8528	0.8247	0.0142
PSO	0.9548	0.9470	0.0039	0.8716	0.8444	0.0126
GWO	0.9447	0.9352	0.0061	0.8622	0.8261	0.0163
AOA	0.9564	0.9388	0.0065	0.8471	0.8228	0.0132
MRFO	0.9771	0.9674	0.0069	0.9229	0.8717	0.0171
SCA	0.9650	0.9533	0.0075	0.8609	0.8338	0.0136
MPA	0.9809	0.9683	0.0068	0.9226	0.8755	0.0194
SAR	0.9615	0.9533	0.0044	0.8797	0.8580	0.0125

Table 5. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus avg. precision.

Precision Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.9909	0.9756	0.0080	0.9520	0.9167	0.0155
AHA-OBL	0.9875	0.9732	0.0101	0.9430	0.9075	0.0251
AHA	0.9752	0.9544	0.0086	0.9008	0.8757	0.0125
HHO	0.9663	0.9478	0.0076	0.8733	0.8560	0.0087
SSA	0.9499	0.9393	0.0047	0.8970	0.8640	0.0147
AO	0.9591	0.9449	0.0070	0.8902	0.8563	0.0170
HGS	0.9453	0.9303	0.0061	0.8657	0.8439	0.0103
PSO	0.9476	0.9368	0.0052	0.8747	0.8502	0.0116
GWO	0.9445	0.9344	0.0041	0.8642	0.8476	0.0115
AOA	0.9458	0.9346	0.0054	0.8776	0.8471	0.0128
MRFO	0.9789	0.9630	0.0073	0.9195	0.8865	0.0156
SCA	0.9699	0.9544	0.0082	0.9045	0.8632	0.0168
MPA	0.9827	0.9660	0.0071	0.9219	0.8948	0.0140
SAR	0.9554	0.9468	0.0045	0.8876	0.8692	0.0096

- **Sensitivity and specificity:** Tables 6 and 7 list the sensitivity and specificity of the two proposed methods (the AHA-ROBL and AHA-OBL) with the 12 wrapper FS algorithms employing the two deep descriptors (VGG19 and ResNet20). By examining

the average sensitivity and specificity values for the TrashNet dataset, it is evident that the AHA-ROBL outperformed all advanced competitor algorithms based on both deep features (VGG19 and ResNet20). Moreover, the average sensitivity and precision obtained by using the AHA-ROBL based on VGG19 are superior to those obtained by using the AHA-ROBL based on ResNet20. It can be seen that the AHA-ROBL based on the deep descriptors has a strong stability for the TrashNet dataset due to the lower values of the standard deviation in terms of the sensitivity and specificity metrics. In addition, the AHA-OBL based on the deep VGG19 descriptor ranked in the second position in terms of the average sensitivity and specificity for the TrashNet dataset. Moreover, the AHA based on the deep VGG19 descriptor ranked in the second position in terms of the average sensitivity and specificity for the TrashNet dataset. It can be seen that the AHA-ROBL based on the deep descriptors has a strong stability for the TrashNet dataset due to the lower values of the standard deviation in terms of the sensitivity and specificity metrics.

Table 6. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus avg. sensitivity.

Sensitivity Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.9922	0.9879	0.0030	0.8874	0.8836	0.0026
AHA-OBL	0.9885	0.9869	0.0011	0.8980	0.8876	0.0073
AHA	0.9740	0.9595	0.0068	0.8866	0.8588	0.01285
HHO	0.9703	0.9506	0.0081	0.8774	0.8402	0.0150
SSA	0.9546	0.9440	0.0062	0.8682	0.8440	0.0140
AO	0.9613	0.9471	0.0080	0.8623	0.8362	0.01612
HGS	0.9469	0.9344	0.0061	0.8528	0.8247	0.0142
PSO	0.9548	0.9470	0.0039	0.8716	0.8444	0.01262
GWO	0.9445	0.9352	0.00612	0.8622	0.8261	0.0163
AOA	0.9564	0.9388	0.0065	0.8471	0.8228	0.0132
MRFO	0.9772	0.9675	0.0069	0.9230	0.8718	0.0171
SCA	0.9650	0.9534	0.0075	0.86093	0.8338	0.01365
MPA	0.9809	0.9684	0.0068	0.9226	0.8755	0.0195
SAR	0.9616	0.9533	0.0045	0.8797	0.8581	0.0126

Table 7. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus avg. specificity.

Specificity Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.9980	0.9971	0.0006	0.9934	0.9922	0.0008
AHA-OBL	0.9967	0.9963	0.0002	0.9882	0.9873	0.0006
AHA	0.9951	0.9925	0.0010	0.9805	0.9775	0.0015
HHO	0.9945	0.9912	0.0010	0.9782	0.9742	0.0014
SSA	0.9914	0.9903	0.0007	0.9788	0.9750	0.0019
AO	0.9922	0.9907	0.0010	0.9805	0.9744	0.0022
HGS	0.9907	0.9887	0.0009	0.9756	0.9717	0.0017
PSO	0.9915	0.9908	0.0004	0.9780	0.9752	0.0009
GWO	0.9906	0.9906	0.0008	0.9748	0.9717	0.0015
AOA	0.9915	0.9896	0.0008	0.9749	0.9722	0.0013
MRFO	0.9953	0.9939	0.00095	0.9854	0.9799	0.0018
SCA	0.9939	0.9917	0.0009	0.9780	0.9742	0.0016
MPA	0.9959	0.9942	0.0010	0.9854	0.98105	0.00166
SAR	0.9929	0.9917	0.0005	0.9788	0.9770	0.0009

- **F-score:** In terms of the F-score, Table 8 reveals that the two proposed methods (the AHA-ROBL and AHA-OBL) were based on the pre-trained CNNs (VGG19 and ResNet20) and outperformed all the other competitors. In addition, fierce competition existed between the MPAs based on ResNet20 and VGG19 for the third position. Moreover, the GWO based on the deep features achieved lower F-score values.

Table 8. Comparison of the performance of deep features models of the AHA-ROBL with other recent optimizers versus F-score.

F-score Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	0.9883	0.9759	0.0063	0.9418	0.9076	0.0153
AHA-OBL	0.9860	0.9720	0.0098	0.9375	0.8995	0.0268
AHA	0.9736	0.9555	0.0072	0.8846	0.8648	0.0118
HHO	0.9660	0.9477	0.0075	0.8708	0.8448	0.0110
SSA	0.9503	0.9397	0.0052	0.8756	0.8507	0.0130
AO	0.9582	0.9445	0.0069	0.8723	0.8426	0.0154
HGS	0.9451	0.9304	0.0059	0.8486	0.8306	0.0114
PSO	0.9481	0.9429	0.0029	0.8680	0.8495	0.0097
GWO	0.9426	0.9329	0.0044	0.8569	0.8330	0.0128
AOA	0.9474	0.9346	0.0055	0.8508	0.8310	0.0118
MRFO	0.9768	0.9642	0.0068	0.9193	0.8765	0.0150
SCA	0.9625	0.9526	0.0065	0.8619	0.8435	0.0125
MPA	0.9814	0.9662	0.0065	0.9192	0.8822	0.0158
SAR	0.9572	0.9484	0.0041	0.8771	0.8609	0.0097

- **Selection ratio:** According to the results of Table 9, which depict the mean rate of the selection ratio and its standard deviation, the AHA-ROBL exhibited excellent performance in selecting relevant deep features from the TrashNet dataset. In addition, we can observe that the proposed AHA-ROBL method provided an excellent behavior for selecting the optimal set of relevant deep features. The deep analysis of the dataset that was used revealed that the quantitative results obtained by using the proposed AHA-ROBL approach performed better with the two pre-trained CNN models (VGG19 and ResNet20) than the optimization algorithms, namely the basic AHA, HHO, SSA, AO, HGS, PSO, GWO, and AOA. Clearly, the proposed VGG19 approach produces significantly superior results to ResNet20. It is important to mention that the second-best place was obtained by the AHA using VGG19.
- **Average execution time:** Table 10 reveals that the two proposed methods (the AHA-ROBL and AHA-OBL) based on the pre-trained CNNs (VGG19 and ResNet20) outperformed 75% of the other competitors. In addition, the AHA outperformed most of the other competitors.

Table 9. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus number of selected features.

Number of Selected Features Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	94.0000	117.9000	18.7458	138.0000	224.9000	40.8034
AHA-OBL	73.0000	237.9000	90.5882	218.0000	385.5667	79.8156
AHA	80.0000	210.8333	70.7487	294.0000	404.1333	61.8952
HHO	99.0000	223.3667	70.9844	193.0000	425.5667	126.6580
SSA	439.0000	464.2000	18.2575	449.0000	482.7667	14.1998
AO	98.0000	260.5333	98.0597	232.0000	339.4667	59.3498
HGS	422.0000	469.4333	21.9430	437.0000	475.6000	19.5141
PSO	445.0000	471.6333	16.5143	455.0000	488.9000	16.1295
GWO	222.0000	413.6667	121.3977	273.0000	491.7333	119.3106
AOA	442.0000	477.2333	18.8634	463.0000	609.7333	128.2274
MRFO	89.0000	210.2333	61.4142	250.0000	376.8333	71.0110
SCA	36	69.6	27.80	86	135.3	46.81
MPA	78.0000	135.7000	41.3139	105.0000	251.0333	95.6839
SAR	411.0000	437.2000	16.0675	440.0000	469.2667	18.1278

Table 10. Comparison of the performance of deep feature models of the AHA-ROBL with other recent optimizers versus avg. time.

Time Algorithm	VGG19			ResNet		
	Best	Mean	STD	Best	Mean	STD
AHA-ROBL	235.5352	342.9767	66.5081	364.7488	475.6399	58.0576
AHA-OBL	179.1031	335.8094	75.5996	297.8248	428.1630	59.2680
AHA	175.2588	313.5663	55.8632	342.3833	434.8414447	47.3143
HHO	339.1078	536.8270	121.4785	477.5016	902.9543	242.9377
SSA	465.2132	487.6391	16.6067	474.1626	500.4554	12.7761
AO	518.7569	601.9692	39.6790	531.2182	588.2628	30.3227
HGS	5.4335	12.7695	4.9629	5.4464	13.0864	6.9104
PSO	488.9208	499.3343	5.5008	492.8176	506.3241	5.2705
GWO	401.2708	579.5520	90.2559	429.0840	640.4219	100.0631
AOA	855.3195	915.3140	27.9433	906.2781	955.8382	25.4345
MRFO	519.2461	687.2089	70.7906	669.6235	842.0057	70.53934
SCA	70.3634	100.0193	29.2336	103.6779	153.3768	38.5945
MPA	303.4597	372.0496	52.9391	395.3910	564.2305	100.2369
SAR	463.6103	478.8563	6.7850	482.5578	499.3555	7.5595

4.4. The Wilcoxon Test

A statistical analysis was necessary to compare the efficiency of the AHA-ROBL and AHA-OBL to the efficiency of other competitive algorithms. Thus, the Wilcoxon rank sum test was used to compare the accuracy values acquired by using the two proposed approaches (the AHA-ROBL and AHA-OBL) and those obtained by using the other algorithms, namely the basic AHA, HHO, SSA, AO, HGSO, PSO, GWO, AOA, MRFO, SCA, MPA, and SAR for the TrashNet dataset in the cases of the VGG19 and ResNet20 deep descriptors. Table 11 contains the results of the Wilcoxon signed-rank test, which was used to evaluate the statistical performance differences between the two proposed algorithms and the other 12 algorithms. A *p*-value of less than 0.05 indicated a statistically significant difference between the two compared algorithms. Following this criterion, the AHA-ROBL outperformed all the other algorithms to varying degrees, indicating that the AHA-ROBL benefits from extensive exploitation. In general, the AHA-ROBL based on the deep descriptor VGG19 had a statistically significant *p*-value in comparison with 85.7% of the algorithms.

Table 11. Wilcoxon rank sum statistical test.

AHA-ROBL vs.	TrashNet Dataset	
	VGG19	ResNet
AHA	7.44×10^{-11}	2.65×10^{-11}
HHO	2.44×10^{-11}	2.39×10^{-11}
SSA	1.76×10^{-11}	2.63×10^{-11}
AO	2.12×10^{-11}	2.64×10^{-11}
HGS	2.04×10^{-11}	2.60×10^{-11}
PSO	8.45×10^{-11}	2.21×10^{-11}
GWO	1.75×10^{-11}	2.60×10^{-11}
AOA	2.02×10^{-11}	2.60×10^{-11}
MRFO	3.04×10^{-11}	3.720×10^{-11}
SCA	3.852×10^{-11}	3.942×10^{-11}
MPA	2.52×10^{-11}	2.73×10^{-11}
SAR	2.08×10^{-11}	2.70×10^{-11}

4.5. Graphical Analysis

Figure 6 depicts the fitness curves derived by using the various optimizers based on VGG19 and ResNet20 for the TrashNet dataset. By analyzing the behavior of the convergence of the two proposed algorithms (the AHA-ROBL and AHA-OBL) for the TrashNet dataset based on the VGG19 deep descriptor, a speed convergence was illustrated by increasing the number of iterations compared to the other 12 algorithms.

For the TrashNet dataset, we can see that the AHA-OBL and the conventional AHA based on the VGG19 descriptor highlighted a great competition in the first iterations. Still, after 20 iterations, the AHA-ROBL and AHA-OBL became more efficient. This behavior can be interpreted through the use of operators, which allows for deeply enhancing the exploitation process.

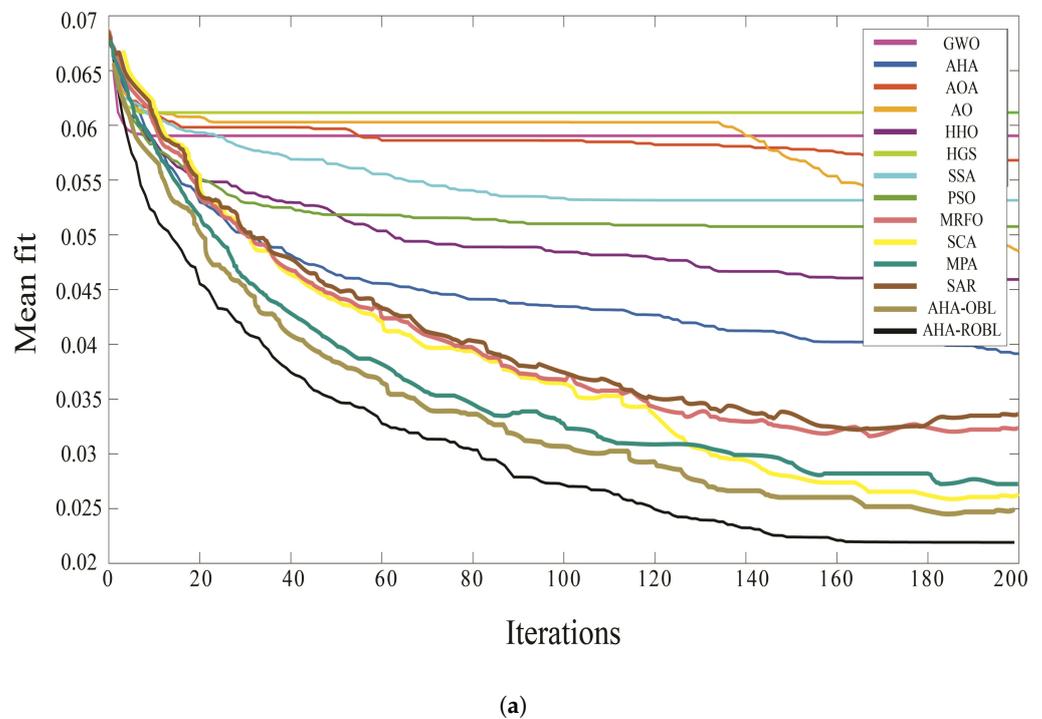


Figure 6. Cont.

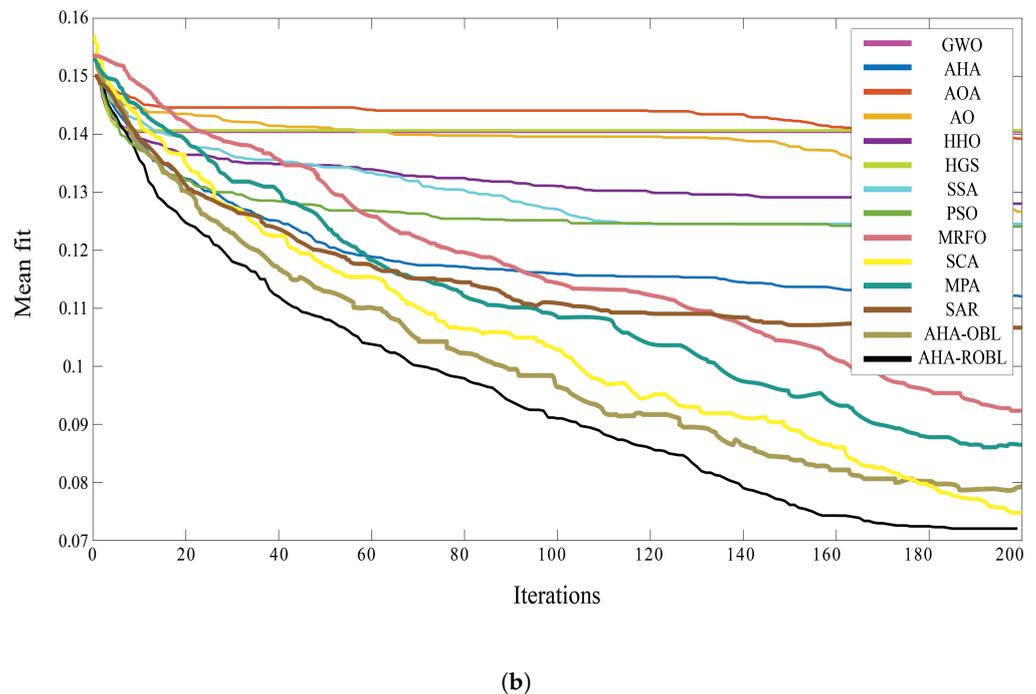


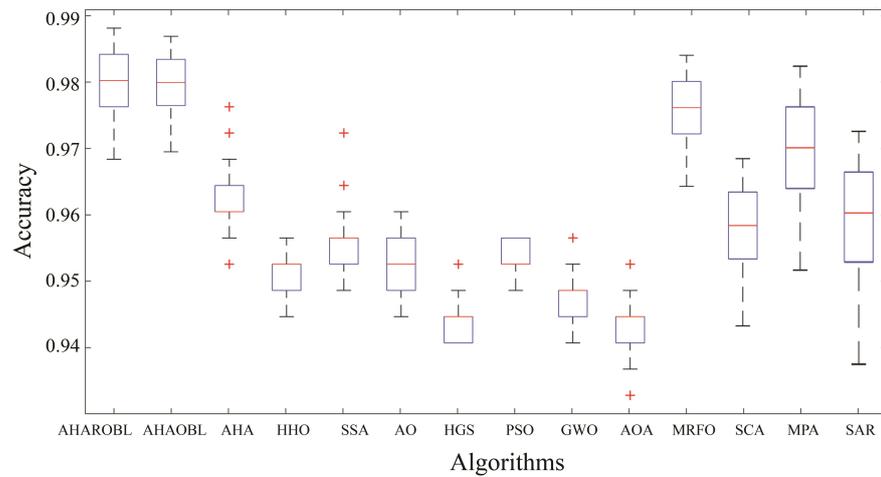
Figure 6. (a) Convergence curve of the AHA-ROBL and AHA-OBL versus other algorithms using VGG19 and (b) convergence curve of the AHA-ROBL and AHA-OBL versus other algorithms using RESNET20.

Additionally, as shown in Figure 7, we plotted a boxplot of the two proposed methods (the AHA-ROBL and AHA-OBL) against the 12 other algorithms, namely the conventional AHA, HHO, SSA, AO, HGS, PSO, GWO, AOA, MRFO, SCA, MPA and SAR, in terms of accuracy. As illustrated in the figure, the two suggested methods, the AHA-ROBL and AHA-OBL, based on the deep features achieved greater mean and median accuracy values than the other advanced algorithms for the TrashNet dataset. The collected results demonstrate the proposed methods' efficacy in maintaining the highest classification accuracy, especially for the VGG19 deep features.

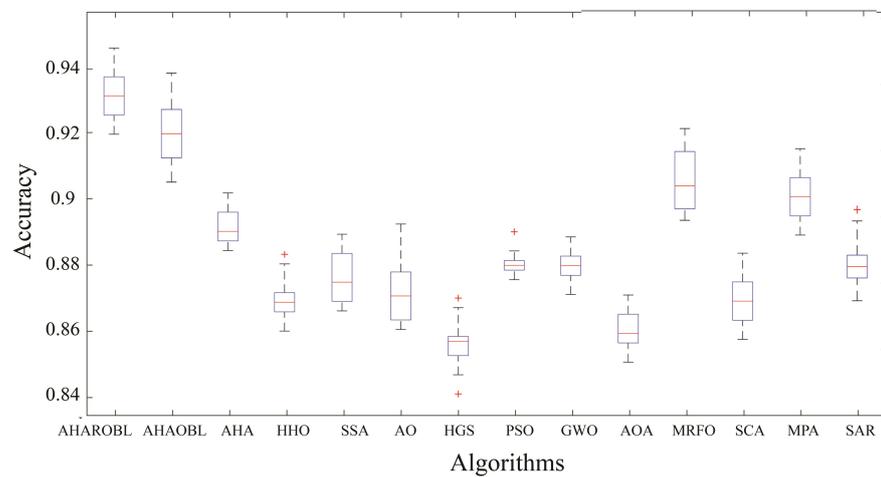
To summarize the results, Figures 8–11 display the mean values for accuracy, fitness, precision, recall, F-score, sensitivity, specificity, and average execution time for the two proposed approaches (the AHA-ROBL and AHA-OBL) based on the pre-trained CNNs (VGG19 and ResNet20) and various computational methods, namely the AHA, HHO, SSA, AO, HGS, PSO, GWO, AOA, MRFO, SCA, MPA, and SAR, for the TrashNet dataset. The results indicate that the two suggested approaches, the AHA-ROBL and AHA-OBL, have a superior performance and outperform all the competitors. As shown in Figures 8 and 9, the AHA-ROBL and AHA-OBL approaches based on the deep features produced higher mean and median accuracy, recall, precision, and fitness values than the other advanced algorithms for the TrashNet dataset. Moreover, as shown in Figures 10 and 11, the AHA-ROBL and AHA-OBL approaches based on the deep features produced higher mean sensitivity and specificity values and higher average execution times than the other advanced algorithms for the TrashNet dataset.

In terms of average accuracy, Figure 8 shows that the two proposed approaches, the AHA-ROBL and AHA-OBL, outperformed the 12 other optimization techniques, namely the basic AHA, HHO, SSA, AO, HGS, PSO, GWO, AOA, MRFO, SCA, MPA, and SAR, utilizing the two pre-trained CNN models (VGG19 and ResNet20).

In terms of average fitness, in Figure 8, the results indicate that the two proposed approaches, the AHA-ROBL and AHA-OBL, have a superior performance and outperform all the competitors.



(a)



(b)

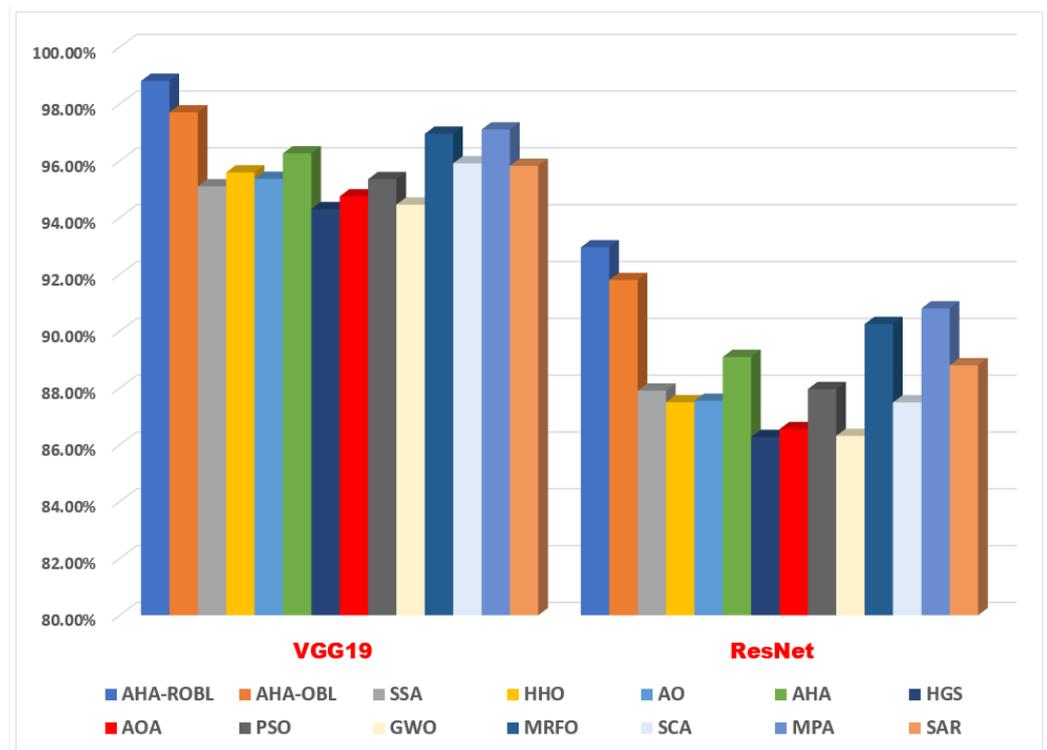
Figure 7. (a) Boxplot of the AHA-ROBL and AHA-OBL approaches versus other swarm intelligence algorithms using VGG19 and (b) boxplot of the AHA-ROBL and AHA-OBL approaches versus other swarm intelligence algorithms using ResNet20.

Regarding the average precision and recall, Figure 9’s values for the TrashNet dataset show that it is clear that the two proposed approaches, the AHA-ROBL and AHA-OBL, exceed all advanced rival techniques based on both the deep features and the average recall and precision values (VGG19 and ResNet20). Additionally, the average recall and precision values obtained by using the two proposed approaches using VGG19 are superior to those obtained using ResNet20.

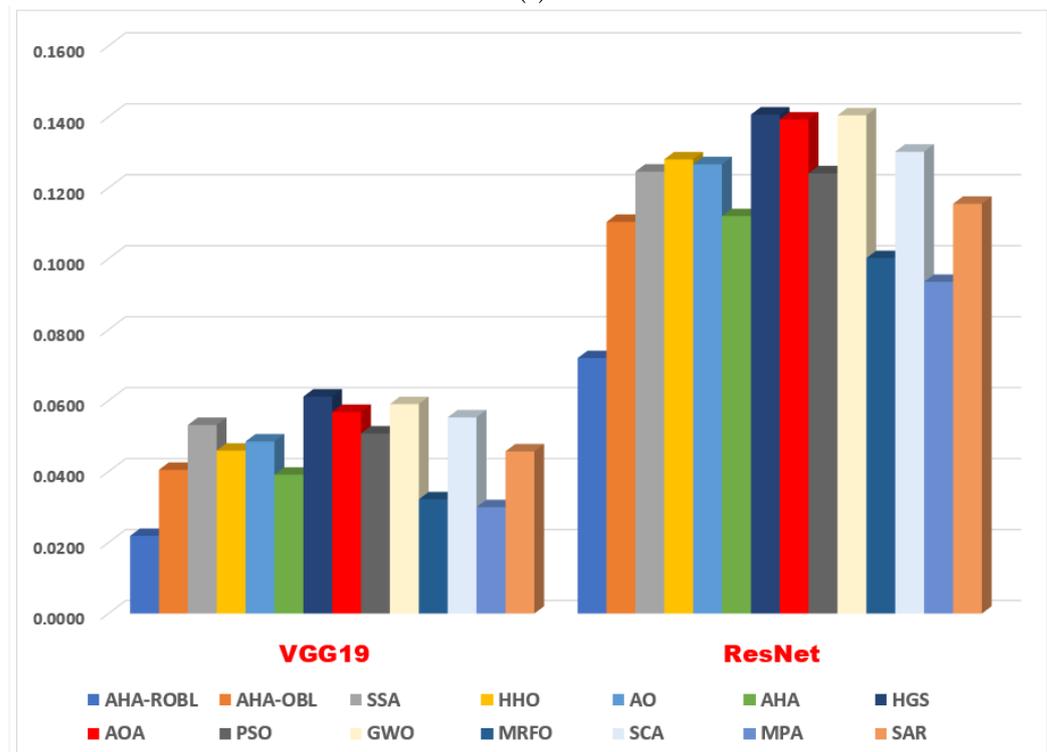
In terms of the average F-score, Figure 10 demonstrates that the two suggested techniques, the AHA-ROBL and AHA-OBL, based on the pre-trained CNNs (VGG19 and ResNet20) beat all the other alternatives.

In terms of average sensitivity and specificity, in Figures 10 and 11, the results indicate that the two proposed approaches, the AHA-ROBL and AHA-OBL, have a superior performance and outperform all the competitors.

In terms of average execution time, in Figure 11, the results indicate that the two suggested approaches, the AHA-ROBL and AHA-OBL, have a superior performance and outperform 75% of the other competitors.

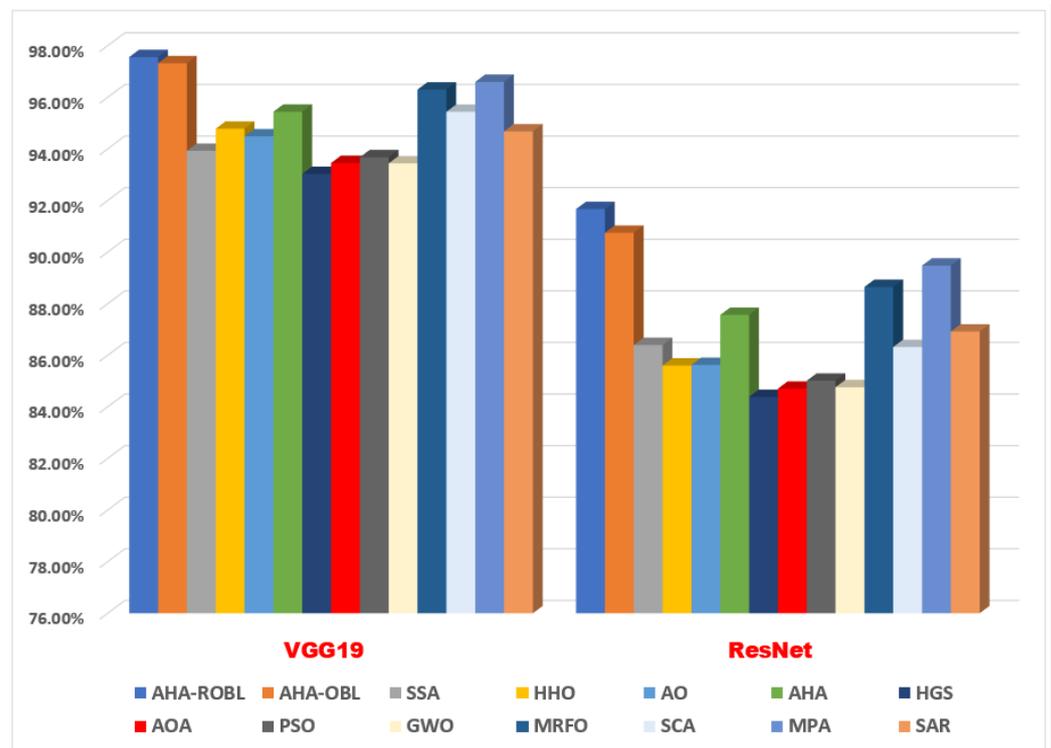


(a)

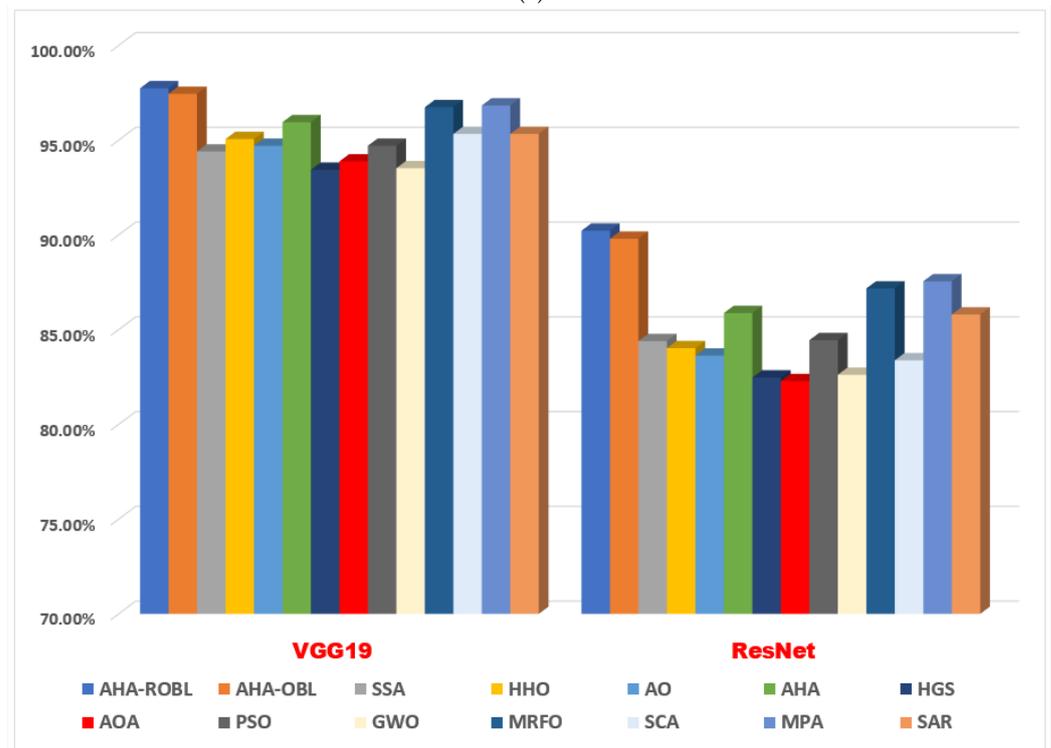


(b)

Figure 8. (a) Avg. accuracy of the proposed approaches versus other algorithms using VGG19 and ResNet20 and (b) avg. fitness of the proposed approaches versus other algorithms using VGG19 and ResNet20.

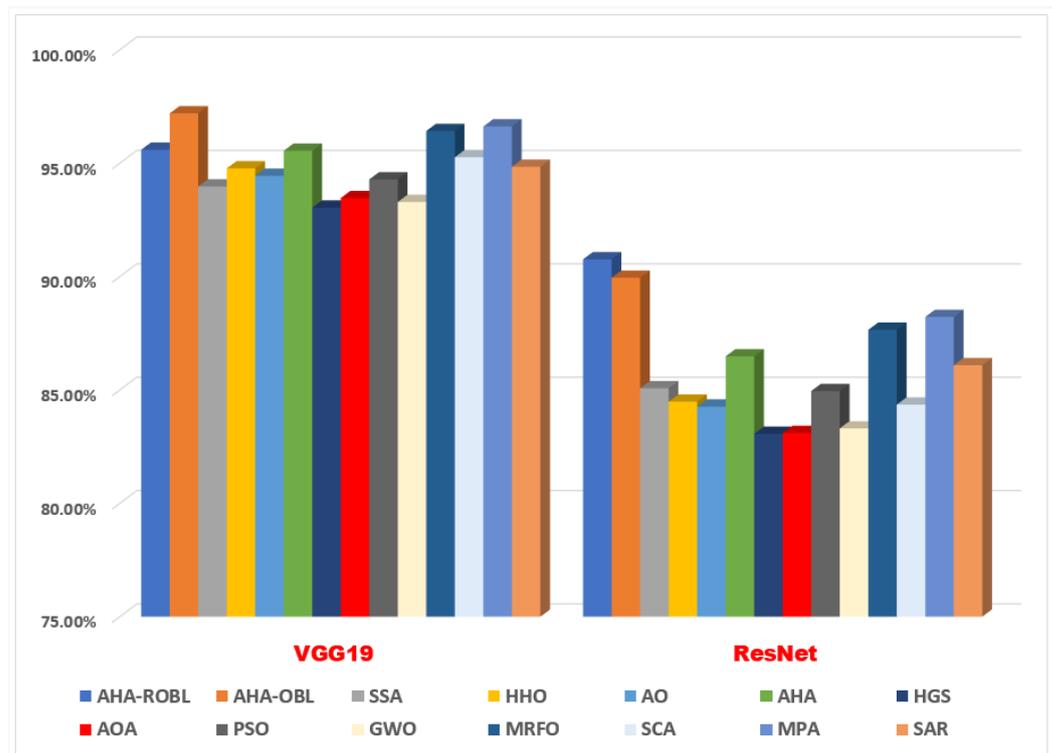


(a)

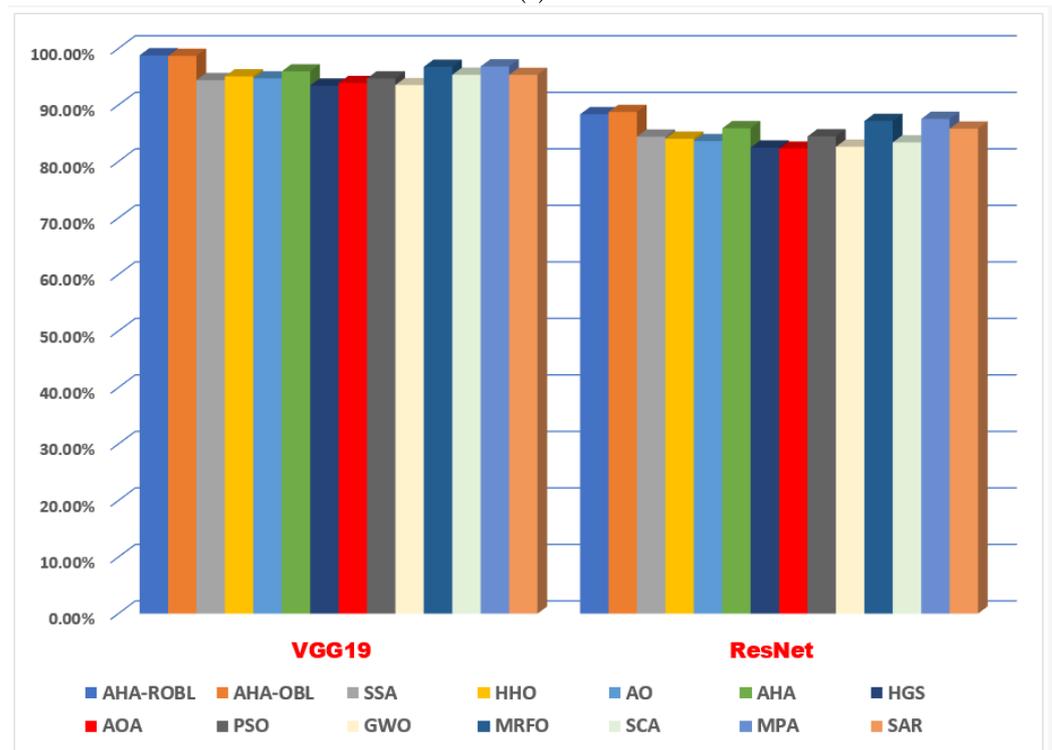


(b)

Figure 9. (a) Avg. precision of the proposed approaches versus other algorithms using VGG19 and ResNet20 and (b) avg. recall of the proposed approaches versus other algorithms using VGG19 and ResNet20.

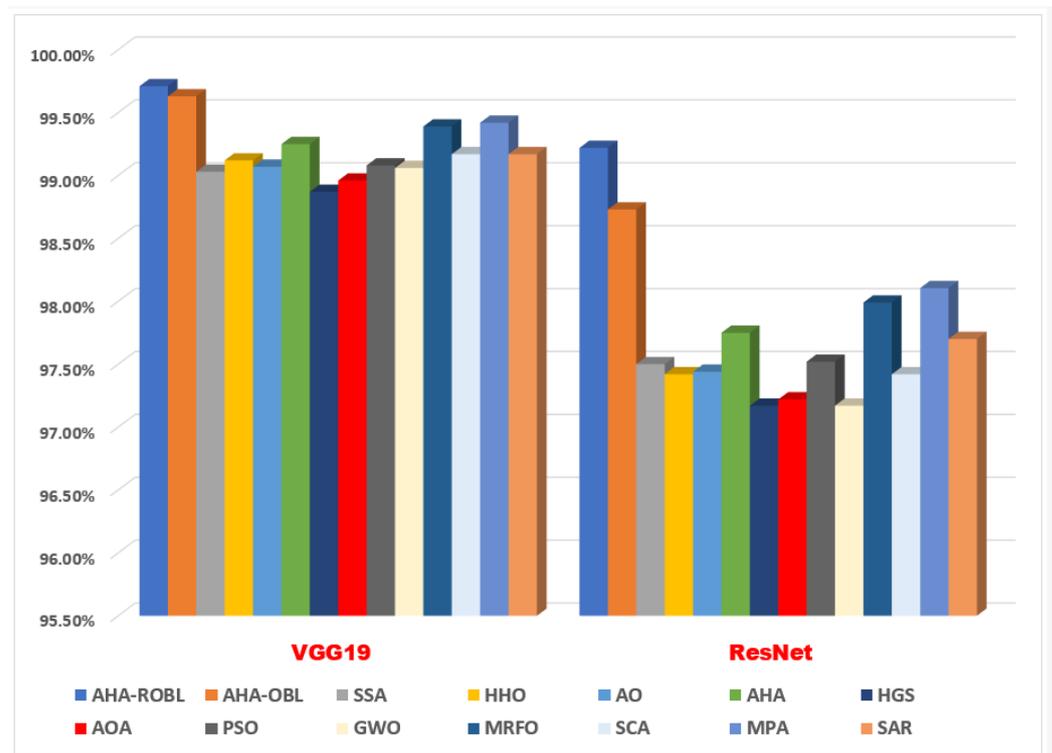


(a)

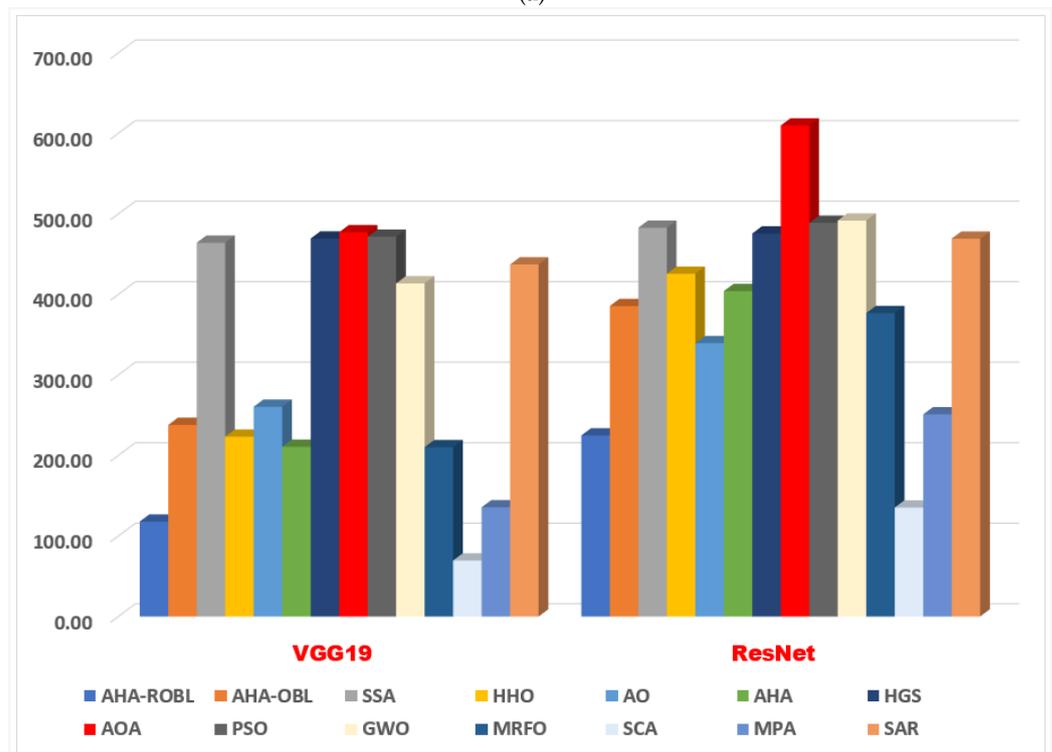


(b)

Figure 10. (a) Avg. F-score of the proposed approaches versus other algorithms using VGG19 and ResNet20 and (b) avg. sensitivity of the proposed approaches versus other algorithms using VGG19 and ResNet20.



(a)



(b)

Figure 11. (a) Avg. specificity of the proposed approaches versus other algorithms using VGG19 and ResNet20 and (b) avg. execution time of the proposed approaches versus other algorithms using VGG19 and ResNet20.

5. Comparative Study with the Existing Works

Previous studies on waste classification focused on applying different traditional/non-traditional mining techniques. This section summarizes all the previous studies' results for the TrashNet dataset (from 2016 to 2022). In order to demonstrate the efficiency of the suggested techniques (the AHA-ROBL and AHA-OBL), numerous algorithms from the literature were chosen for a fair comparison, including machine-learning and deep-learning algorithms. Table 12 illustrates the proper rate of classification's performance across the TrashNet dataset.

Many state-of-the-art algorithms used pre-trained networks [8,102,107] or fine-tuned pre-trained networks [102,107,132,133]. However, their methods did not achieve a good performance with the classification problem. In our research, we focused more on how to improve the performance of these pre-trained networks by using modified optimization techniques. However, our two proposed methods, the AHA-ROBL and AHA-OBL, which depend on using pre-trained networks (i.e., VGG19 and ResNet) and apply some feature selection methods, achieved higher results than WasNet method. The proposed methods' results reached 98.81% and 98.60%, respectively.

Table 12. A comparative study of the AHA-ROBL based on pre-trained CNNs with existing algorithms for TrashNet dataset.

Article	Methodology	Accuracy
[134]	Self-monitoring ResNet module	95.80%
[135]	WasNet	96%
[136]	Mb Xception	94.34%
[8]	Inception ResNet	88.60%
[107]	Fined-tuned DenseNet121	95%
[107]	DenseNet121	89%
[107]	Inception-ResNet V2	89%
[102]	Fine-tuned VGG16	93%
[102]	Fine-tuned AlexNet	91%
[102]	ResNet	88.66%
[137]	Inception-ResNet	88.34%
[8]	Inception	87.71%
[133]	Fine-tuned OscarNet	88.42%
[102]	Modified kNN	88%
[132]	Fine-tuned MobileNet	87.20%
[103]	Modified XGB	70.10%
[103]	Modified RF	62%
[117]	Modified SVM	63%
Our Proposed Model	AHA-ROBL(VGG19 and ResNet)	98.81%
	AHA-OBL (VGG19 and ResNet)	98.60%

6. Conclusions and Future Work

Waste classification has been a difficult task overall. However, the high number of attributes produced by pre-trained CNNs prompted us to integrate meta-heuristics to select the optimal set of deep-learning attributes. The majority of meta-heuristics suffer from the problem of exploitation. We solved this problem at the level of the AHA algorithm by using ROBL and OBL and applying them to waste classification using the two pre-trained CNN networks VGG19 and ResNet20. By analyzing the obtained results, we noted that the proposed AHA-ROBL and AHA-OBL algorithms manage to improve the performance of waste classification and are more competitive than other algorithms, namely the AHA, HHO, SSA, AO, HGSO, PSO, GWO, AOA, MRFO, SCA, MPA and SAR, in terms of accuracy, recall, precision, fitness, F-score, and statistical tests for the TrashNet dataset.

In the future, self-checking the parameters of the AHA algorithm may be considered. Moreover, the processing of large datasets and the choice of a different architecture may be taken into consideration.

Author Contributions: Conceptualization, D.S.A.E.; Data curation, M.A.S.A. and D.S.A.E.; Formal analysis, M.A.S.A.; Funding acquisition, M.A.S.A.; Investigation, M.A.S.A. and D.S.A.E.; Methodology, M.A.S.A., F.R.P.P. and D.S.A.E.; Project administration, D.S.A.E.; Resources, D.S.A.E.; Software, D.S.A.E.; Supervision, D.S.A.E.; Validation, F.R.P.P.; Writing—original draft, M.A.S.A., F.R.P.P. and D.S.A.E.; Writing—review & editing, M.A.S.A. and D.S.A.E. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Project No. AN000550).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Geyer, R.; Jambeck, J.; Law, K. Producción, uso y destino de todos los plásticos jamás fabricados. *Sci. Adv.* **2017**, *3*, 1207–1221.
- Kumar, S.; Smith, S.; Fowler, G.; Velis, C.; Rena, K.; Kumar, R.; Cheeseman, C. Challenges and opportunities associated with waste management in India. *R. Soc. Open Sci.* **2017**, *4*, 160764. [[CrossRef](#)] [[PubMed](#)]
- Bircanoğlu, C.; Atay, M.; Beşer, F.; Genç, Ö.; Kızrak, M.A. RecycleNet: Intelligent waste sorting using deep neural networks. In Proceedings of the 2018 Innovations in Intelligent Systems and Applications (INISTA), Thessaloniki, Greece, 3–5 July 2018; pp. 1–7.
- Borowski, P.F. Environmental pollution as a threats to the ecology and development in Guinea Conakry. *Environ. Prot. Nat. Resour. Środowiska I Zasobów Nat.* **2017**, *28*, 27–32. [[CrossRef](#)]
- Zelazinski, T.; Ekielski, A.; Tulska, E.; Vladut, V.; Durczak, K. Wood dust application for improvement of selected properties of thermoplastic starch. *Inmateh. Agric. Eng* **2019**, *58*, 37–44.
- Tiyajamorn, P.; Lorprasertkul, P.; Assabumrungrat, R.; Poomarin, W.; Chanchaen, R. Automatic Trash Classification using Convolutional Neural Network Machine Learning. In Proceedings of the 2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), Bangkok, Thailand, 18–20 November 2019; pp. 71–76.
- Yu, Y. A Computer Vision Based Detection System for Trash Bins Identification during Trash Classification. In Proceedings of the Journal of Physics: Conference Series, 2nd International Conference on Electronic Engineering and Informatics, Lanzhou, China, 17–19 July 2020; IOP Publishing: Bristol, UK, 2020; Volume 1617, p. 012015.
- Ruiz, V.; Sánchez, Á.; Vélez, J.F.; Raducanu, B. Automatic image-based waste classification. In Proceedings of the International Work-Conference on the Interplay Between Natural and Artificial Computation, Almería, Spain, 3–7 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 422–431.
- Singh, A.; Rai, N.; Sharma, P.; Nagrath, P.; Jain, R. Age, Gender Prediction and Emotion recognition using Convolutional Neural Network. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC), New Delhi, India, 20–21 February 2021.
- Mohammadzadeh, H.; Gharehchopogh, F.S. An efficient binary chaotic symbiotic organisms search algorithm approaches for feature selection problems. *J. Supercomput.* **2021**, *77*, 9102–9144. [[CrossRef](#)]
- Naseri, T.S.; Gharehchopogh, F.S. A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems. *J. Netw. Syst. Manag.* **2022**, *30*, 1–27. [[CrossRef](#)]
- Abd Elminaam, D.S.; Nabil, A.; Ibraheem, S.A.; Houssein, E.H. An efficient marine predators algorithm for feature selection. *IEEE Access* **2021**, *9*, 60136–60153. [[CrossRef](#)]
- Faramarzi, A.; Heidarinejad, M.; Stephens, B.; Mirjalili, S. Equilibrium optimizer: A novel optimization algorithm. *Knowl.-Based Syst.* **2020**, *191*, 105190. [[CrossRef](#)]
- Elgamal, Z.M.; Yasin, N.M.; Sabri, A.Q.M.; Sihwail, R.; Tubishat, M.; Jarrah, H. Improved equilibrium optimization algorithm using elite opposition-based learning and new local search strategy for feature selection in medical datasets. *Computation* **2021**, *9*, 68. [[CrossRef](#)]
- AbdElminaam, D.S.; Neggaz, N.; Gomaa, I.A.E.; Ismail, F.H.; Elsayy, A. AOM-MPA: Arabic Opinion Mining using Marine Predators Algorithm based Feature Selection. In Proceedings of the 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC), Cairo, Egypt, 26–27 May 2021; pp. 395–402.
- Abd Elminaam, D.S.; Neggaz, N.; Ahmed, I.A.; Abouelyazed, A.E.S. Swarming behavior of Harris hawks optimizer for Arabic opinion mining. *Comput. Mater. Contin.* **2021**, *69*, 4129–4149. [[CrossRef](#)]
- Shaban, H.; Houssein, E.H.; Pérez-Cisneros, M.; Oliva, D.; Hassan, A.Y.; Ismaeel, A.A.; AbdElminaam, D.S.; Deb, S.; Said, M. Identification of Parameters in Photovoltaic Models through a Runge Kutta Optimizer. *Mathematics* **2021**, *9*, 2313. [[CrossRef](#)]

18. Abdelminaam, D.S.; Said, M.; Houssein, E.H. Turbulent flow of water-based optimization using new objective function for parameter extraction of six photovoltaic models. *IEEE Access* **2021**, *9*, 35382–35398. [[CrossRef](#)]
19. Deb, S.; Abdelminaam, D.S.; Said, M.; Houssein, E.H. Recent Methodology-Based Gradient-Based Optimizer for Economic Load Dispatch Problem. *IEEE Access* **2021**, *9*, 44322–44338. [[CrossRef](#)]
20. Deb, S.; Houssein, E.H.; Said, M.; Abdelminaam, D.S. Performance of Turbulent Flow of Water Optimization on Economic Load Dispatch Problem. *IEEE Access* **2021**, *9*, 77882–77893. [[CrossRef](#)]
21. El-Ashmawi, W.H.; Abd Elminaam, D.S. A modified squirrel search algorithm based on improved best fit heuristic and operator strategy for bin packing problem. *Appl. Soft Comput.* **2019**, *82*, 105565. [[CrossRef](#)]
22. Abdul-Minaam, D.S.; Al-Mutairi, W.M.E.S.; Awad, M.A.; El-Ashmawi, W.H. An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem. *IEEE Access* **2020**, *8*, 97959–97974. [[CrossRef](#)]
23. Dizaji, Z.A.; Gharehchopogh, F.S. A hybrid of ant colony optimization and chaos optimization algorithms approach for software cost estimation. *Indian J. Sci. Technol.* **2015**, *8*, 128. [[CrossRef](#)]
24. Gharehchopogh, F.S.; Abdollahzadeh, B. An efficient harris hawk optimization algorithm for solving the travelling salesman problem. *Clust. Comput.* **2022**, *25*, 1981–2005. [[CrossRef](#)]
25. Gharehchopogh, F.S.; Farnad, B.; Alizadeh, A. A modified farmland fertility algorithm for solving constrained engineering problems. *Concurr. Comput. Pract. Exp.* **2021**, *33*, e6310. [[CrossRef](#)]
26. Zaman, H.R.R.; Gharehchopogh, F.S. An improved particle swarm optimization with backtracking search optimization algorithm for solving continuous optimization problems. *Eng. Comput.* **2021**, *2021*, 1–35. [[CrossRef](#)]
27. Goldanloo, M.J.; Gharehchopogh, F.S. A hybrid OBL-based firefly algorithm with symbiotic organisms search algorithm for solving continuous optimization problems. *J. Supercomput.* **2022**, *78*, 3998–4031. [[CrossRef](#)]
28. Lan, P.; Xia, K.; Pan, Y.; Fan, S. An improved equilibrium optimizer algorithm and its application in LSTM neural network. *Symmetry* **2021**, *13*, 1706. [[CrossRef](#)]
29. Ma, H.; Simon, D.; Siarry, P.; Yang, Z.; Fei, M. Biogeography-based optimization: A 10-year review. *IEEE Trans. Emerg. Top. Comput. Intell.* **2017**, *1*, 391–407. [[CrossRef](#)]
30. Niccolai, A.; Bettini, L.; Zich, R. Optimization of electric vehicles charging station deployment by means of evolutionary algorithms. *Int. J. Intell. Syst.* **2021**, *36*, 5359–5383. [[CrossRef](#)]
31. Das, S.; Mullick, S.S.; Suganthan, P.N. Recent advances in differential evolution—an updated survey. *Swarm Evol. Comput.* **2016**, *27*, 1–30. [[CrossRef](#)]
32. Saidala, R.K.; Devarakonda, N. Multi-swarm whale optimization algorithm for data clustering problems using multiple cooperative strategies. *Int. J. Intell. Syst. Appl.* **2018**, *11*, 36. [[CrossRef](#)]
33. Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 43–55.
34. Elsis, M. Optimal design of nonlinear model predictive controller based on new modified multitracker optimization algorithm. *Int. J. Intell. Syst.* **2020**, *35*, 1857–1878. [[CrossRef](#)]
35. Massobrio, R.; Toutouh, J.; Nesmachnow, S.; Alba, E. Infrastructure deployment in vehicular communication networks using a parallel multiobjective evolutionary algorithm. *Int. J. Intell. Syst.* **2017**, *32*, 801–829. [[CrossRef](#)]
36. Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013.
37. Garg, H. Multi-objective optimization problem of system reliability under intuitionistic fuzzy set environment using Cuckoo Search algorithm. *J. Intell. Fuzzy Syst.* **2015**, *29*, 1653–1669. [[CrossRef](#)]
38. Bolaji, A.L.; Al-Betar, M.A.; Awadallah, M.A.; Khader, A.T.; Abualigah, L.M. A comprehensive review: Krill Herd algorithm (KH) and its applications. *Appl. Soft Comput.* **2016**, *49*, 437–446. [[CrossRef](#)]
39. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [[CrossRef](#)]
40. Eusuff, M.M.; Lansey, K.E. Water distribution network design using the shuffled frog leaping algorithm. In Proceedings of the Bridging the Gap: Meeting the World’s Water and Environmental Resources Challenges, Orlando, FL, USA, 20–24 May 2001; pp. 1–8.
41. Ma, H.; Ye, S.; Simon, D.; Fei, M. Conceptual and numerical comparisons of swarm intelligence optimization algorithms. *Soft Comput.* **2017**, *21*, 3081–3100. [[CrossRef](#)]
42. Garg, V.; Deep, K. Performance of Laplacian Biogeography-Based Optimization Algorithm on CEC 2014 continuous optimization benchmarks and camera calibration problem. *Swarm Evol. Comput.* **2016**, *27*, 132–144. [[CrossRef](#)]
43. Yang, G.P.; Liu, S.Y.; Zhang, J.K.; Feng, Q.X. Control and synchronization of chaotic systems by an improved biogeography-based optimization algorithm. *Appl. Intell.* **2013**, *39*, 132–143. [[CrossRef](#)]
44. García-Torres, J.M.; Damas, S.; Cordon, O.; Santamaría, J. A case study of innovative population-based algorithms in 3D modeling: Artificial bee colony, biogeography-based optimization, harmony search. *Expert Syst. Appl.* **2014**, *41*, 1750–1762. [[CrossRef](#)]
45. Ma, H. An analysis of the equilibrium of migration models for biogeography-based optimization. *Inf. Sci.* **2010**, *180*, 3444–3464. [[CrossRef](#)]
46. Ma, H.; Ni, S.; Sun, M. Equilibrium species counts and migration model tradeoffs for biogeography-based optimization. In Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference, Shanghai, China, 15–18 December 2009; pp. 3306–3310.

47. Ma, H.; Fei, M.; Ding, Z.; Jin, J. Biogeography-based optimization with ensemble of migration models for global numerical optimization. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012; pp. 1–8.
48. Gong, W.; Cai, Z.; Ling, C.X.; Li, H. A real-coded biogeography-based optimization with mutation. *Appl. Math. Comput.* **2010**, *216*, 2749–2758. [[CrossRef](#)]
49. Niu, Q.; Zhang, L.; Li, K. A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells. *Energy Convers. Manag.* **2014**, *86*, 1173–1185. [[CrossRef](#)]
50. Roy, P.; Mandal, D. Quasi-oppositional biogeography-based optimization for multi-objective optimal power flow. *Electr. Power Components Syst.* **2011**, *40*, 236–256. [[CrossRef](#)]
51. Kim, S.S.; Byeon, J.H.; Lee, S.; Liu, H. A grouping biogeography-based optimization for location area planning. *Neural Comput. Appl.* **2015**, *26*, 2001–2012. [[CrossRef](#)]
52. Feng, Q.; Liu, S.; Wu, Q.; Tang, G.; Zhang, H.; Chen, H. Modified biogeography-based optimization with local search mechanism. *J. Appl. Math.* **2013**, *2013*, 960524. [[CrossRef](#)]
53. Lim, W.L.; Wibowo, A.; Desa, M.I.; Haron, H. A biogeography-based optimization algorithm hybridized with tabu search for the quadratic assignment problem. *Comput. Intell. Neurosci.* **2016**, *2016*, 27. [[CrossRef](#)]
54. Yang, Y. A modified biogeography-based optimization for the flexible job shop scheduling problem. *Math. Probl. Eng.* **2015**, *2015*, 184643. [[CrossRef](#)]
55. Li, X.; Yin, M. Hybrid differential evolution with biogeography-based optimization for design of a reconfigurable antenna array with discrete phase shifters. *Int. J. Antennas Propag.* **2011**, *2011*, 685629. [[CrossRef](#)]
56. Sinha, S.; Bhola, A.; Panchal, V.; Singhal, S.; Abraham, A. Resolving mixed pixels by hybridization of biogeography based optimization and ant colony optimization. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, Australia, 10–15 June 2012; pp. 1–6.
57. Wang, G.G.; Gandomi, A.H.; Alavi, A.H. An effective krill herd algorithm with migration operator in biogeography-based optimization. *Appl. Math. Model.* **2014**, *38*, 2454–2462. [[CrossRef](#)]
58. Heidari, A.A.; Abbaspour, R.A.; Jordehi, A.R. An efficient chaotic water cycle algorithm for optimization tasks. *Neural Comput. Appl.* **2017**, *28*, 57–85. [[CrossRef](#)]
59. Krithiga, R.; Ilavarasan, E. A Novel Hybrid Algorithm to Classify Spam Profiles in Twitter. *Webology* **2020**, *17*, 260–279.
60. Sawhney, R.; Mathur, P.; Shankar, R. A firefly algorithm based wrapper-penalty feature selection method for cancer diagnosis. In Proceedings of the International Conference on Computational Science and Its Applications, Melbourne, Australia, 2–5 July 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 438–449.
61. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [[CrossRef](#)]
62. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Ala'M, A.Z.; Mirjalili, S.; Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl.-Based Syst.* **2018**, *154*, 43–67. [[CrossRef](#)]
63. Sayed, G.I.; Khoriba, G.; Haggag, M.H. A novel chaotic salp swarm algorithm for global optimization and feature selection. *Appl. Intell.* **2018**, *48*, 3462–3481. [[CrossRef](#)]
64. Harifi, S.; Khalilian, M.; Mohammadzadeh, J.; Ebrahimnejad, S. Emperor Penguins Colony: A new metaheuristic algorithm for optimization. *Evol. Intell.* **2019**, *12*, 211–226. [[CrossRef](#)]
65. Zheng, T.; Luo, W. An improved squirrel search algorithm for optimization. *Complexity* **2019**, *2019*, 6291968. [[CrossRef](#)]
66. Wang, Y.; Du, T. An improved squirrel search algorithm for global function optimization. *Algorithms* **2019**, *12*, 80. [[CrossRef](#)]
67. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]
68. Faramarzi, A.; Heidarinejad, M.; Mirjalili, S.; Gandomi, A.H. Marine predators algorithm: A nature-inspired Metaheuristic. *Expert Syst. Appl.* **2020**, *152*, 113377. [[CrossRef](#)]
69. Houssein, E.H.; Abdelminaam, D.S.; Hassan, H.N.; Al-Sayed, M.M.; Nabil, E. A Hybrid Barnacles Mating Optimizer Algorithm With Support Vector Machines for Gene Selection of Microarray Cancer Classification. *IEEE Access* **2021**, *9*, 64895–64905. [[CrossRef](#)]
70. Zhao, W.; Wang, L.; Mirjalili, S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114194. [[CrossRef](#)]
71. Halim, A.H.; Ismail, I.; Das, S. Performance assessment of the metaheuristic optimization algorithms: An exhaustive review. *Artif. Intell. Rev.* **2021**, *54*, 2323–2409. [[CrossRef](#)]
72. Liu, M.; Li, Y.; Huo, Q.; Li, A.; Zhu, M.; Qu, N.; Chen, L.; Xia, M. A two-way parallel slime mold algorithm by flow and distance for the travelling salesman problem. *Appl. Sci.* **2020**, *10*, 6180. [[CrossRef](#)]
73. Premkumar, M.; Jangir, P.; Sowmya, R.; Alhelou, H.H.; Heidari, A.A.; Chen, H. MOSMA: Multi-objective slime mould algorithm based on elitist non-dominated sorting. *IEEE Access* **2020**, *9*, 3229–3248. [[CrossRef](#)]
74. İzci, D.; Ekinci, S.; Ekinci, S. Comparative performance analysis of slime mould algorithm for efficient design of proportional–integral–derivative controller. *Electrica* **2021**, *21*, 151–159. [[CrossRef](#)]
75. Kumari, S.; Chugh, R. A novel four-step feedback procedure for rapid control of chaotic behavior of the logistic map and unstable traffic on the road. *Chaos Interdiscip. J. Nonlinear Sci.* **2020**, *30*, 123115. [[CrossRef](#)] [[PubMed](#)]

76. Mirjalili, S. SCA: A sine cosine algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
77. Mirjalili, S.; Mirjalili, S.M.; Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **2016**, *27*, 495–513. [[CrossRef](#)]
78. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, Australia, 27 November 27–1 December 1995; Volume 4, pp. 1942–1948.
79. Liu, Y.; Gong, D.; Sun, J.; Jin, Y. A many-objective evolutionary algorithm using a one-by-one selection strategy. *IEEE Trans. Cybern.* **2017**, *47*, 2689–2702. [[CrossRef](#)]
80. El-Sehiemy, R.; Abou El Ela, A.A.; Shaheen, A. A multi-objective fuzzy-based procedure for reactive power-based preventive emergency strategy. In *International Journal of Engineering Research in Africa*; Trans Tech Publications Ltd.: Bäch, Switzerland, 2015; Volume 13, pp. 91–102.
81. Shaheen, A.M.; El-Sehiemy, R.A. Application of multi-verse optimizer for transmission network expansion planning in power systems. In Proceedings of the 2019 International Conference on Innovative Trends in Computer Engineering (ITCE), Aswan, Egypt, 2–4 February 2019; pp. 371–376.
82. Shaheen, A.M.; El-Sehiemy, R.A.; Elattar, E.E.; Abd-Elrazek, A.S. A modified crow search optimizer for solving non-linear OPF problem with emissions. *IEEE Access* **2021**, *9*, 43107–43120. [[CrossRef](#)]
83. Jeddi, B.; Einaddin, A.H.; Kazemzadeh, R. A novel multi-objective approach based on improved electromagnetism-like algorithm to solve optimal power flow problem considering the detailed model of thermal generators. *Int. Trans. Electr. Energy Syst.* **2017**, *27*, e2293. [[CrossRef](#)]
84. Yu, W.; Zhang, J. Multi-population differential evolution with adaptive parameter control for global optimization. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 1093–1098.
85. Pedrosa Silva, R.C.; Lopes, R.A.; Guimarães, F.G. Self-adaptive mutation in the differential evolution. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; pp. 1939–1946.
86. Gao, X.Z.; Wang, X.; Ovaska, S.J.; Zenger, K. A hybrid optimization method based on differential evolution and harmony search. *Int. J. Comput. Intell. Appl.* **2014**, *13*, 1450001. [[CrossRef](#)]
87. Islam, S.M.; Das, S.; Ghosh, S.; Roy, S.; Suganthan, P.N. An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2011**, *42*, 482–500. [[CrossRef](#)]
88. Biswas, S.; Kundu, S.; Das, S.; Vasilakos, A.V. Teaching and learning best differential evolution with self adaptation for real parameter optimization. In Proceedings of the 2013 IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 1115–1122.
89. Zou, D.; Wu, J.; Gao, L.; Li, S. A modified differential evolution algorithm for unconstrained optimization problems. *Neurocomputing* **2013**, *120*, 469–481. [[CrossRef](#)]
90. Bujok, P.; Tvrdík, J.; Poláková, R. Differential evolution with rotation-invariant mutation and competing-strategies adaptation. In Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, 6–11 July 2014; pp. 2253–2258.
91. Gong, W.; Cai, Z.; Wang, Y. Repairing the crossover rate in adaptive differential evolution. *Appl. Soft Comput.* **2014**, *15*, 149–168. [[CrossRef](#)]
92. Tran, D.H.; Cheng, M.Y.; Cao, M.T. Hybrid multiple objective artificial bee colony with differential evolution for the time–cost–quality tradeoff problem. *Knowl.-Based Syst.* **2015**, *74*, 176–186. [[CrossRef](#)]
93. Chang, L.; Liao, C.; Lin, W.; Chen, L.L.; Zheng, X. A hybrid method based on differential evolution and continuous ant colony optimization and its application on wideband antenna design. *Prog. Electromagn. Res.* **2012**, *122*, 105–118. [[CrossRef](#)]
94. Biswal, B.; Behera, H.S.; Bisoi, R.; Dash, P.K. Classification of power quality data using decision tree and chemotactic differential evolution based fuzzy clustering. *Swarm Evol. Comput.* **2012**, *4*, 12–24. [[CrossRef](#)]
95. Chakraborti, T.; Chatterjee, A.; Halder, A.; Konar, A. Automated emotion recognition employing a novel modified binary quantum-behaved gravitational search algorithm with differential mutation. *Expert Syst.* **2015**, *32*, 522–530. [[CrossRef](#)]
96. Basak, A.; Maity, D.; Das, S. A differential invasive weed optimization algorithm for improved global numerical optimization. *Appl. Math. Comput.* **2013**, *219*, 6645–6668. [[CrossRef](#)]
97. Abdullah, A.; Deris, S.; Anwar, S.; Arjunan, S.N. An evolutionary firefly algorithm for the estimation of nonlinear biological model parameters. *PLoS ONE* **2013**, *8*, e56310. [[CrossRef](#)]
98. Zheng, Y.J.; Xu, X.L.; Ling, H.F.; Chen, S.Y. A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing* **2015**, *148*, 75–82. [[CrossRef](#)]
99. Sharma, M.; Kaur, P. A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. *Arch. Comput. Methods Eng.* **2021**, *28*, 1103–1127. [[CrossRef](#)]
100. Xue, Y.; Xue, B.; Zl, M. Self-Adaptive particle swarm optimization for large-scale feature selection in classification. *ACM Trans. Knowl. Discov. Data* **2019**, *13*, 1–27. [[CrossRef](#)]
101. Zhang, K.; Lan, L.; Wang, Z.; Moerchen, F. Scaling up kernel svm on limited resources: A low-rank linearization approach. In Proceedings of the Artificial Intelligence and Statistics, PMLR, La Palma, Spain, 21–23 April 2012; pp. 1425–1434.
102. Costa, B.S.; Bernardes, A.C.; Pereira, J.V.; Zampa, V.H.; Pereira, V.A.; Matos, G.F.; Soares, E.A.; Soares, C.L.; Silva, A.F. Artificial intelligence in automated sorting in trash recycling. In Proceedings of the Anais do XV Encontro Nacional de Inteligência Artificial e Computacional, Sao Paulo, Brazil, 22–25 October 2018; pp. 198–205.

103. Satvilkar, M. Image Based Trash Classification Using Machine Learning Algorithms for Recyclability Status. Ph.D. Thesis, National College of Ireland, Dublin, Ireland, 2018.
104. Sousa, J.; Rebelo, A.; Cardoso, J.S. Automation of waste sorting with deep learning. In Proceedings of the 2019 XV Workshop de Visão Computacional (WVC), Sao Paulo, Brazil, 9–11 September 2019; pp. 43–48.
105. Zhu, S.; Chen, H.; Wang, M.; Guo, X.; Lei, Y.; Jin, G. Plastic solid waste identification system based on near infrared spectroscopy in combination with support vector machine. *Adv. Ind. Eng. Polym. Res.* **2019**, *2*, 77–81. [[CrossRef](#)]
106. Özkan, K.; Ergin, S.; Işık, Ş.; Işıklı, İ. A new classification scheme of plastic wastes based upon recycling labels. *Waste Manag.* **2015**, *35*, 29–35. [[CrossRef](#)] [[PubMed](#)]
107. Aral, R.A.; Keskin, Ş.R.; Kaya, M.; Hacıömeroğlu, M. Classification of trashnet dataset based on deep learning models. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2058–2062.
108. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
109. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*. [[CrossRef](#)]
110. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
111. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
112. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
113. Hamida, M.A.; El-Sehiemy, R.A.; Ginidi, A.R.; Elattar, E.; Shaheen, A.M. Parameter identification and state of charge estimation of Li-Ion batteries used in electric vehicles using artificial hummingbird optimizer. *J. Energy Storage* **2022**, *51*, 104535. [[CrossRef](#)]
114. Abid, M.S.; Apon, H.J.; Morshed, K.A.; Ahmed, A. Optimal Planning of Multiple Renewable Energy-Integrated Distribution System with Uncertainties Using Artificial Hummingbird Algorithm. *IEEE Access* **2022**, *10*, 40716–40730. [[CrossRef](#)]
115. Ramadan, A.; Kamel, S.; Hassan, M.H.; Ahmed, E.M.; Hasanien, H.M. Accurate Photovoltaic Models Based on an Adaptive Opposition Artificial Hummingbird Algorithm. *Electronics* **2022**, *11*, 318. [[CrossRef](#)]
116. Sadoun, A.M.; Najjar, I.R.; Alsuruji, G.S.; Abd-Elwahed, M.; Elaziz, M.A.; Fathy, A. Utilization of improved machine learning method based on artificial hummingbird algorithm to predict the tribological behavior of Cu-Al₂O₃ nanocomposites synthesized by in situ method. *Mathematics* **2022**, *10*, 1266. [[CrossRef](#)]
117. Yang, M.; Thung, G. Classification of trash for recyclability status. *CS229 Proj. Rep.* **2016**, *2016*, 3.
118. Zheng, Y.; Yang, C.; Merkulov, A. Breast cancer screening using convolutional neural network and follow-up digital mammography. In *Computational Imaging III*; International Society for Optics and Photonics: Bellingham, WA, USA, 2018; Volume 10669, p. 1066905.
119. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity mappings in deep residual networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 630–645.
120. Zayed, M.E.; Zhao, J.; Li, W.; Elsheikh, A.H.; Abd Elaziz, M. A hybrid adaptive neuro-fuzzy inference system integrated with equilibrium optimizer algorithm for predicting the energetic performance of solar dish collector. *Energy* **2021**, *235*, 121289. [[CrossRef](#)]
121. Aarts, E.; Aarts, E.H.; Lenstra, J.K. *Local Search in Combinatorial Optimization*; Princeton University Press: Princeton, NJ, USA, 2003.
122. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Sydney, Australia, 28 November–1 December 2005; Volume 1, pp. 695–701.
123. Long, W.; Jiao, J.; Liang, X.; Cai, S.; Xu, M. A random opposition-based learning grey wolf optimizer. *IEEE Access* **2019**, *7*, 113810–113825. [[CrossRef](#)]
124. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
125. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [[CrossRef](#)]
126. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Future Gener. Comput. Syst.* **2019**, *101*, 646–667. [[CrossRef](#)]
127. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. *Swarm Intell.* **2007**, *1*, 33–57. [[CrossRef](#)]
128. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
129. Hashim, F.A.; Hussain, K.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **2021**, *51*, 1531–1551. [[CrossRef](#)]
130. Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [[CrossRef](#)]
131. Shabani, A.; Asgarian, B.; Salido, M.; Gharebaghi, S.A. Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems. *Expert Syst. Appl.* **2020**, *161*, 113698. [[CrossRef](#)]

132. Rabano, S.L.; Cabatuan, M.K.; Sybingco, E.; Dadios, E.P.; Calilung, E.J. Common garbage classification using mobilenet. In Proceedings of the 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 29 November–2 December 2018; pp. 1–4.
133. Kennedy, T. OscarNet: Using transfer learning to classify disposable waste. *CS230 Report: Deep Learning*; Stanford University: Stanford, CA, USA, 2018.
134. Zhang, Q.; Zhang, X.; Mu, X.; Wang, Z.; Tian, R.; Wang, X.; Liu, X. Recyclable waste image recognition based on deep learning. *Resour. Conserv. Recycl.* **2021**, *171*, 105636. [[CrossRef](#)]
135. Yang, Z.; Li, D. WasNet: A Neural Network-Based Garbage Collection Management System. *IEEE Access* **2020**, *8*, 103984–103993. [[CrossRef](#)]
136. Shi, C.; Xia, R.; Wang, L. A Novel Multi-Branch Channel Expansion Network for Garbage Image Classification. *IEEE Access* **2020**, *8*, 154436–154452. doi: 10.1109/ACCESS.2020.3016116. [[CrossRef](#)]
137. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.