

Adaptation of metaheuristic algorithms to improve training performance of an ESZSL model

Şifa ÖZSARI¹, Mehmet Serdar GÜZEL¹, Gazi Erkan BOSTANCI¹, Ayhan AYDIN*¹

Department of Computer Engineering, Faculty of Engineering, Ankara University, Ankara, Turkey

Received: 24.08.2020

Accepted/Published Online: 10.12.2020

Final Version: 31.05.2021

Abstract: Zero-shot learning (ZSL) is a recent promising learning approach that is similar to human vision systems. ZSL essentially allows machines to categorize objects without requiring labeled training data. In principle, ZSL proposes a novel recognition model by specifying merely the attributes of the category. Recently, several sophisticated approaches have been introduced to address the challenges regarding this problem. Embarrassingly simple approach to zero-shot learning (ESZSL) is one of the critical of those approaches that basically proposes a simple but efficient linear code solution. However, the performance of the ESZSL model mainly depends on parameter selection. Metaheuristic algorithms are considered as one the most sophisticated computational intelligence paradigms that allows to approximate optimization problems with high success. This paper addresses this problem by adapting leading metaheuristic algorithms to automatically train the parameters of a linear ESZSL model. The model is statistically validated by performing a series of experiments with benchmark datasets.

Key words: Zero-shot learning, metaheuristic algorithms, genetic algorithm, particle swarm optimization

1. Introduction

Classification is a process related to categorization. In machine learning, this action involves supervised learning techniques that aims to design models for different problems. Each models are able to map an input to output for a specific problem based on sample input/output pairs. There are many different classification tasks that can be encountered in machine learning and specific approaches to modeling that can be used for each. One of the critical problem encountered in this field is to perform classification automatically. Consequently, it has been extensively studied and various ideas have been proposed. However, recognition systems are not able to cope with the appearance of a new class after the training phase is completed. Another critical issues are to collect as many sample images as possible to generate object classes appropriately. Besides, these images also have to be taken from different angles in various contexts. Finally, it should be noted that collecting sample images for every object is not possible. For instance, despite there exists lots of animal classes for which collecting sample in an effortless manner is possible, there are animals in extinction, as shown in Figure 1, which does not allow to collect image samples effortlessly. These problems motivate researches to focus on ZSL approaches.

ZSL is a new machine learning technique that aims to recognize new sample sets and assign them into unseen but suitable classes without requiring any training examples but only necessitating the descriptions of these instances. The descriptions of the new class examples are associated with the definitions of the classes previously learned. In other words, ZSL is an approach which can solve a task without encountering any sample

*Correspondence: ayaydin@ankara.edu.tr



Figure 1. Ili Pika was last seen in 2017, after 20 years of disappearance.

of that task at training stage [1]. This enormous paradigm is inspired from humans that any human being is able to recognize a new (unseen) object by reading and analyzing only some semantic definitions by considering the similarities between the definition of the new object and the concepts already learned [1]. The seen and unseen classes are linked by the semantic descriptions. According to this, an expected idea, visual semantic mapping, is employed. This compares both seen and unseen classes in semantic space [2]. The earlier literature includes indirect learning method for unseen samples, which however, associates one semantic description with vast number of visual samples [3, 4].

Alternatively, several recent approaches handle the problem directly in which "generative adversarial networks (GANs)" are employed, which recognize unseen samples from the random noises and semantic definitions [5, 6]. One of the notable approaches in the ZSL domain is the one-line linear solution introduced by Romero and Torr [1] in 2015. According to the proposed model, a loss function, and a regularizer, simpler but also efficient way of constructing the complete process is introduced. The solution involves two critical parameters directly influences the accuracy of the model. They employ the validation set so as to tune the hyper-parameters of the methods. This set contains instances, which belongs to "20%" of the training classes, selected randomly. They use the value range, 10^b for $b = -3, -2, \dots, 2, 3$ in order to tune all hyperparameters. Nevertheless, it should be stated that the accuracy rates for parameters of real numbers parameters have not been observed. This motivates authors to overcome this problem by employing approximation algorithms. Accordingly, two leading metaheuristics algorithms, namely: genetic algorithm (GA) and particle swarm optimization (PSO) are adapted and then evaluated to estimate the parameters of ESZSL. The confidence of the algorithms is tested on benchmark datasets statistically. The results reveal the performance of these algorithms for this problem. The structure of paper consists of the Sections named as: materials and methods, experimental results, and conclusion.

2. Materials and methods

The main concept of employing semantic description to represent a class was first discussed in [7]. In this study, error-correcting code is seen as a kind of communication problem where the correct output class identity is "transmitted" over a channel for a new example. Binary descriptors are used as error-correcting codes. However, they do not have a semantic meaning. In computer vision, the methods of feature sharing between object classes were studied in [8]. This method uses experiences with previously learned classes to make learning of new classes easier. Features that have been learned and proven useful in the previous classification process are selected and are adapted to the new classification process. In this adaptation, the properties of classes previously learned are changed with the similar features from the new class. Another study focused on predicting the visual properties of new objects using visual property predictions for object recognition. They introduced attribute based classification. Instead of training images, object recognition is performed on the basis of high level definitions of target objects. The high level definitions consist of semantic attributes. New classes can be identified according to their representation, without new training phase [9].

Until 2012, studies concerning computer vision have shown limited performance due to hardware limitation and time costs. Krizhevsky et al. achieved a success of 83.6% in Deep Learning's Image Net Large-Scale Visual Recognition Competition (ILSVRC) in 2012 [10]. This success, in essence, has changed the direction of the researches in this field. Accordingly, many academic studies based on deep learning have been carried out within the following years. Despite deep learning is able to achieve outstanding performance, it still needs to handle a crucial problem that already exists in traditional machine learning methods according to which, the data should be diverse enough to reflect the real world and the data must have sufficient information. To overcome these difficulties, there has recently been an increase in research of ZSL [11]. Most studies employing zero-shot have learning consists of two stages: training and inference. In the training phase, information of attributes is obtained, and, in the inference, phase this information is used to categorize the unseen samples [9, 12–14]. For example, in DAP, [9], at training stage, the posterior of each attribute is estimated by learning probabilistic classifier. At the second stage, the preceding estimators are used to calculate the class posteriors, and the new classes are predicted using their attributes signatures. Similarly, in IAP, a probability classifier is generated for each training class, but at inference phase, forecasts are combined considering both the attributes of training classes and test classes. A "multiclass" classifier is used to estimate the class posterior of seen classes.

ESZSL describes zero-shot learning approach by considering the relationships between features, attributes, and classes [1]. This framework consists of two linear layers. Framework integrates the training stage and the inference stage. The weights of the first layer, which are learned at the training stage, model the association among the features and the attributes. On the other hand, the weights of the second layer are not learned, they are obtained from the environment. The relationship between the attributes and the classes is modeled by the second layer. This layer, in essence, is fixed by employing the prearranged attribute signatures of the classes.

In Study [15], Gencer Sumbul et al. 2017 examined problems that occurred in several cases regarding the recognition and detection systems. These situations where problems arise are the escalation in spatial and spectral resolution, the emergence of novel details, the increase of new classes, and the diversity of target classes.

2.1. ESZSL

ESZSL is a one-line liner code solution introduced by Romero and Torr in 2015. The approach proposed in the study is mainly based on transfer learning. In the transfer learning, which is also acknowledged as learning to learn [16] or inductive transfer [17–19], the information obtained while solving a problem is stored and then is

used to solve other related problems. In terms of this feature, it is similar to “zero-shot learning”. However, there is a significant difference between transfer learning and ZSL. In transfer learning, data about new tasks is given as a series of labeled samples. Romero and Torr [1] present a linear model, which is based on defining the relationship between features, attributes, and classes. This model is similar to one introduced in [20]. The presented framework is able to integrate both training and inference stage, overcoming the general deficiencies, which are seen previous studies.

ESZSL has important notations used in training and inference phase. At the training phase, there are “ z classes”, each of which has a signature composed of a attributes. These signatures are symbolized in a matrix namely, $S \in [0, 1]^{a \times z}$. The instances obtainable at training phase are symbolized by $X \in R^{d \times m}$. Here d represents the dimensionality of the data, and m is the number of samples. The ground truth labels of each training instance that belongs to any of the z classes are indicated by $Y \in \{-1, 1\}^{m \times z}$. ESZSL framework integrates the training stage and the inference stage. The weights of the first layer are obtained at the training stage. In this case, the solution is expressed in equation 1 [1].

$$V = (XX^T + \gamma I)^{-1}XY S^T(S S^T + \lambda I)^{-1} \tag{1}$$

where, I is unit matrix and γ, λ are hyper parameters, which affect the performance of the algorithm. In this study, with using GA and PSO, these hyper parameters have been tuned.

In the inference phase, the classes of samples never seen before are estimated using the model obtained from the training phase. When a new sample is given, x is estimated using the equation 2 [1]:

$$\arg \max_i X^T V S_i' \tag{2}$$

here, S' refers to the attribute signatures of the new set of z' classes. A sample architecture of the framework is shown in Figure 2 [1], and this framework is adapted from a previous study can be seen in [20].

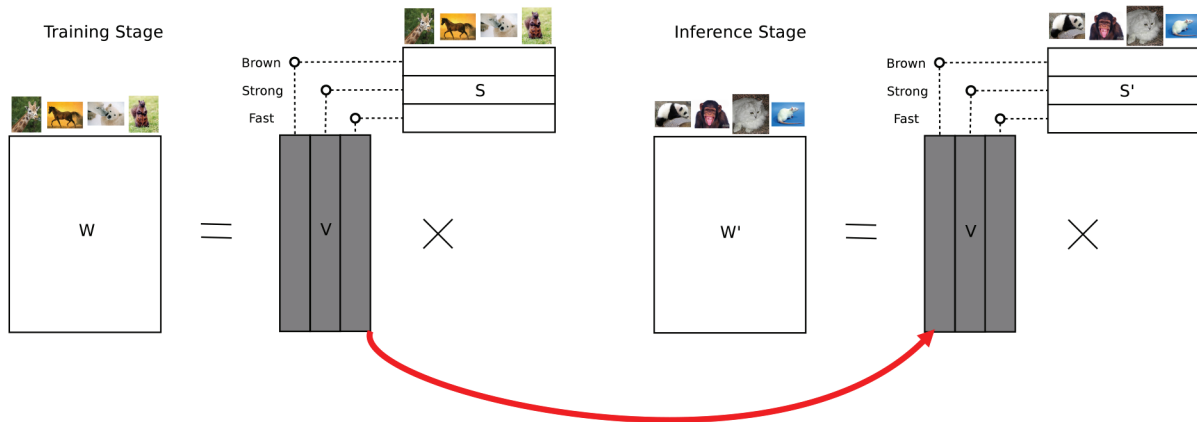


Figure 2. Summary of the ESZSL framework can be seen in [1].

According to the framework, at the training phase, the V matrix, which maps from the feature space to the attribute space, is learned through the signature matrix (S) with the training samples. At inference stage, the final linear model (W') is obtained by using the matrix V , which is learned at the training stage, together with the attribute signatures of the test classes (S').

2.2. Datasets

For the experimental process of this study, benchmark datasets that are commonly preferred in zero-shot learning are employed. These are the “Animals with Attributes dataset (AwA1)” [21], “the SUN scene attributes database (SUN)” [22], the “aPascal/aYahoo objects dataset (aPY)” [23], “the Caltech-UCSD Birds 200-2011 (CUB)” [24], and the “Animals with Attributes dataset 2 (AwA2)” [25]. These datasets contain set of images with various classes in a different context. The aPY, which contains images of objects, is a “coarse-grained” and “small-scale” dataset. It involves 64 attributes and 32 classes. The original “AwA1” and “AwA2” are coarse-grained datasets and have 50 classes and 85 attributes. In “AwA1” and “AwA2”, there are images of various animal classes. The “CUB” is a medium scale and fine-grained dataset of birds. It contains 200 classes and 312 attributes. The “SUN” is a fine-grained and medium-scale dataset. The “SUN” has 717 types of scenes and 102 attributes. Table 1 illustrates the characteristics of aforementioned datasets and sample images obtained from “aPY” dataset are shown in Figure 3.

Table 1. The characteristics of datasets.

	“aPY”	“AwA1”	“AwA2”	“CUB”	“SUN”
“Training classes”	20	40	40	150	707
“Test classes”	12	10	10	50	10
“Instances”	15339	30475	37322	11788	14340
“Attributes”	64	85	85	312	102



Figure 3. Sample images obtained from ”aPY” dataset [23].

2.3. Genetic algorithm

Genetic algorithm is a random search algorithm that aims to find the optimal solution. It was developed by Holland in 1975 based on the theory of evolution [26]. Since this algorithm is based on the principle of natural selection, it is included in the group of heuristic algorithms. According to natural selection, the most suitable individuals in the population are selected to produce the next generation. Based on this notion in genetic

algorithm, firstly an initial population is randomly created. Each candidate solution (individual) consists of a set of genes. Generally, binary form (0s and 1s) is used to represent individuals, but there are different coding methods. It is aimed to find the most optimal solution with the evolution of individuals in the population. The genetic algorithm works iteratively, and the population that gets after each iteration is called the new generation. The following steps are performed to each generation:

1. The fitness value of each individual in the population is determined according to fitness function in the optimization problem.
2. Generally, in order to protect best individuals, the best candidate solutions in a certain percentage are transferred to next population without any processing. This is called elitism.
3. The individuals that will be used to create the individuals of the new generation are determined according to the selection method.
4. After the selection process, a new generation is created by applying crossover and mutation to these selected individuals. Crossover is creation of new individuals by crossing two individuals according to crossover method. Mutations are random changes in individuals to prevent the population from becoming same (to preserve diversity).
5. The new population is then used in the next generation.
6. Until the stop condition is met, these steps are repeated.

It should be noted that the algorithm is terminated in a certain number of iterations or when a certain fitness value is provided. The flowchart representation of genetic algorithm (GA) is shown in Figure 4 and algorithm 1 shows the pseudocode of GA. GA has an ease implementation and it can be directly carried out on continuous and discrete problems. Therefore, genetic algorithm has been used on many problems ranging from engineering applications to healthcare, and it will be kept using [27].

Algorithm 1 Pseudocode of GA.

```

Generate the initial population
Compute fitness
while termination condition not met do
  Elitism
  Selection
  Crossover
  Mutation
  Compute fitness
end while

```

2.4. Particle swarm optimization

Particle swarm optimization (PSO) is also a metaheuristic technique and was proposed by Kennedy and Eberhart [28]. PSO is adapted by the movements of animals that move in flocks, such as fish or birds, to meet their basic needs, such as finding food, for optimizing continuous nonlinear functions. In these groups, a leader guides the navigation of the entire herd. The movement of each individual is determined by the position of the leader (global best) and the local best. The working steps of the PSO algorithm are basically given as follows:

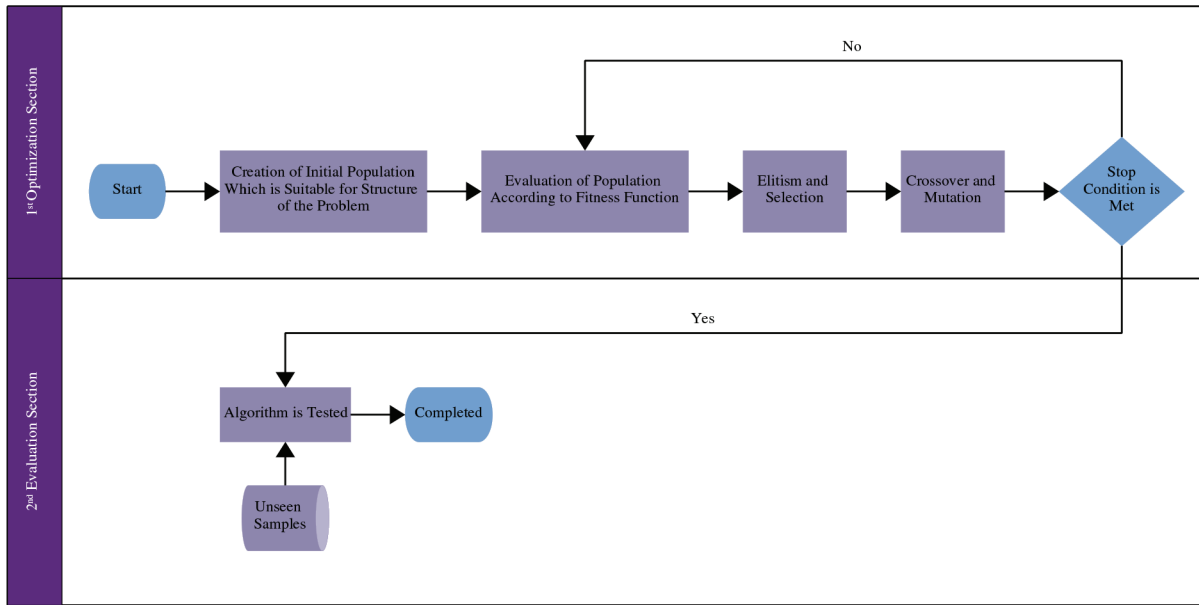


Figure 4. A detailed flowchart representation of GA.

- A random assignment is made to the position and speed of each particle “*p*” in the herd. In the N-dimensional search space, the velocity and position of the particle “*p*” are denoted by the vectors. $X_p=(x_{p1},x_{p2},x_{p3},\dots,x_{kn})$ and $V_p=(v_{p1},v_{p2},v_{p3},\dots,v_{pn})$ correspond to position and the flight speed of the particle “*p*” in the search space.
- Each particle “*p*” knows its position, the fitness value for that position and the best value it has found until then. This best value is called as “pbest”.
- In each iteration “*t*”, the movement of particle is determined according to pbest and gbest. While “gbest” refers the best global position found by all particles, “pbest” refers the current location of the corresponding particle.

The velocity and position of the particle are updated for each iteration based on the equations given below:

$$V_{pd}^{t+1} = wv_{pd}^t + c_1r_1(p_{pd}^t - x_{pd}^t) + c_2r_2(g_d^t - x_{pd}^t) \tag{3}$$

$$x_{pd}^{t+1} = x_{pd}^t + V_{pd}^{t+1} \tag{4}$$

here, “*w*”, which is called the inertia weight suggested by [29], is utilized to control the parameter of the swarm velocity, whereas “*c*₁” and “*c*₂” are called acceleration coefficients. They are the weights that determine how much a particle must move towards its local best and global best within swarm. In general, while the values of “*c*₁” and “*c*₂” are kept constant, “*r*₁” and “*r*₂” are randomly generated variables, defined in [0, 1]. When “*c*₁” and “*c*₂” (the acceleration coefficients) are multiplied with random variables: “*r*₁” and “*r*₂”, velocity of the swarm can be controllable [30], and *p* and *g* refer pbest and gbest parameters, respectively. Figure 5 illustrates the flowchart of PSO algorithm and also Algorithm 2 involves the pseudocode of PSO.

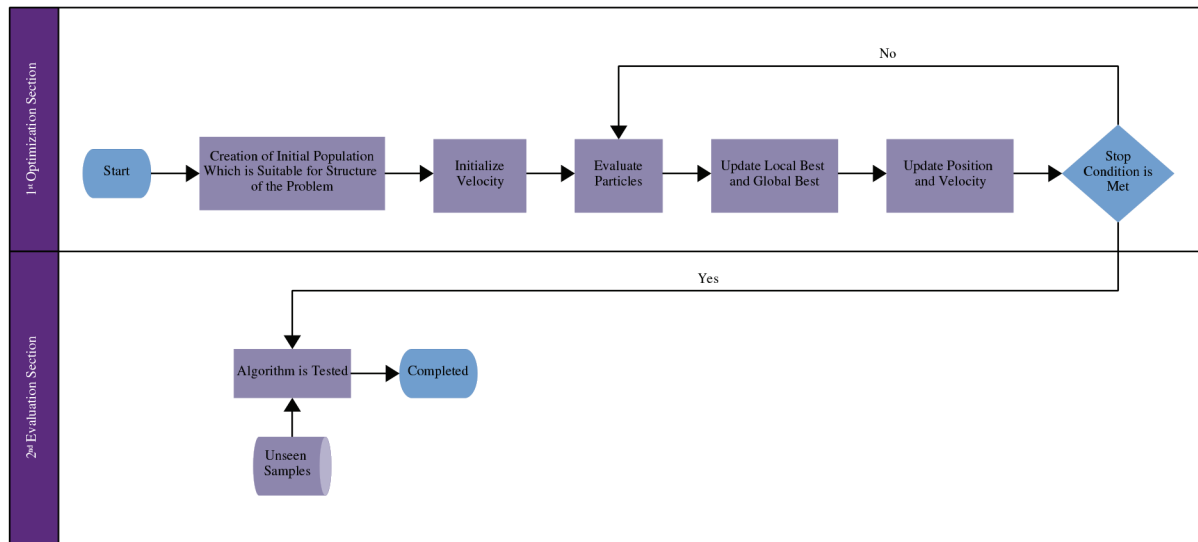


Figure 5. A detailed flowchart representation of PSO algorithm.

Algorithm 2 Conventional PSO approach [31–35].

```

Initialize population with random position
Initialize velocity of population
while stop condition is not met do
    Calculate the fitness value of each particle
    Update the pbest of each particle
    Update the gbest
    Calculate the velocity of each particle
    Update the position of each particle
end while
    
```

3. The Experimental section and discussion

This section represents the proposed model performance using GA and PSO for tuning hyper parameters on benchmark datasets. In order to train the models and perform the experiments, a computer of medium configuration is employed. The features of the computer are: Intel Core i5, 3.40 GHz CPU, and 4 GB RAM. During the evaluation process, the accuracy rate is used as fitness function, and all of the training examples are used for optimization. After the parameter are adjusted, the model is obtained according to these parameters. The success of GA and PSO has been determined by classifying examples that have not been seen before (i.e. they are not used in the training phase) using this model.

3.1. Parameter settings

There are some parameters that affect the success of the genetic algorithm and PSO, and therefore, their choices are crucial. In this study, parameters which are used in genetic algorithm are set as follows:

- “Population size”: 100
- “Number of iteration”: 100
- “Crossover rate”: 0.8

- “Mutation rate”: 0.1

Parameters which are used in particle swarm optimization are set as follows:

- “Population size”: 100
- “Number of iteration”: 100
- Constants “ c_1 ” and “ c_2 ”: 2

Table 2a illustrates the best, average, and worst accuracy rate (%) obtained only by executing the ESZSL algorithm without employing parameter optimization algorithms. In Table 2a, results for datasets are not obtained from [1]. Accuracy rates of datasets are obtained by running 10 times ESZSL using $b = -3, -2, -1, 0, 1, 2, 3$ values for 10^b .

Table 2. Accuracy rate (%) results. (a) The accuracy rate of datasets obtained without parameter optimization; (b) the accuracy rates obtained from optimized model using GA on benchmark datasets; (c) the accuracy rates obtained from the optimized model using PSO on benchmark datasets.

(a)

Dataset	Best accuracy	Average accuracy	Worst accuracy
“aPY”	29.45	25.07	22.92
“AwA1”	45.79	44.50	44.15
“AwA2”	53.14	51.53	50.92
“CUB”	53.93	50.73	48.03
“SUN”	60.00	58.25	57.5

(b)

Dataset	Best accuracy	Average accuracy	Worst accuracy
“aPY”	31.26	31.14	27.59
“AwA1”	59.14	58.69	51.73
“AwA2”	57.73	57.41	53.51
“CUB”	54.94	54.66	52.75
“SUN”	67.50	66.65	60.00

(c)

Dataset	Best accuracy	Average accuracy	Worst accuracy
“aPY”	31.30	31.28	30.94
“AwA1”	59.14	59.02	54.46
“AwA2”	57.78	57.77	57.68
“CUB”	54.904	54.84	54.23
“SUN”	67.50	67.32	60.00

3.2. Application of GA on the problem

GA is applied to problem by using the fitness function. Firstly, initial population, which is suitable for structure of problem, is created randomly. An individual consists of two real value in the range of $(-4, 4)$. One value is for λ , the other is for γ . Once the initial population has been created, the fitness value of individuals is calculated using the fitness function. In order to protect best individuals, the best candidate solutions in a certain percentage are transferred to next population without any processing. Individuals to be used in crossover are determined by triple tournament selection method. As a crossover method, single point crossover has been

used. After crossover has been done, mutation process is started. Value change mutation method is applied in a low percentage to ensure diversity of population. A new population is obtained after mutation. This population is evaluated by sending to the fitness function. Until the stop condition is met, elitism, selection, crossover, and mutation process are repeated. When 75% of the population has same individual, in other words, diversity of population is smaller than 25% of the population size, reproduction is done in order to preserve diversity. Genetic algorithm is applied to dataset of “aPY”, “AwA1”, “AwA2”, “SUN”, and “CUB”. GA is executed in total 10 times for 100 iterations on datasets. Table 2b illustrates the best, average, and worst accuracy rate by using the GA optimized model. According to the result shown in Table 2b, it can be stated that (ESZSL+GA) is able to obtain better results in all datasets.

3.3. Application of PSO on the problem

In addition to GA, PSO is also applied to problem by using the fitness function. Firstly, initial population, which is suitable for structure of problem is created. A particle consists of two real value in the range of $[-4, 4]$. One value is for λ , the other is for γ . Initial velocity of particles are zero. Once the population has been created, the fitness value of individuals is calculated using the fitness function. According to fitness value of the particles, the gbest is determined. The pbest of each particle is equal to itself. Equations 3 and 4 are used in order to modify position and velocity values of particles. If the velocity of a particle exceeds the lower limit, its value is substituted with the lower limit, and if it exceeds the upper limit, its value is substituted with the upper limit. In the same manner, if the position of particle exceeds the lower limit, its value is substituted with the lower limit, and if it exceeds the upper limit, its value is also substituted with the upper one. After updating of velocity and position are completed, the fitness value of particles is calculated. The pbest and the gbest are updated by considering this new fitness values. According to pbest and gbest values, velocity and position parameter of each particle is recalculated. These steps are repeated until the stop condition is accomplished. As in the genetic algorithm, when 75% of the population has same individual, reproduction is done in order to preserve diversity. (PSO+ESZSL) is applied to “aPY”, “AwA1”, “AwA2”, “SUN”, and “CUB” datasets. PSO is executed in total 10 times with 100 iterations on these benchmark datasets. Table 2c presents the best, average, and worst accuracy rate (%) generated by employing the PSO algorithm so as to optimize the parameters of the model.

According to the results shown in Table 2c, it can be stated that (ESZSL+PSO) yields better results in “aPY”, “AwA2” datasets than (ESZSL+GA) can achieve. While both models have the same accuracy rate in the “AwA1” and “SUN” datasets, (ESZSL+GA) achieves better results in the “CUB” dataset. Figures 6a, 6b, 7a, 7b, 8 illustrate the convergence graph of GA and PSO on “aPY”, “AwA1”, “AwA2”, “CUB”, and “SUN” datasets, respectively. These figures belong to the individual with the best fitness value (highest success rate).

Once the convergence graphs are analyzed, it should be noted that despite PSO achieves the best success rate in a shorter time on “aPY”, “AwA2”, and “SUN” datasets, GA shows a faster convergence in “AwA1” and “CUB” datasets. In addition, computation complexity of algorithms are addressed to analyze their effectiveness. By considering running times, it is calculated $O(n^3)$ for GA and $O(n^2)$ for PSO, respectively. As can be understood from these values, PSO produces faster results than GA. Finally, t-test is applied to the ESZSL model and the model used GA (ESZSL+GA) and PSO (ESZSL+PSO) so as to verify the efficiency of the algorithm in statistical manner. The t-test was invented in 1908 by William Sealy Gosset, an Irish chemist who used it to monitor quality of beverage for the factory where he worked [36]. It is generally used to evaluate whether the difference between the results of two sets of data is statistically significant or random. In this

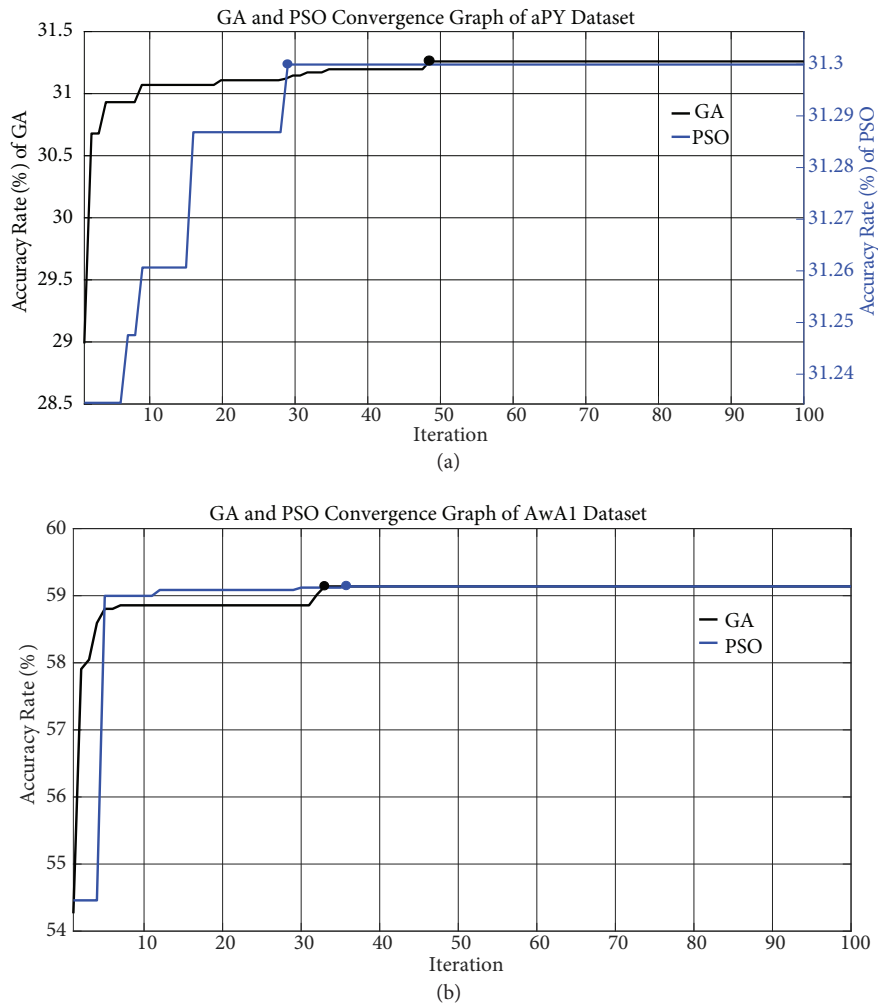


Figure 6. Convergence graph of GA and PSO methods on “aPY”, ”AwA1” datasets (a) “aPY” dataset, (b) “AwA1” dataset.

study, it is applied to measure the influence of GA and PSO approaches to the model. Table 3a shows the t-test results using the accuracy parameter obtained from PSO and GA, Table 3b illustrates the t-test results between the ESZSL model and the GA optimized model (ESZSL+GA) based on abovementioned five datasets, and Table 3c illustrates the t-test results between ESZSL model and PSO (ESZSL+PSO). It should be noted that $P(T \leq t) \ll 0.05$ is accepted during t-tests for all datasets and the results of the ESZSL model are obtained by performing execution.

Once the Table 3a is analyzed, it can be noted that t stat is higher than both the t critical one-tailed value and the two-tailed value for “aPY”, “AwA1”, “AwA2”, and “CUB” datasets. However, t stat value is smaller for “SUN” dataset. This means that the performance differences between the accuracy of the optimized GA model and PSO model are statistically significant for “aPY”, “AwA1”, “AwA2”, and “CUB” datasets. Since the maximum accuracy rates obtained from (ESZSL+GA) and (ESZSL+PSO) models for “SUN” dataset in almost all executions are the same; t test results are obtained close to each other as it is expected. According to the Table 3b, t stat value is higher than both the t critical one-tailed and the two-tailed for all datasets. This means that the performance differences between the ESZSL model and the optimized GA model are statistically

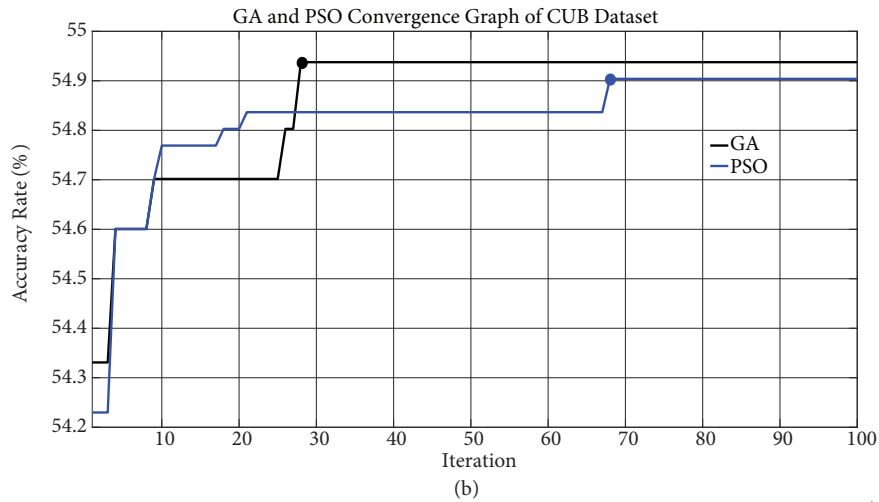
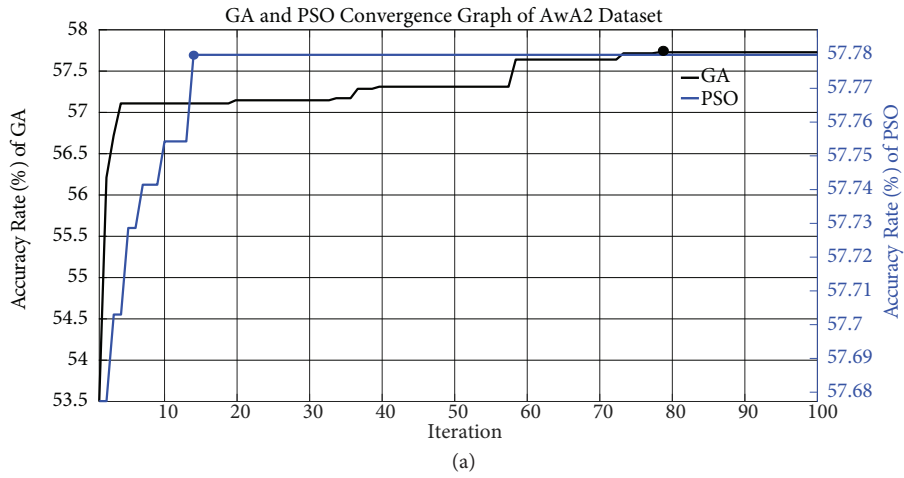


Figure 7. Convergence graph of GA and PSO methods on "AwA2" and "CUB" datasets. (a) "AwA2" dataset, (b) "CUB" dataset.

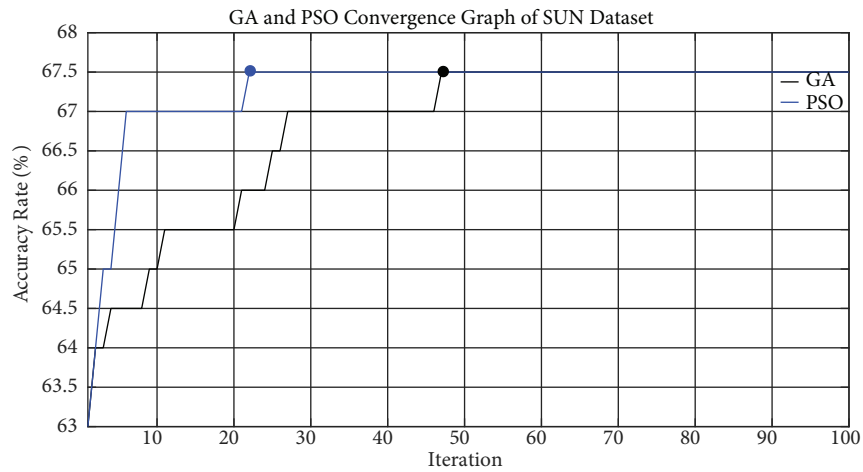


Figure 8. Convergence graph of GA and PSO methods on "SUN" dataset.

Table 3. t-test results (a) t-test results between the optimized PSO model and the optimized GA model; (b) t-test results between the ESZSL model and the GA optimized model; (c) t-test results between the ESZSL model and the PSO optimized model.

(a)

Dataset	t Stat	P(T<=t) one-tail	t Critical one-tail	P(T<=t) two-tail	t Critical two-tail
“aPY”	6.4130	6.17E-05	1.8331	0.000123	2.2622
“AwA1”	3.5196	0.00326	1.8331	0.006519	2.2622
“AwA2”	8.3843	7.59E-06	1.8331	1.52E-05	2.2622
“CUB”	4.3208	0.000966	1.8331	0.001931	2.2622
“SUN”	1	0.171718	1.8331	0.343436	2.2622

(b)

Dataset	t Stat	P(T<=t) one-tail	t Critical one-tail	P(T<=t) two-tail	t Critical two-tail
“aPY”	10.8951	8.72668E-07	1.8331	1.74534E-06	2.2622
“AwA1”	90.1443	6.44805E-15	1.8331	1.28961E-14	2.2622
“AwA2”	29.6115	1.39491E-10	1.8331	2.78983E-10	2.2622
“CUB”	7.2369	2.44244E-05	1.8331	4.88489E-05	2.2622
“SUN”	39.4974	1.06292E-11	1.8331	2.12584E-11	2.2622

(c)

Dataset	t Stat	P(T<=t) one-tail	t Critical one-tail	P(T<=t) two-tail	t Critical two-tail
“aPY”	10.9603	8.29834E-07	1.8331	1.65967E-06	2.2622
“AwA1”	94.2268	4.3298E-15	1.8331	8.6596E-15	2.2622
“AwA2”	30.6532	1.02473E-10	1.8331	2.04946E-10	2.2622
“CUB”	7.52325	1.80191E-05	1.8331	3.60383E-05	2.2622
“SUN”	31.6593	7.68135E-11	1.8331	1.53627E-10	2.2622

significant. $P(T \leq t)$ values for all datasets are very close to zero, which shows that the evaluation confidence is high. Table 3c presents a similar result with Table 3b that t stat value is higher than both the t critical one-tailed and the t critical two-tailed for all datasets. Accuracy differences between the ESZSL model and the optimized PSO model are statistically significant. In this table, it can be seen that $P(T \leq t)$ values for all datasets are very close to zero as well. This indicate that the confidence of the evaluation is higher than 99%. Consequently, the proposed approach provides a significant increase in the accuracy parameter, and it is proved that the evaluation confidence is high.

4. Conclusion

Automatic classification is one of the significant problems discussed in machine learning; thus, it has been studied extensively. Various approaches have been proposed to address this problem. One of the important difficulties encountered in the automatic classification is the appearance of a new class after the training phase.

This is the main reason motivates researchers to focus on ZSL approaches to handle this problem. One of the notable approach used by the ZSL domain is the one-line linear solution ESZSL, mainly involving two parameters that affect the accuracy of the model completely. Hence, parameter selection is a crucial step for this algorithm. It should be noted that one of the critical contribution of this study is to apply leading metaheuristic algorithms to optimize corresponding parameters for a better overall training performance. Benchmark datasets are employed for validation. According to the results, models optimized with GA and PSO approaches show better performance for all datasets. Results also reveal that despite its simplicity, PSO achieves more successful results than GA in most datasets for this problem. T-test is also applied to compare the base model and proposed models. Consequently, the differences in performances were found to be statistically important. It should also be noted that the configuration of the computer used during the experimental process are insufficient to run such large datasets, since it takes long time to execute the codes especially during the training phase. Consequently, population size and the number of iteration are set to a low value for GA and PSO approaches. Results encourage authors to perform new experiments by using more powerful computers for future studies. This will allow to generate larger population size with more iterations. Besides, authors are planning to adapt different optimization algorithms to this problem in order to obtain better results.

References

- [1] Romera-Paredes B, Torr P. An embarrassingly simple approach to zero-shot learning. In: International Conference on Machine Learning; Lille, France; 2015. pp. 2152-2161.
- [2] Li J, Jing M, Lu K, Ding Z, Zhu L et al. Leveraging the invariant side of generative zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. pp. 7402-7411.
- [3] Ding Z, Shao M, Fu Y. Low-rank embedded ensemble semantic dictionary for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. pp. 2050-2058
- [4] Zhang L, Xiang T, Gong S. Learning a deep embedding model for zero-shot learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. pp. 2021-2030.
- [5] Xian Y, Lorenz T, Schiele B, Akata Z. Feature generating networks for zero-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. pp. 5542-5551.
- [6] Mishra A, Krishna RS, Mittal A, Murthy HA. A generative model for zero shot learning using conditional variational autoencoders. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops; 2018. pp. 2188-2196.
- [7] Dietterich TG, Bakiri G. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research* 1994; 2: pp. 263-286. doi: 10.1613/jair.105
- [8] Bart E, Ullman S. Cross-generalization: Learning novel classes from a single example by feature replacement. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); 2005. pp. 672-679.
- [9] Lampert CH, Nickisch H, Harmeling S. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2013; (36)3: pp. 453-465. doi: 10.1109/TPAMI.2013.140
- [10] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems; 2012. pp. 1097-1105.
- [11] Soysal OA, Guzel MS. An introduction to zero-shot learning: an essential review. In: 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA); Ankara, Turkey; 2020. pp. 1-4.
- [12] Kankuekul P, Kawewong A, Tangruamsub S, Hasegawa O. Online incremental attribute-based zero-shot learning. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition; 2012. pp. 3657-3664.

- [13] Norouzi M, Mikolov T, Bengio S, Singer Y, Shlens J et al. Zero-shot learning by convex combination of semantic embeddings. arXiv preprint arXiv:1312.5650 2013.
- [14] Jayaraman D, Grauman K. Zero-shot recognition with unreliable attributes. In: *Advances in Neural Information Processing Systems*; 2014. pp. 3464-3472.
- [15] Sumbul G, Cinbis RG, Aksoy S. Fine-grained object recognition and zero-shot learning in remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing* 2017; 56(2): 770-779. doi: 10.1109/TGRS.2017.2754648.
- [16] Lawrence ND, Platt JC. Learning to learn with the informative vector machine. In: *Proceedings of The Twenty-first International Conference on Machine Learning*; 2004. pp. 65.
- [17] Croonenborghs T, Driessens K, Bruynooghe M. Learning relational options for inductive transfer in relational reinforcement learning. In: *International Conference on Inductive Logic Programming*; 2007. pp. 88-97.
- [18] Raykar VC, Krishnapuram B, Bi J, Dundar M, Rao RB. Bayesian multiple instance learning: automatic feature selection and inductive transfer. In: *Proceedings of The 25th International Conference on Machine Learning*; 2008. pp. 808-815.
- [19] Rückert U, Kramer S. Kernel-based inductive transfer. In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*; 2008. pp. 220-233.
- [20] Akata Z, Perronnin F, Harchaoui Z, Schmid C. Label-embedding for attribute-based classification. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2013. pp. 819-826.
- [21] Lampert CH, Nickisch H, Harmeling S. Learning to detect unseen object classes by between-class attribute transfer. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*; 2009. pp. 951-958.
- [22] Patterson G, Hays J. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*; 2012. pp. 2751-2758.
- [23] Farhadi A, Endres I, Hoiem D, Forsyth D. Describing objects by their attributes. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*; 2009. pp. 1778-1785.
- [24] Wah C, Branson S, Welinder P, Perona P, Belongie S. The caltech-ucsd birds-200-2011 dataset; 2011.
- [25] Xian Y, Lampert CH, Schiele B, Akata Z. Zero-shot learning—A comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2018; 41(9): 2251-2265. doi: 10.1109/TPAMI.2018.2857768.
- [26] Holland JH et al. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT press, 1992.
- [27] ÖZSARI Ş, UĞUZ H, ÇAY T. Interview in land consolidation using genetic algorithm. *Communications Faculty of Sciences University of Ankara Series A2-A3* 2018; 60 (2): 129-146. doi: 10.1501/commua1-2_0000000119.
- [28] Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*; 1995. pp. 1942-1948.
- [29] Shi Y, Eberhart RC. Parameter selection in particle swarm optimization. In: *International Conference on Evolutionary Programming*; 1998. pp. 591-600.
- [30] Sengupta S, Basak S, Peters RA. Particle swarm optimization: a survey of historical and recent developments with hybridization perspectives. *Machine Learning and Knowledge Extraction* 2019; 1(1): 157-191. doi: 10.3390/make1010010.
- [31] Güzel MS, Kara M, Beyazkılıç MS. An adaptive framework for mobile robot navigation. *Adaptive Behavior* 2017; 25(1): 30-39. doi: 10.1177/1059712316685875.
- [32] Neoh SC, Morad N, Marzuki A, Lim CP, Aziz ZA. A multi-resolution GA-PSO layered encoding cascade optimization model. In: *Innovations in Swarm Intelligence 248th ed*. Berlin, Heidelberg, Germany: Springer, 2009, pp. 121-140.

- [33] Xie B, Chen S, Liu F. Biclustering of gene expression data using PSO-GA hybrid. In: 2007 1st International Conference on Bioinformatics and Biomedical Engineering; 2007. pp. 302-305.
- [34] Robinson J, Rahmat-Samii Y. Particle swarm optimization in electromagnetics. IEEE transactions on antennas and propagation 2004; 52(2): 397-407. doi: 10.1109/TAP.2004.823969.
- [35] Gidemen G, Furat M. PID denetleyici optimizasyonu Üzerine uygulaması. In: 2017 International Artificial Intelligence and Data Processing Symposium (IDAP); 2017. pp. 1-6.
- [36] Simon D. Evolutionary Optimization Algorithms. Hoboken, NJ, USA: John Wiley & Sons, 2013.