



Research article

Convolutional neural network with group theory and random selection particle swarm optimizer for enhancing cancer image classification

Kun Lan^{1,2}, Gloria Li^{1,2}, Yang Jie^{1,2}, Rui Tang³, Liansheng Liu^{4,*} and Simon Fong^{1,2}

¹ Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau 999078, China

² DACC Laboratory, Zhuhai Institutes of Advanced Technology of the Chinese Academy of Sciences, Zhuhai 519080, China

³ Department of Management and Science and Information System, Faculty of Management and Economics, Kunming University of Science and Technology, Kunming 650093, China

⁴ Department of Medical Imaging, First Affiliated Hospital of Guangzhou University of Chinese Medicine, Guangzhou 510405, China

* **Correspondence:** Email: ljsjnu@sina.com; Tel: +8602036598876.

Abstract: As an epitome of deep learning, convolutional neural network (CNN) has shown its advantages in solving many real-world problems. Successful CNN applications on medical prognosis and diagnosis have been achieved in recent years. Their common goal is to recognize the insights from the subtle details from medical images by building a suitable CNN model with maximum accuracy and minimum error. The CNN performance is extremely sensitive to the parameter tuning for any given network structure. To approach this concern, a novel self-tuning CNN model is proposed with a significant characteristic of having a metaheuristic-based optimizer. The most optimal set of parameters is often found via our proposed method, namely group theory and random selection-based particle swarm optimization (GTRS-PSO). The insights of symmetric essentials of model structure and parameter correlation are extracted, followed by the hierarchical partitioning of parameter space, and four operators on those partitions are designed for moving neighborhoods and formulating the swarm topology accordingly. The parameters are updated by a random selection strategy at each interval of partitions during the search process. Preliminary experiments over two radiology image datasets: breast cancer and lung cancer, are conducted for a comprehensive comparison of GTRS-PSO versus other optimization algorithms. The results show that CNN with GTRS-PSO optimizer can achieve the best performance for cancer image classifications, especially when there are symmetric components inside the data properties and model structures.

Keywords: convolutional neural network; metaheuristic; particle swarm optimization; group theory; random selection; symmetry; breast and lung cancer; image classification

1. Introduction

Catastrophic illnesses including breast and lung cancers are the most leading factors among all global deaths in the past decade [1]. They cause the most top categories of severe diseases by affecting those individuals in urban and rural livings of both developed and developing countries [2]. It is very obvious that the trend of morbidity increases among young people gradually every year, and the reasons why these deadly sicknesses happen include but not limited to side effects of medication and genetics, stress, obesity, unhealthy diets and improper lifestyles, etc. Although there is still some doubt whether cancer screening is beneficial or harmful for patients, it is critical to detect illnesses in early stage for treating and enhancing the survival of cancers and early diagnosis and screening are two common clinical measures for most prevalent cancers [3,4]. The risks of death are decreasing to a low degree by taking early diagnosis, reducing difficulties of barriers and improving access to cancer medical services. Besides that, even if the diseases are confirmed, it is still vital to detect features of such illnesses by different techniques during the physical examination.

In most of these cases, the physical examination for inspecting suspicious masses on tissues is usually conducted by the experimented radiologist or medical specialist with the respective domain knowledge [5]. However, it is easy to omit the subtle information from detection due to various reasons, such as irregular and fuzzy masses, certain angle orientation and focus of the subject on nature, human errors, etc. And this also causes the occurrence of misjudgments consequently, which makes the early detection very difficult for visual determination on whether the patient is normal or otherwise. Hence, it is always desirable for medical practitioners to have computer-aided detection and diagnosis (CAD) that can offer an objective assessment based on the results of data processing [6]. The functioning strategy of the CAD system is mainly to use some computer techniques to detect the pathological changes in potential abnormal regions, highlight the suspicious components that may be ignored by naked human eyes and inform the medical experts to spot them for the further study and diagnosis.

As the crucial part of the artificial intelligence (AI) fields, deep learning (DL) acts as a collection of intelligent algorithms that attempt to provide the high-level presentations of data features using multiple processing layers containing complex structures with many nonlinear transformations and obtain the relatively accurate results through such a model of classification or prediction instead of manual diagnosis. As a famous member of deep learning family, the convolutional neural network (CNN) would be able to extract multiple data features from different underlying levels by convoluting, nonlinear mapping, pooling and fully connecting the feature details from raw to fine. Distinguished from the conventional structure of artificial neural network (ANN) with full connections of all layered neurons in machine learning, the convolutional layer and the previous layer are linked in a unique way by local connection and weight sharing, their purposes are to reduce the input of dimensions, number of parameters and complexity of entire network architecture. Additionally, these two main strategies can impose the robustness and prevent over-fitting of the model effectively. Furthermore, based on those prominent essentials of CNN, we explore the internal characteristics of symmetries inside the model from the aspects of architecture construction and parameter correlation to get the deeper understanding of CNN.

In order to analyze the symmetry, the brief concept of group theory is introduced hereby. Group theory is the study of algebraic structures known as groups. Groups are special sets equipped with an operation (like multiplication, addition, or composition) that satisfies certain fundamental axioms of properties. In reality, real-world applications from many disciplines can be modeled by group theory (e.g., magic cube, crystals, hydrogen atom system, polynomial roots, cryptography, etc.), especially for those who have the symmetric characteristics of the whole system. Group theory usually embodies the internal symmetry of some structures in the form of automorphism groups. The internal symmetry of the structure often exists as an invariant property simultaneously. The combinations of these operations based on group theory can decompose and reconstruct the components of the entire system and help find the more intrinsic and underlying formation of system structure.

Nevertheless, a critical problem resides in CNN model associating with its technical limitations, its performance highly relies on parameter configuration when the model structure is determined during the process of learning. So, it is not parameter-free and needs a specific strategy for parameter initialization and learning because the value range of those variables may vary from relatively tiny to extremely huge from the continuous parameter space. As a consequence, CNN is quite sensitive to parameter configuration, while changing the parameter values slightly could lead to totally different results probably, and the inaccurate model would become very fatal to a patient in some important assessment and recommendation of clinical tasks.

Thereby, the optimization of CNN parameters emerges significantly and under this circumstance, we propose a metaheuristic-based optimization mechanism for solving parameter learning problem and tuning CNN model to its tip-top state with the highest performance. The parameters that are being optimized differ from hyperparameters (related to network architecture, e.g., convolutional filter size, stride step, pooling dimension, number of feature maps, learning rate, stacked layer design, etc.) in the model, they are those weights and biases of neurons connecting to each other of different layers within the architecture, and they are also learnable when the training process is started while hyperparameters are usually predefined before that. The critical component of CNN optimizer is presented as the group theory and random selection-based particle swarm optimization (GTRS-PSO). According to the symmetry analysis of both local connection and weight sharing in CNN, the symmetric group is adopted for particle encoding with the discretization of continuous values into several discrete intervals from the parameter space. And then, the property of parameter space is studied and it is further divided into four hierarchical partitions called conjugacy class, cyclic form, orbital plane and orbit based on group theory on those discrete intervals. Next, four operators are designed to search and move the neighborhoods on each level of the hierarchical partitions, providing a balanced relation between exploitation and exploration for particle update. At last, the formulating of swarm topology with the random selection from updated permutations of those intervals is carried out to ensure that the search process is proceeding into the next iteration gradually.

The highlights of CNN with GTRS-PSO are listed as follows:

- Symmetry of CNN model is revealed via parameter correlation and model structure studies;
- Parameter space is decomposed into four hierarchical partitions exhaustively and exclusively by discretization of intervals, so the landscape of solutions becomes more explicit;
- Function composition-based operators on those layered partitions are defined for neighborhood movements, and the swarm topology with random selection within discrete intervals is designed for the complete search procedure;
- A tradeoff between diversification (exploration) and intensification (exploitation) is kept and the

computational complexity is reduced due to the utilization of group theory.

Many researchers are dedicating themselves to the works related to CNN optimization. Sahiner et al. [7] are the pioneers who attempt to solve the mammography analysis problem by the three layers of local texture features in CNN. After that, lots of progress works have been done to follow up with the significant meaning of CNN research. A customized CNN with intensive dropout and input distortion techniques is presented by Li et al. [8] to avoid over-fitting for classifying lung disease. A research paper by Kooi et al. [9] shows the superior result generated by CNN compared to other state-of-the-art CAD techniques. Although CNN has performed brilliantly in medical image related tasks, it is realized that we need further optimization for fast diagnosis and better decisions. Some ordinary methods such as Bayesian are deployed in hyperparameter and parameter optimization tasks of a neural network [10]. The most used approaches of the CNN optimizer are optimizations based on gradient descent, examples can be found in batch gradient descent (BGD), stochastic gradient descent (SGD) and mini-batch gradient descent (MBGD), or with the transfer learning of parameter fine-tuning [11]. The technology of automated machine learning (AutoML) is a promising solution for the construction of a deep model without human assistance [12], and as far as the architecture optimization is concerned, the grid and random methods are very practical and efficient to search the hyperparameter space [13].

Metaheuristics are also combined with CNN optimization as a common application. The EvoNets proposed by Liu et al. [14] is an evolutionary approach with genetic algorithm (GA) to search the optimal architecture of the CNN model for best performance. Their work is further improved by Parsa et al. [15] with three different schemes of GA (steady state, generational and elitism) to optimize CNN from the same point of view. Pawelczyk et al. [16] make the improved version of GA via the assistance of gradient learning during the search process to optimize the parameters from the genetically trained deep neural network. Silva et al. [17] reduce the false positive rate of lung image classification with conventional particle swarm optimization (PSO) when optimizing hyperparameters of CNN. Ribalta et al. [18] propose a PSO-based optimization method for hyperparameter selection in deep neural networks such like LeNet and SimpleNet. They show that PSO can allow a deep neural network of a minimal structure to obtain the good performance compared to other methods. Lan et al. [19] optimize CNN parameters using PSO with Lévy flight and long-tailed distribution to make a balance between local and global search. Also a confidence function-based PSO can extract appropriate information from normal distribution knowledge, and the CNN model is tested using a fast linear prediction to minimize the fitness function score [20]. In the reports of research works [21], the simulated annealing (SA) acts as the optimizer of solution vectors, hyperparameters or parameters in CNN to accelerate training process while preserving solution quality. SA and its two variants named macrocanonical annealing (MA) and threshold accepting (TA) are tested and compared to optimize the network model [22], the common thing of these three methods is that they all belong the single-solution optimization approach. The strategy is to select the best value of objective function on the last layer of the model, and then update the weights and biases on the previous layer. Rosa et al. [23] optimize CNN hyperparameters using harmony search (HS) for detecting patterns in one dimensional respiratory data. And many variants of HS, like improved HS, global-best HS, self-adaptive HS, are also involved in the CNN optimization tasks then.

For symmetry analysis in CNN, Gens et al. [24] introduce SymNets as the deep symmetry structure for the generalized form. They instantiate the model with the affine group including rotation, scaling and shearing operations to handle parameters in high dimensional symmetry space. The training and optimization method is backpropagation with partial derivative of loss function. The

paperwork of Vijay et al. [25] reveals the scale invariance of CNN weights and solves the problem of weight space symmetry by constraining the convolutional filters on the unit-norm manifold. SGD works on that manifold as the optimizing strategy of the model.

Table 1. Summary and comparison of different parameter optimization approaches in CNN.

	Gradient descent	Grid and random	Single solution	Population
Theoretical background	Gradient descent optimization	Grid decomposition and random access	Single candidate optimization in metaheuristic	Swarm intelligence in metaheuristic
Typical examples	SGD, BGD, MBGD, Adam	GS, RS	SA, MA, TA	GA, PSO, HS
Advantages	Deterministic to guarantee the global optimum, easy to implement and fast to execute	Competitive to optimize baseline, high reproducible ability, easy to implement and fast to execute	Heuristic to search the global optimum, concentrate on single solution quality and update strategy, high intensification of solutions	Heuristic to search the global optimum, concentrate on population strategy and agent behavior, high diversity of solutions
Disadvantages	Difficulty with escaping local optimum, gradients vanishing and exploding	Slow convergence, extremely random quality of a solution	Low diversification of solutions, high computational cost for complex problem	High computational cost for complex problem, tradeoff between diversification and intensification

To sum up briefly about the reviewed literature in Table 1, we find that most majority of works are focused on architecture or hyperparameter optimization in CNN, few of them are devoted to parameter or weight optimization of CNN model. The gradient descent-based fine-tuning algorithms would encounter serious problems of gradient exploding or vanishing during backpropagation learning, while extremely large or small multiplied values of weights can lead to a very unstable model and get stuck in the suboptimal state of parameter space. If the learning rate is too low, the algorithm becomes very hard to converge. On the contrary, the high rate would make the search skip the optimum and vibrate around it. Similar to the gradient descent-based methods, the grid and random search is also a stochastic approach of optimization and it would have the same issues with gradient descent, such as the difficulty with escaping local optimum and slow convergence, which can lead to the low quality of solutions. Therefore, the metaheuristics are introduced to improve the optimization strategy and overcome the disadvantages mentioned above. There are various types of metaheuristics, although some are also applied to optimize the CNN parameters, many of them are single-solution approaches and have limited performance compared to population-based ones. The drawbacks of single solution-based metaheuristics are mainly about the lack of low diversity of candidates and this would also potentially cause the fact that the global optimum may not reach and it may get stuck in local optimum. Meanwhile, the population-based metaheuristics can overcome these weaknesses by adopting the swarm intelligence strategy and boosting the diversity of solutions from the search space. One of its main challenges is to maintain the tradeoff or balanced relationship between diversification and intensification during the search process. Also, the key challenge in SymNets is that it is intractable to

maintain the explicit representation when extending original space to affine space, and the high dimensional feature maps are required to compute the symmetry. The situation is also true for the steepest gradient descent update of weights on the manifold because of expensive computation.

In this research work, our contribution is to present a group theoretic PSO as the CNN optimizer, it is one of the population-based swarm intelligent metaheuristics and can encode solutions of particles in the form of symmetric groups, which is more intuitive than affine group and manifold and easy to implement. It extracts the weight symmetry without a much larger scale of original solution space, so the computational time complexity remains constant with the CNN architecture as well. Group theory also offers a powerful tool to study the properties of parameter space systematically by hierarchical partitions, and the integrated PSO would be able to escape from the suboptimal state and make the search balanced between diversification and intensification in terms of the dynamic swarm topology with hierarchical operators on those partitions.

The remainder of this paper is organized as below: Section 2 describes the symmetry in CNN model, the mechanism of optimizer of CNN parameters and the design idea of integrated methodology of GTRS-PSO as the CNN optimizer based on particle representation, space landscape, neighborhood movement, swarm topology and random selection. The materials for some preliminary experiments are introduced as well. Section 3 shows the experimental results compared to other optimization algorithms embedded in CNN model. Section 4 discusses the results, gives the reasons behind its superior performances and analyzes the time complexity of GTRS-PSO optimizer. Section 5 concludes the paper. Some related definitions and concepts in group theory used in this paper are listed out in the last section of Supplementary.

2. Materials and methods

In this section, we start to describe the details of design principles of our proposed method to incorporate with the analysis of symmetries of parameters and the study of metaheuristic-based optimizer structure of CNN. The experimental materials are followed by then, which involves data visualization, hyperparameter configuration and computational environment.

2.1. CNN symmetry

As mentioned in the introduction, two main highlighted features of the convolutional layer in CNN are local connection and weight sharing. Derived from the cortical structure of human brain, the idea of local perceptual strategy is designed for establishing the link between convolutional filter in the current layer and some regions of nodes from the previous layer, which refers to the fact that only partial nodes of the human vision neurons would react in the process of sensing the external objects. The correlation between nodes with relatively close distance is strong, and vice versa. This local correlation theory is also applicable to image classification tasks. It can be seen from Figure 1 that nodes in the certain region of the input image are connected to the filter to receive partial information, and the complete information is aggregated after the whole process of convolution. As for weight sharing, during the sliding of convolutional layer upon the image with multiple input channels, all the regions of the nodes share the same collection of weight coefficients. The statistical properties extracted by the same convolutional filter are similar at various locations of the image, only distinguishing filters will correspond to different weight parameters to detect extra

features within that image.

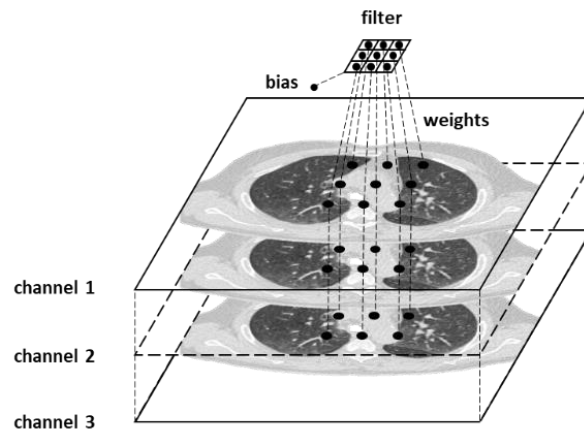


Figure 1. Symmetry of local connection and weight sharing in CNN with multiple image channels.

Basically, we suppose that all weights inside a single filter have the same range of value intervals based on the aforementioned assumptions, thus considering the weight reparameterization of $m \times n$ matrix in a filter:

$$w = [w_1 \quad w_2 \quad \cdots \quad w_n] = \begin{bmatrix} I_{11} & I_{12} & \cdots & I_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ I_{m1} & I_{m2} & \cdots & I_{mn} \end{bmatrix} \quad (1)$$

where n is the size of weight dimension and m is number of value intervals of one particular weight w_i . This matrix stands for the weight space with symmetric representation, and similarly, the bias is reorganized as the form of $m \times 1$ vector:

$$b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \quad (2)$$

2.2. CNN optimizer

Since this research work is aiming at the effort of GTRS-PSO optimizer, so LeNet with the simplest structure is chosen as the demonstration of CNN parameter optimization. Figure 2 illustrates its structure with feature extraction (two convolutional layers c_1 and c_2 , two subsampling layers s_1 and s_2 , two non-linear ReLU rectifications r_1 and r_2) and classification (two fully connected layers n_1 and n_2 ,) parts. Notice that the red numbers are hyperparameters predefined before training and the sum of those blue ones are the parameter dimension of both weights and biases of the entire model. The black numbers are determined by the size of the input image.

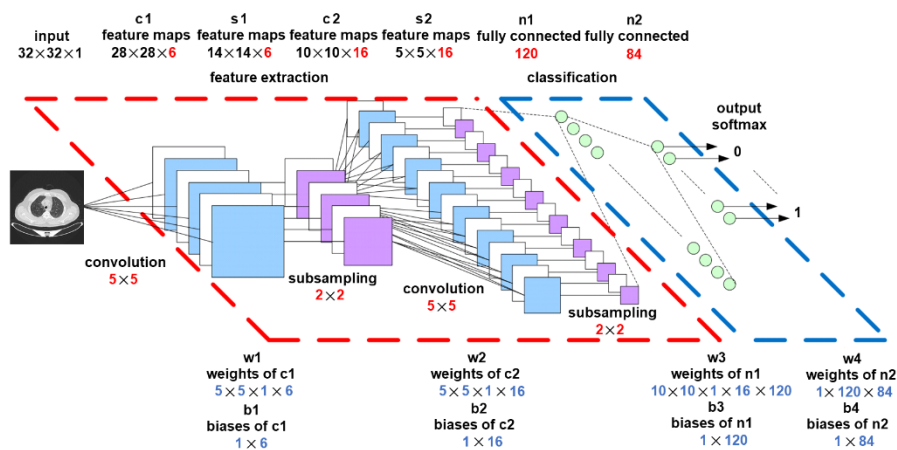


Figure 2. Architecture of LeNet as a standard CNN model.

Figure 3 demonstrates two optimization approaches of CNN, they are traditional gradient descent in the top right and population-based metaheuristic in the bottom right. The fatal issue of gradient descent encountered during backpropagation is that the model has been trapped into suboptimal state and the gradient may be vanishing or exploding. Then our motivation is to replace it with the population-based metaheuristic, and the proposed GTRS-PSO can avoid this issue through multiple search agents called particles and appropriate operators to move the neighborhoods and search the space in equilibrium between diversification (exploration) and intensification (exploitation).

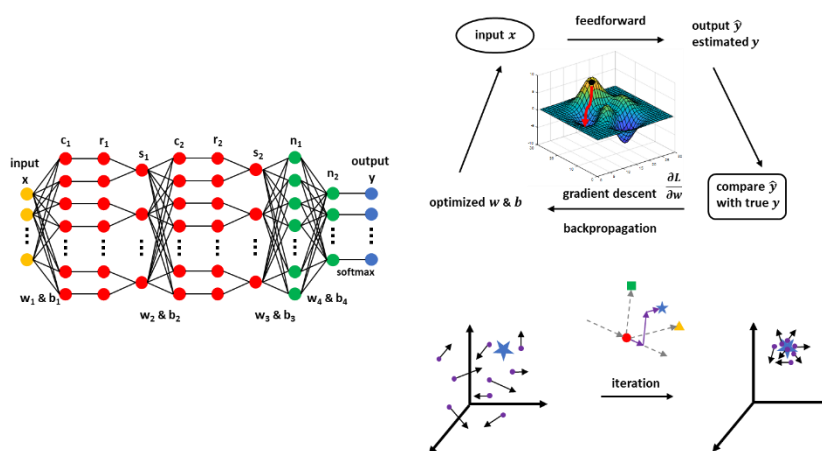


Figure 3. Optimization strategies of a standard CNN model implemented by traditional gradient descent and population-based metaheuristic.

The objective fitness of the optimization is the cross-entropy as the loss function, it is a widely used criterion to evaluate the extent of how an algorithm works to fit with the original inputs.

$$l_i = -\log \left(\frac{e^{f y_i}}{\sum_j e^{f_i}} \right) \tag{3}$$

where f_i is the array of predicted class scores for every single instance in the label class, y_i is one-hot encoding value of that label class, and l_i is the loss value of cross-entropy computed by the output

of the SoftMax classifier. The fraction after the log is the normalized probability of the correct prediction. Furthermore, the regularization with coefficient λ is added to evaluate the model more reliably and prevent over-fitting.

$$L = \frac{1}{N} \sum_i l_i + \frac{1}{2} \lambda \sum_k \sum_l w_{k,l}^2 \quad (4)$$

2.3. GTRS-PSO as CNN optimizer

As a unifying mathematical framework for population-based metaheuristic of CNN optimizer, group theory and random selection are integrated with PSO to form up the search of parameters in this subsection. We use group theory to redesign PSO from four different aspects, namely particle representation of intervals, landscape of parameter space, neighborhood movement on defined operators and dynamic swarm topology. The random value is selected from each interval after the update of particle velocity, and it works together with the trend of both local and global optima to update the position of that particle at the next iteration.

2.3.1. Particle representation

The essentials of symmetry in CNN network structure lead to effective representation of the particle by the symmetric group. According to the definitions of permutation and symmetric group, the encoding is a mapping function composed of the matrix with two rows that maps the index in the top to its corresponding value at the bottom.

$$\begin{aligned} p &= \begin{pmatrix} I \\ p(I) \end{pmatrix} = \begin{pmatrix} I_1 & \cdots & I_i & \cdots & I_n \\ p(I_1) & \cdots & p(I_i) & \cdots & p(I_n) \end{pmatrix} \\ &= (p(I_1) \ p(I_2) \ \cdots)(\cdots \ p(I_{i-1}) \ p(I_i) \ \cdots)(\cdots \ p(I_{n-1}) \ p(I_n)) \end{aligned} \quad (5)$$

The form of matrix is sometimes abbreviated to one row vector with only image values inside several pairs of parentheses. The discrete intervals are those elements of the mapping from the parameter space and the permutation p stands for the transformation from one state to another.

2.3.2. Solution landscape

The structure analysis of solution landscape can divide the parameter space into hierarchical components of conjugacy classes, cyclic forms, orbital planes and orbits, respectively. Each of them is related to the different level of search, the partition scheme is complete and exhaustive of the space and the efforts are put on the concentration of more effective regions. In Figure 4, an example of space partitioning of a set with eight elements is shown, and the plot with only three dimensions is displayed because of intuitive visualization.

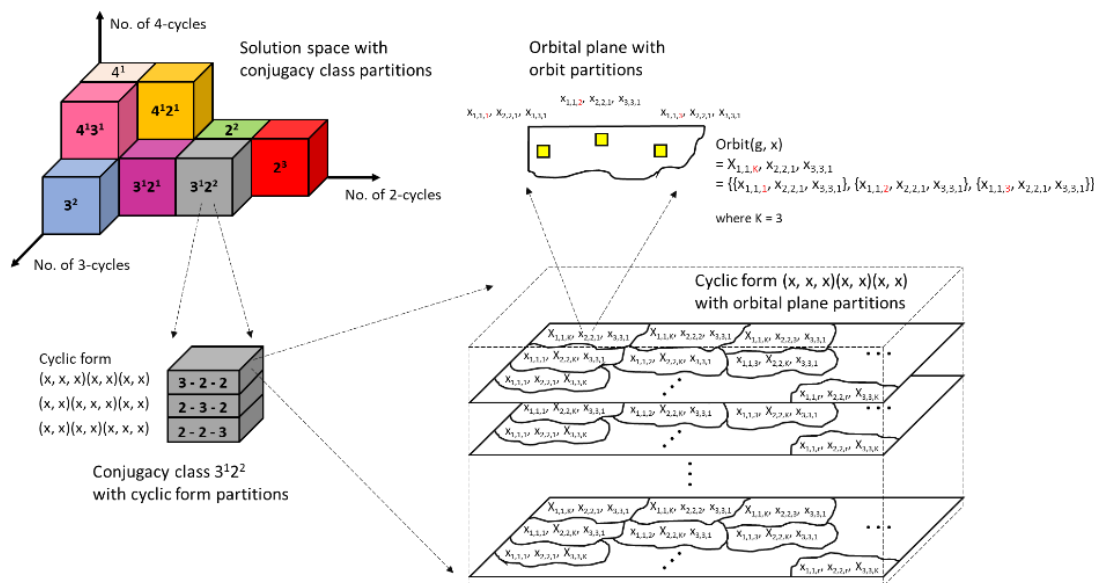


Figure 4. Hierarchical partitioning with four layers of the solution space.

Conjugacy class. The first order partition of hierarchy is the conjugacy class from the solution space. According to its definition, two elements within the same conjugacy class are conjugate to each other, so the structure of a conjugacy class has several cyclic factors, where each of them is a cycle with the length between one and n . Two cycles with the same length are merged in the form similar to exponential notation. A conjugacy class with the structure of multiple cyclic factors of k_i i -cycles is denoted by factor form:

$$conjugacy\ class = 1^{k_1} 2^{k_2} \dots i^{k_i} \dots n^{k_n}, \quad i \in [1, n], k_i \in [0, n] \tag{6}$$

$$\sum_{i=1}^n i k_i = n \tag{7}$$

Cyclic form. The second order partition of hierarchy is the cyclic form of a conjugacy class. According to the definitions of conjugacy class and cycle, the cyclic forms are specified by both length of each cycle and ordering of these cycles, and thus cyclic form differs from each other based on these two metrics. We can determine the relation of two cyclic forms in terms of the permutation of their cyclic factors, the notation of a cyclic form is:

$$cyclic\ form = (x\ x\ \dots)(\dots\ x\ x\ \dots)(\dots\ x\ x) \tag{8}$$

Orbital plane. The third order partition of hierarchy is the orbital plane within a cyclic form. According to the definition, given a set of X with n letters, X is divided into a collection of subsets $\{X_k\}$ using k cyclic forms. If p_1 and p_2 are two permutations of the symmetric group S_n , let $Y_k(p)$ denote the element positions of nontrivial cycles (not unit cycle) of $\{X_k\}$ in alphabetical order, then the orbital plane is the set of partitions of X where the element positions in $\{X_k\}$ are the same under two group actions of p_1 and p_2 . The associated equivalence relation for this definition is:

$$orbital\ plane = \{X_k\}, \quad iff\ X_i(p_1) = X_i(p_2), \forall i \in [1, k] \tag{9}$$

Orbit. The last order partition of hierarchy is the orbit per orbital plane. According to the definitions of group action and orbit, the orbit is the collection of all elements in a given set X to which its subset x can be moved by group action g from a given group G :

$$\text{orbit} = \text{orbit}(g, x) = \{gx | g \in G\}, \quad \forall x \in X \quad (10)$$

As far as the parameter space of CNN is concerned, the basic element is the discrete value interval of each weight and the purpose of space partitioning is to provide the systematical analysis of solution landscape and the design foundations of neighborhood movements in the following subsection.

2.3.3. Neighborhood movement

In the proposed method, a movement of a particle is defined as a particular group action on the incumbent solution that can change it to a new one. And the neighborhood of a particle is defined as the region of space reached by particular steps of group actions with its center to be the initial position itself. Four operators based on the above hierarchical partitions are presented to form up the velocity update strategy of GTRS-PSO. Let ∇ denote the operator based on various group actions and \otimes denote the group multiplication.

Conjugator. The conjugator jumps neighborhoods of a particle to a different conjugacy class, so it is an inter conjugacy class operator. By its definition, the operation of conjugators promotes global exploration at the early stages or when the process is trapped into the local optimum. For example:

$$\begin{aligned} x &= (1 \ 3 \ 2)(4 \ 6)(5 \ 8)(7)X \in CC_3 = 3^1 2^2 1^1 \\ \nabla_{cnj} \otimes x &= x (2 \ 4) x^{-1} (2 \ 4)^{-1} = (1 \ 6)(2 \ 4)(3)(5)(7)(8)X \in CC_3 = 2^2 1^4 \end{aligned} \quad (11)$$

Cycler. The cycler enables the particle to move across different cyclic form structures inside the same conjugacy class, and it belongs to the operator of the intra conjugacy class, inter cyclic form and has the effort of global exploration at the early and middle stages of the search process. For example:

$$\begin{aligned} x &= (1 \ 3 \ 2)(4 \ 6)(5 \ 8)(7)X \in CF : 3 - 2 - 2, (x \ x \ x)(x \ x)(x \ x) \\ \nabla_{cyc} \otimes x &= (1 \ 4 \ 2) x = (1 \ 3)(2 \ 4 \ 6)(5 \ 8)(7)X \\ &\in CF : 2 - 3 - 2, (x \ x)(x \ x \ x)(x \ x) \end{aligned} \quad (12)$$

Swapper. The swapper changes the contents in each cycle while keeps the structure unchanged within a cyclic form, this is done by taking the conjugation and the result seems to be swapped between two different cycles. The swap operator is intra cyclic form, inter orbital plane and concentrates on the local exploitation at the middle and late stages of the search process. For example:

$$\begin{aligned} x &= (1 \ 3 \ 2)(4 \ 6)(5 \ 8)(7)X \in OP, (1 \ 3 \ 2)(x \ x)(x \ x)(x) \\ \nabla_{swp} \otimes x &= (2 \ 4) x (2 \ 4)^{-1} = (1 \ 3 \ 4)(2 \ 6)(5 \ 8)(7)X \\ &\in OP, (1 \ 3 \ 4)(x \ x)(x \ x)(x) \end{aligned} \quad (13)$$

Traverser. The traverser searches along the orbits one by one from each cycle, just like the traversal on an orbital plane, and obviously it is intra orbital plane. It works to intensify the local exploitation at the late stages of the search process. For example:

$$x = (1 \ 3 \ 2)(4 \ 6)(5 \ 8)(7)X$$

$$\nabla_{trv} \otimes x = orbit((1 \ 3 \ 2), X) = (1 \ 2 \ 3)X, (1 \ 3 \ 2)X, (e)X \tag{14}$$

The mapping relation of parameter space from flattened encoding to hierarchical partitioning is established in Figure 5. Each update step of intervals of the flattened vector is linked to a specific region of hierarchical space with the same dimension, and the update with random selection of intervals maps to the certain point of hierarchical space. Four operators (traverser in yellow, swapper in green, cycler in red and conjugator in blue) take place concurrently to move neighborhoods in the hierarchical space heuristically.

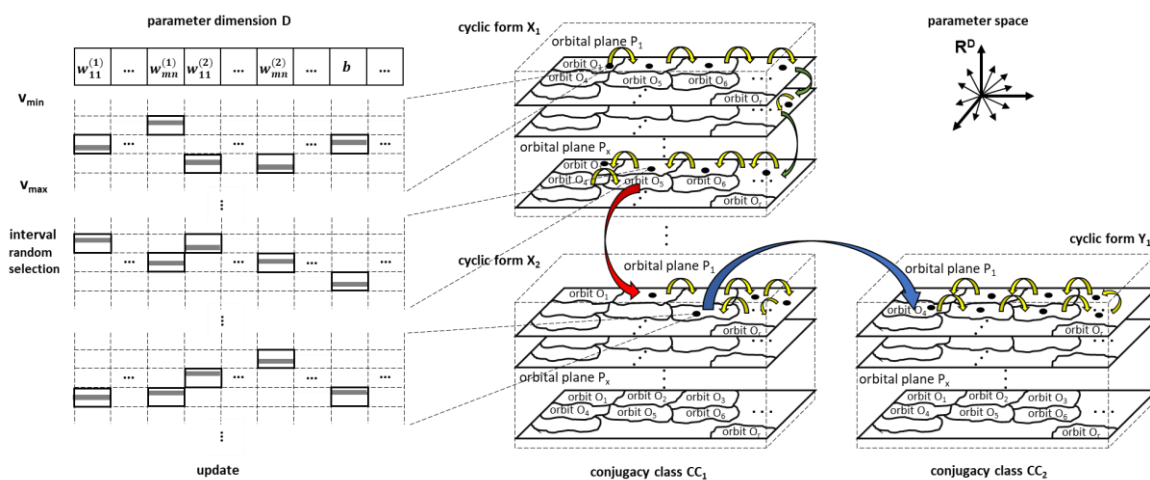


Figure 5. Mapping relation from flattened vector to hierarchical space.

2.3.4. Swarm topology

Preserving the guidance trend of both local and global optima in traditional PSO, GTRS-PSO imposes the inertia item by defined operators instead of purely random search. Let ∇ denote the operator based on various group actions by group multiplication \otimes and its formulas are:

$$\nabla = (r_{cnj} \nabla_{cnj}) \otimes (r_{cyc} \nabla_{cyc}) \otimes (r_{swp} \nabla_{swp}) \otimes (r_{trv} \nabla_{trv}) \tag{15}$$

$$v_{k+1}^i = R(\nabla \otimes v_k^i) + r_1(p_k^i - x_k^i) + r_2(p_k^g - x_k^i) \tag{16}$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i \tag{17}$$

where R is random selection from intervals, k is the count for iterations, i is the index of particles, p_k^i is the local best fitness of i^{th} particle after k iterations and p_k^g is the global best fitness of the entire swarm after k iterations.

2.4. Materials

For validating the proposed GTRS-PSO optimizer of CNN, two empirical datasets of clinical

diagnosis of breast and lung cancers are introduced. The breast dataset is the digital database for screening mammography, and it is the collaborative efforts of Massachusetts General Hospital, Sandia National Laboratories and the University of South Florida Computer Science and Engineering Department. The lung cancer dataset is collected in specialist hospitals of Iraq oncology teaching hospital and their national center for cancer diseases. As well known, the bright and scattered distribution of irregular calcification tissue in a breast MRI or the irregular lobulation sign a lung CT image (Figure 6b,d) is an important clinical symptom to diagnose. This lesion can also be detected by the margin or border of normal tissue and sick area using their significant difference of obscured, circumscribed and speculated texture properties. It is also challenging to identify breast and lung cancers very precisely due to the dense tissue under mammogram screening and the analogous symptoms from an infection or other breast diseases.

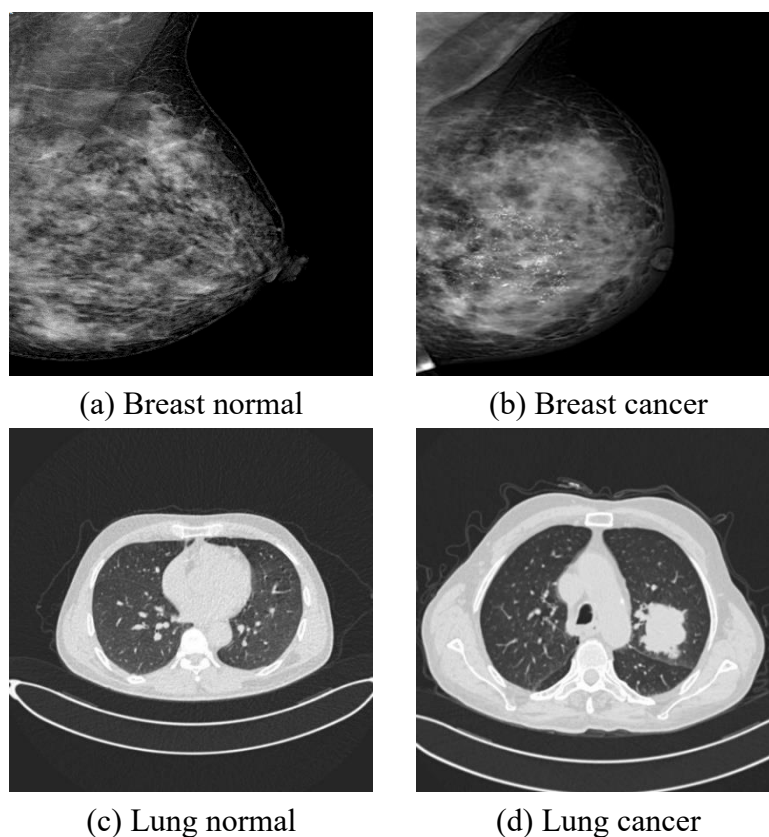


Figure 6. Visualization of breast and lung CT images with binary labels.

Table 2. Data descriptions of breast and lung images.

Dataset	Image no.	Train / Test	Task
Breast	18860	10-fold cross validation	Binary classification (normal and cancer)
Lung	1097	10-fold cross validation	Binary classification (normal and cancer)

The detailed information about dataset descriptions and hyperparameter and parameter settings of CNN and GTRS-PSO is displayed in Tables 2 and 3, respectively. The experiment mainly emphasizes on the evaluation of loss function of cross-entropy of CNN optimizers over a number of

iterations, the targets of optimization are weights and biases in the model, and the compared algorithms are Adam (adaptive moment estimation, which is gradient descent with extra RMSprop and Momentum technologies), RS, SA, GA, PSO, LLT-PSO (our proposed method from previous research and publication) [26] and the proposed GTRS-PSO. Each of them is a typical algorithm from the reviewed literature and summarized methods in Section 1, and the experiment aims at providing the comprehensive comparison of the efforts of various optimization technologies utilized in CNN model of the parameter learning process. The training and testing mechanism is 10-fold cross validation.

Table 3. Hyperparameter setting of CNN and parameter setting of GTRS-PSO.

Hyperparameter	Value	Task
λ	[0.5, 2.0]	Regularization coefficient in Eq (4)
α	10^{-4}	Learning rate of CNN
fs	3	Filter size of CNN
ps	2	Pooling or subsampling rate of CNN
fm	16	Feature map number of CNN
act	ReLU	Activation function of CNN
fc	64	Fully connected layer neuron number of CNN
m	10	Interval number in Eq (1)
r	(0.0, 1.0)	Uniform distributed random number in Eqs (15) and (16)
c	1.0	Acceleration coefficient in Eq (16)
v_{min}, v_{max}	[-2.0, 2.0]	Velocity range in Eq (16)
k	2000	Iteration number in Eqs (16) and (17)
pop	30	Population size in Eqs (16) and (17)

The experiments are conducted under the computational environment of a PC with the hardware equipment of HP EliteDesk 800 G2 Tower with Intel (R) Core (TM) i7-6700 CPU @ 3.40GHz, 16GB RAM and Nvidia GeForce GTX 750. The software programming platform on Ubuntu 16.04 operating system is TensorFlow 1.2.1, Python 3.7.2 and SymPy 1.7.1, which is a Python library of symbolic mathematics for group theory implementation.

3. Results

The experiment is a case study and the explicit performances of CNN models with compared algorithms on breast and lung cancer datasets are tabulated in Table 4. The performances include final results of cross-entropy and computational time after a certain number of iterations. Overall, the fitness values of every different optimizer are all less than 1.0, and GTRS-PSO has the number which is even less than 0.5 on both datasets. Meanwhile, the Adam and RS optimizers wrapped in CNN generate the worst fitness scores, but they all share the fairly less time than others, which are less than 1 hour. The time costs of the rest of the algorithms would vary from almost 300 to 500 minutes and they are much larger than Adam and RS, because their search mechanisms differ. According to the factor of fitness score, it can be inferred that our proposed GTRS-PSO outperforms other optimizers based on fitness value, and its time cost is relatively acceptable and even shorter than LLT-PSO, but more than PSO and other metaheuristic approaches. Figure 7 shows the descending curves of cross-entropy during the

whole optimization process of different methods. All the curves have the trend of decaying and the final fitness score will get converged over some iterations. At first glance, one may realize that the plots of Adam, RS and SA have a different shape compared to other optimizers, it is similar to a smooth and descending curve while others share the shape of stair like curves. Furthermore, there are convexity and concavity in SA plot, on the contrary, Adam and RS plots only contain convex component. All the visual differences are caused by the search mechanisms of multiple optimizers behind.

Table 4. Comparative results of CNN performances with various optimizers on breast and lung datasets.

	Breast		Lung	
	Fitness	Time (min)	Fitness	Time (min)
CNN-Adam	0.518	40.723	0.641	21.485
CNN-RS	0.584	15.285	0.590	12.541
CNN-SA	0.497	391.606	0.564	288.973
CNN-GA	0.501	417.385	0.497	301.245
CNN-PSO	0.478	378.539	0.527	279.782
CNN-LLT-PSO	0.470	446.443	0.503	354.384
CNN-GTRS-PSO	0.440	472.846	0.476	337.456

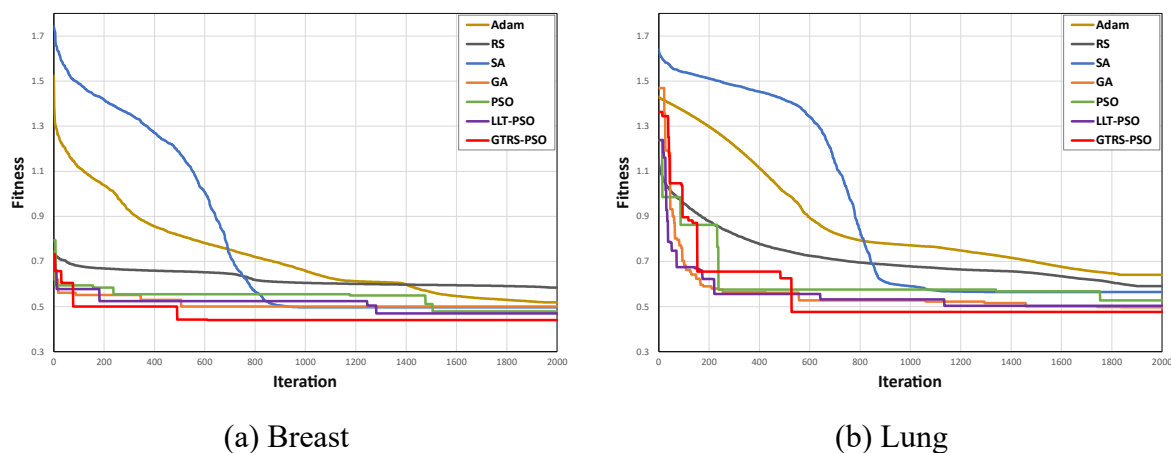


Figure 7. Converting trends of CNN fitness with various optimizers on breast and lung datasets.

4. Discussion

Overall, the proposed GTRS-PSO as the CNN optimizer has the best performance, the curve shape is different from Adam, RS and SA because its search mechanism is metaheuristic style, the fitness score remains fixed and the update of global best value would not take place until a better solution is found, so the shape is a little bit like the slop-down pattern with different levels of stairs. For Adam, it searches according to the gradient, so the whole process works more smoothly and gradually, the situation is also true for RS. As for SA, its curve keeps a high value of fitness in the early stage but suddenly drops to a low level because of the physical phenomenon of annealing during the cooling process. The converging rate of GTRS-PSO is fast on breast dataset but it is slow on lung

dataset as far as PSO and LLT-PSO are concerned, the reason behind this phenomenon is that four operators in Eq (15) are function composition-based and they form up the final update of inertia weight of neighborhood movement in Eq (16), each of them is carried out concurrently with different happening rate controlled by a random number. So in the early stage of the whole process, the search of GTRS-PSO would focus on a broader region of exploration and its converging rate is slower, but after some iterations, it intensifies the search within a narrower area of exploitation and the converging rate becomes faster. From Figure 7 we can also find that the population-based methods would always be initialized with smaller fitness scores than those of single-based, gradient descent-based and random-based methods. The best initial value will be selected among a number of candidates of potential solutions, and in the subsequent search steps, the population can offer the sustainable support to find the global optimum and better exploration of diversities, so they show the faster convergence rate than others as well.

Adam is gradient descent-based method and RS is randomness oriented, they may get stuck in local optimum. And SA would not be able to produce good result because it is single solution-based metaheuristic. The inertia update in PSO is quite random, and it has the strong capability to control the particle behavior via altering mean and standard deviation of the Gaussian distribution to which the search space is linked. The particles near the global best would generate the small standard deviation, so that the neighborhood close to these particles can be searched in the first priority, and this will lead to the early convergence furthermore. Although its converging rate is descending rapidly, the final fitness score is not good enough because of lack of local search or intensification ability. As for LLT-PSO, it is derived initially from conventional PSO with the extra Lévy flight (long-tailed distribution) for partial swarm as leaders, thus its converging curve has the similar shape with PSO and its performance is better than PSO due to the particle ability of Lévy flight of escaping from local optimum. In Eq (16) of GTRS-PSO, the meaning of inertia item ∇ with random selection R is the moving step in search space and the coefficients of local and global items are the area (r_1, r_2) of the reachable region in that space. All these three items are controlled by the dynamic configuration in GTRS-PSO, and a tradeoff between exploration and exploitation is maintained, which causes the result that the search space is decomposed systematically and the vast majority of the space can be searched and the duplicated search space with symmetric components is avoided, hence GTRS-PSO can get the best performance.

Structural complexity of CNN: for a single convolutional layer, it is $O(m^2 * n^2 * c_{in} * c_{out})$, where m is the size of feature map, n is the size of filter, c_{in} is the number of feature maps from the previous layer and c_{out} is the number of feature maps from the current layer. For a single fully connected layer, it is $O(c_{in} * c_{out})$ and it has the same scale as a single convolutional layer because m and n are fixed for specific applications. Totally, the structural complexity of CNN is $O(d * m^2 * n^2 * c_{in} * c_{out})$, where d is the depth of CNN model.

Time complexity of GTRS-PSO: in the single iteration, each operator has the complexity of $O(D)$, where D is the dimension of objective function and it is a linear rearrangement of the sequence of all elements. The objective function evaluating cost is $O(cof)$, the time complexity during the single iteration is $O[N * (D + cof)]$, where N is population number, and total time complexity is $O[k * N * (D + cof)]$, where k is iteration number.

Total complexity of CNN with GTRS-PSO optimizer: the objective function evaluating cost equals the update process of CNN, $O(cof) = O(d * m^2 * n^2 * c_{in} * c_{out})$, and the dimension of objective function is the number of weights and biases in CNN, which has the same scale as CNN

structural complexity. So the total complexity is $O(k * N * D) \leq O(D^2)$.

5. Conclusions

A novel group theoretic particle swarm optimization with random selection is proposed to optimize the CNN parameters of weights and biases. The novelty of this research includes applying the group theory framework combined with PSO search scheme in order to improve the solution landscape analysis and design the corresponding operators that can search for the best neighborhood solutions systematically. The swarm topology keeps the tradeoff for balancing both exploitation and exploration. The symmetries that are based on local connection and weight sharing in CNN are presented and extracted by group theory encoding. A case study on breast and lung cancer datasets is conducted and the preliminary results demonstrate that CNN with GTRS-PSO optimizer can have the optimal performance overall.

Acknowledgments

The authors are thankful for the financial support from the research grants, MYRG2016-00069, entitled ‘Nature-Inspired Computing and Metaheuristics Algorithms for Optimizing Data stream mining Performance’, EF003/FST-FSJ/2019/GSTIC, code no. 201907010001, FDCT/126/2014/A3, entitled ‘A Scalable Data Stream Mining Methodology: Stream-based Holistic Analytics and Reasoning in Parallel’ offered by FDCT and RDAO/FST, the University of Macau and the Macau SAR government.

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. World Health Organization, World health statistics 2020: monitoring health for the SDGs, sustainable development goals, 2020.
2. J. D. Li, G. Chen, M. Wu, Y. Huang, W. Tang, Downregulation of CDC14B in 5218 breast cancer patients: A novel prognosticator for triple-negative breast cancer, *Math. Biosci. Eng.*, **17** (2020), 8152–8181.
3. Y. Ouyang, Z. Zhou, W. Wu, J. Tian, F. Xu, S. Wu, et al., A review of ultrasound detection methods for breast microcalcification, *Math. Biosci. Eng.*, **16** (2019), 1761–1785.
4. M. Zhang, Y. Zhou, Y. Zhang, High expression of TLR2 in the serum of patients with tuberculosis and lung cancer, and can promote the progression of lung cancer, *Math. Biosci. Eng.*, **17** (2020), 1959–1972.
5. G. Sun, T. Zhao, Lung adenocarcinoma pathology stages related gene identification, *Math. Biosci. Eng.*, **17** (2020), 737–746.
6. J. Gao, Q. Jiang, B. Zhou, D. Chen, Convolutional neural networks for computer-aided detection or diagnosis in medical image analysis: an overview, *Math. Biosci. Eng.*, **16** (2019), 6536–6561.

7. B. Sahiner, H. P. Chan, N. Petrick, D. Wei, M. A. Helvie, D. D. Adler, et al., Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images, *IEEE T. Med. Imaging*, **15** (1996), 598–610.
8. Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, M. Chen, Medical image classification with convolutional neural network, in *2014 13th international conference on control automation robotics & vision (ICARCV)*, IEEE, (2014), 844–848.
9. T. Kooi, G. Litjens, B. Van Ginneken, A. Gubern-Mérida, C. I. Sánchez, R. Mann, et al., Large scale deep learning for computer aided detection of mammographic lesions, *Med. Image. Anal.*, **35** (2017), 303–312.
10. J. Snoek, O. Rippel, K. Swersky, R. Kiros, N. Satish, N. Sundaram, et al., Scalable bayesian optimization using deep neural networks, in *International conference on machine learning*, PMLR, (2015), 2171–2180.
11. M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, et al., A state-of-the-art survey on deep learning theory and architectures, *Electronics*, **8** (2019), 292.
12. X. He, K. Zhao, X. Chu, AutoML: A Survey of the State-of-the-Art, *Knowl-Based. Syst.*, **212** (2021), 106622.
13. L. Li, A. Talwalkar, Random search and reproducibility for neural architecture search, in *Uncertainty in Artificial Intelligence*, PMLR, (2020), 367–377.
14. P. Liu, M. D. El Basha, Y. Li, Y. Xiao, P. C. Sanelli, R. Fang, Deep evolutionary networks with expedited genetic algorithms for medical image denoising, *Med. Image. Anal.*, **54** (2019), 306–315.
15. P. Esfahanian, M. Akhavan, Gacnn: Training deep convolutional neural networks with genetic algorithm, preprint, arXiv:1909.13354.
16. K. Pawelczyk, M. Kawulok, J. Nalepa, Genetically-trained deep neural networks, in *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO)*, ACM, (2018), 63–64.
17. G. L. F. da Silva, T. L. A. Valente, A. C. Silva, A. C. de Paiva, M. Gattass, Convolutional neural network-based PSO for lung nodule false positive reduction on CT images, *Comput. Meth. Prog. Bio.*, **162** (2018), 109–118.
18. P. R. Lorenzo, J. Nalepa, M. Kawulok, L. S. Ramos, J. R. Pastor, Particle swarm optimization for hyper-parameter selection in deep neural networks, in *Proceedings of the genetic and evolutionary computation conference (GECCO)*, ACM, (2017), 481–488.
19. K. Lan, D. T. Wang, S. Fong, L. S. Liu, K. K. Wong, N. Dey, A survey of data mining and deep learning in bioinformatics, *J. Med. Syst.*, **42** (2018), 1–20.
20. T. Serizawa, H. Fujita, Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization, preprint, arXiv:2001.05670.
21. J. Tóth, H. Toman, A. Hajdu, Efficient sampling-based energy function evaluation for ensemble optimization using simulated annealing, *Pattern. Recogn.*, **107** (2020), 107510.
22. L. R. Rere, B. A. Wardijono, Y. I. Chandra, A comparison study of three single-solution based metaheuristic optimisation for stacked auto encoder, in *Journal of Physics: Conference Series*, IOP Publishing, (2019), 012066.
23. G. Rosa, J. Papa, A. Marana, W. Scheirer, D. Cox, Fine-tuning convolutional neural networks using harmony search, in *Iberoamerican Congress on Pattern Recognition*, (2015), 683–690.
24. R. Gens, P. M. Domingos, Deep symmetry networks, *Adv. Neur. Inf.*, **27** (2014), 2537–2545.

25. V. Badrinarayanan, B. Mishra, R. Cipolla, Understanding symmetries in deep networks, preprint, arXiv:1511.01029.
26. K. Lan, L. Liu, T. Li, Y. Chen, S. Fong, J. A. L. Marques, et al., Multi-view convolutional neural network with leader and long-tail particle swarm optimizer for enhancing heart disease and breast cancer detection, *Neural. Comput. Appl.*, **32** (2020), 15469–15488.



AIMS Press

©2021 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)