Analyze COVID-19 CT images based on evolutionary algorithm with dynamic searching space

Yunhong Gong · Yanan Sun · Dezhong Peng · Peng Chen · Zhongtai Yan · Ke Yang

Received: 18 March 2021 / Accepted: 17 August 2021 / Published online: 6 September 2021
© The Author(s) 2021

Abstract
The COVID-19 pandemic has caused a global alarm. With the advances in artificial intelligence, the COVID-19 testing capabilities have been greatly expanded, and hospital resources are significantly alleviated. Over the past years, computer vision researches have focused on convolutional neural networks (CNNs), which can significantly improve image analysis ability. However, CNN architectures are usually manually designed with rich expertise that is scarce in practice. Evolutionary algorithms (EAs) can automatically search for the proper CNN architectures and voluntarily optimize the related hyperparameters. The networks searched by EAs can be used to effectively process COVID-19 computed tomography images without expert knowledge and manual setup. In this paper, we propose a novel EA-based algorithm with a dynamic searching space to design the optimal CNN architectures for diagnosing COVID-19 before the pathogenic test. The experiments are performed on the COVID-CT data set against a series of state-of-the-art CNN models. The experiments demonstrate that the architecture searched by the proposed EA-based algorithm achieves the best performance yet without any preprocessing operations. Furthermore, we found through experimentation that the intensive use of batch normalization may deteriorate the performance. This contrasts with the common sense approach of manually designing CNN architectures and will help the related experts in handcrafting CNN models to achieve the best performance without any preprocessing operations.

Keywords Evolutionary algorithms · Variable-length chromosomes · COVID-19 · Batch normalization

Introduction
At the beginning of 2020, the outbreak of an atypical and person-to-person transmissible pneumonia caused by severe acute respiratory syndrome coronavirus 2 (SARS-COV-2, also known as COVID-19) spread worldwide. After a year, COVID-19 is still spreading, and there were more than 55.56 million confirmed cases and 1.33 million deaths worldwide as of November 10, 2020 [1]. As a practical and sensitive modality, radiological computed tomography (CT) is widely used to diagnose COVID-19 pneumonia, even in asymptomatic individuals. It is considered a particularly important screening tool to help doctors diagnose patients who have false-negative reverse transcription-polymerase chain reaction results. However, it is difficult for radiologists to distinguish COVID-19 cases from other similar viral
pneumonia CT images, such as SARS. This is because they often overlap with other infectious and inflammatory lung diseases.

At present, data engineering experts are devoting to utilizing artificial intelligence technology to handle the difficult COVID-19 diagnosis task and save more lives [2,3]. With the development of deep learning, thousands of neural network algorithms have been proposed in recent decades. Because of hierarchically abstract representations with local operations, convolutional neural networks (CNNs), as one of the main categories, are widely adopted in computer vision applications. Compared with standard neural networks, the critical advantage of the feature sharing mechanism in CNNs allows them to rely on far fewer weights and variables. By reducing the number of weights and variables that need to be learned, the complexity of the network is decreased, and the image generalization ability is improved. Moreover, pooling operations allow CNN architectures to be less affected by small variations in position. CNNs have been successfully applied to object detection [4], biological image classification [5], cross-modal retrieval [6] and other visual recognition tasks. Since LeNet-5 [68] was proposed in 1989, a series of CNNs variants have been developed, such as AlexNet [7] which uses dropout [8] to avoid overfitting, VGGNet [9], which trains a deeper neural network by $3 \times 3$ filters throughout the network for considerable scale image recognition, and ResNet [10] whose main contribution lies in its reliance on residual learning by skip connections. Due to the enormous efforts of medical experts and the careful designs of statistical researchers, significant progress has been made to automatically interpret CT images and predict whether or not the CTs are positive for COVID-19 through deep learning [11], and the burden of medical professionals in reading CT scans has been alleviated.

While deep learning techniques are well developed, applying them to COVID-19 presents several challenges. First, the lack of clear rules regarding the hyperparameters, including the filter sizes and the choice of pooling function, consumes considerable time in designing architectures and hampers its widespread application. As a result, neural network architectures cannot quickly be properly designed according to the actual situation. Another inevitable limitation is that only a few data sets are publicly available, and almost all the data come from a few sources, mainly in China. As data might not represent the global population, model bias will be generated, which will lead to poor performance [12–14]. Finally, most of existing CNNs are problem-specified, and model architectures need to be redesigned according to different characteristics. Currently, many researchers are attempting to automate yield hyperparameter settings, which are competitive with those manually designed by deep learning experts. Neural architecture search (NAS) is a technology that designs the proper CNN architectures without intensive human effort.

Numerous NAS strategies are used to explore the space of hyperparameter settings, including random search [15–17], Bayesian optimization [18–20], reinforcement learning [21–23], and evolutionary algorithms (EAs) [24–26].

With comprehensive studies on EAs, many efforts have been devoted to solving their weaknesses. For example, an automatic CNN architecture design method with EA [27] was developed to address image classification tasks effectively. A new type of EA-based Haar filter [28] was developed to detect all faces under different conditions, such as illumination variation and rotated faces. A novel evolutionary technique [29] defined the network architecture by a sequence of repeating neurocells, which reduces the searching space complexity. In [30], the authors described the advantages of EAs by controlled comparison of algorithms for architecture search in image classification. Specifically, evolutionary methods have a faster search speed than reinforcement learning approaches, especially in the early stages [30]. Compared with random search strategies, EAs generate smaller architectures and usually complete config searching tasks with competitive performance [30]. Even though EAs play a significant role in the NAS without additional expert knowledge, the fixed-length chromosome genetic encoding strategy may not be suitable for optimizing deep learning hyperparameters, because the performance of CNNs highly depends on the model depth [31]. Compared with the fixed-length chromosome encoding strategy, the variable-length chromosome encoding strategy is qualified for exploring dynamic solution space, which improves locality and significantly reduces redundancy in the representation [32]. For example, VLCGA-NB [33] selects sentiment types in tweets based on the examination of features by a variable-length chromosome EA. A version of the genetic algorithm [34] that uses the variable-length chromosome is proposed to deal with a time series discretization task.

This paper designs an effective and efficient variable-length EA method with a dynamic solution space to diagnose COVID-19 on lung CT images. The contributions of this paper can be summarized as follows:

- A novel EA with variable-length chromosomes is proposed to search hyperparameter settings of CNNs for COVID-19 diagnosis tasks. With the proposed balance operation, including the growth strategy and the shrink strategy, the length of chromosomes may be changed, and the hyperparameter searching space may be altered adaptively, which can significantly reduce dependence on data quality and neural network redundancy without careful pretreatment and expert knowledge.

- A new batch normalization (BN) finding is first presented. The experimental results show that the BN operation indeed boosts CNN performance. However, BN operations should not be used after every convolution
operation, and excess BN operations may poorly affect image analysis performance.

**Background and related work**

**Deep learning-based diagnosis of COVID-19**

Rapid and valid COVID-19 discrimination methods should be designed to control the COVID-19 pandemic and reduce mortality by receiving timely treatment. The traditional nucleic acid test has a nearly 30% false-negative rate [35], which means that negative nucleic acid test results cannot rule out COVID-19 infection. Virus carriers without the correct test result may further spread the disease and produce worse outcomes. In contrast, CT images can effectively diagnose COVID-19. Therefore, the correct recognition of virus-carrying patients through CT images can effectively reduce pressure on hospitals.

To alleviate the burden of medical professionals and efficiently allocate medical resources, many clinicians and scientists aim to develop an accurate COVID-19 diagnosis method based on CT images through novel algorithms. In the field of computer science, machine learning methods have the ability to solve practical difficulties. Moreover, simple machine learning methods have made it possible to accurately diagnose COVID-19 [36,37]. A sophisticated and effective form of machine learning, known as deep learning, achieves state-of-the-art results in many fields, and CNNs are widely used in deep learning areas to address COVID-19 diagnosis projects. For instance, a tailored CNN called COVID-NET [38] was designed to classify COVID-19 cases from an open and large chest X-ray image data set. A transfer learning-based method with CNNs [39] was adopted to diagnose coronavirus disease in a small X-ray image data set. A new deep anomaly diagnosis model [40] for fast and reliable screening of COVID-19 chest images was developed with a backbone network, a classification head, and an anomaly detection head. The backbone network uses an 18-layer residual CNN pre-trained on the ImageNet data set.

**Batch normalization (BN)**

Over the last decade, the impressive progress of deep learning hinged on several significant advances in terms of hardware, data sets, and architecture techniques [41–44]. One of the most prominent examples of such advances is BN [45]. In neural networks, the distribution of data has a significant impact on the training phase. In [45], the authors defined internal covariate shift (ICS) as the change in the distribution of network activation due to the change in network parameters during training. To stabilize the layer input distribution, BN operations, which control the first two moments (mean and variance) of these distributions, are introduced after every hidden layer (Fig. 1). Despite their pervasiveness, the effectiveness of the working mechanism is still poorly understood. In addition, there are still controversies about how BN works. In [46], the effectiveness of BN was investigated, and the conclusion was that the connection between BN and ICS performance is tenuous. With more research on BNs, these findings not only help researchers understand more BN details but also help the development and effectiveness of deep learning-related algorithmic toolkits.

**Evolutionary algorithm with a dynamic searching space**

Compared with traditional machine learning methods, deep learning methods brilliantly represent data through hidden layers. The deep network structure can easily learn patterns from raw data and abstractly express data with layer-by-layer extraction, which improves task performance. However, this kind of improvement quickly reaches the upper limit. Specifically, the gradient in the backpropagation algorithm will be smaller after more hidden layers. As a result, the performance of deep learning tasks will decrease. For example, the CNN used in AlphaGo has only 13 layers, and the accuracy of the chess move prediction reached 57%. However, when the number of CNN layers was further increased, the prediction accuracy decreased [47].

EAs are distinguished global optimizers for black-box functions [48] and can provide a promising set of methods for structural and parametric learning of CNNs [49]. EAs consist of the following essential components: initialization, parent selection, crossover, mutation, and environmental selection. The initialization defines a gene encoding strategy, including genotype [50,51] and phenotype [29,52]. In most common EAs, the individuals are represented by a fixed-length encoding [26,53–55]. Because the depth of a particular CNN greatly affects its performance [9], the variable-length chromosome initialization strategy is suitable for complicated occasions [52]. Figure 2a shows the differences between the fixed-length chromosome strategies and the variable-length chromosome strategies in the initialization stage. Instead of generating the same length parents by the a fixed-length stra-
egy, the length of parents generated by the variable-length strategy is unequal. Tournament selection [53] and fitness proportionate selection [56] are often used for parent selection. The tournament selection method iteratively selects parents with some prespecified threshold values in descending order. In the fitness proportionate selection method, the $i$th individual is chosen with probability $\frac{f(\alpha_i)}{\sum_{j=1}^{N} f(\alpha_j)}$, where $\alpha_i \in \{\alpha_1, \ldots, \alpha_N\}$ and $f(\alpha_i)$ is the fitness of the individual $\alpha_i$. After all the needed parents are selected, the offspring are generated by a crossover operation, which is key to forming new individuals based on exchanging genes. As shown from Fig. 2b, the lengths of the children generated by the fixed-length strategy are equal, but the variable-length strategy obtains the opposite result. The mutation operation generates new chromosomes through mutation probability so that it shows new traits that determine the EA global search ability. Environmental selection addresses elitism by preserving excellent individuals according to fitness evaluation, thus enhancing diversity by initializing new individuals.

The proposed algorithm

In this section, the proposed evolutionary algorithm with a variable-length chromosome strategy for COVID-19 (EAVL-COVID) diagnosis is described in detail.

Algorithm overview

**Algorithm 1:** The framework of the EAVL-COVID

```plaintext
1 $P_0$ ← Initialize the variable-length chromosome individuals with the gene encoding strategy; 
2 $t ← 0$; 
3 while termination criterion is not satisfied do 
4   $p ← 0$; 
5     for $p < \text{popsizes}$ do 
6       Evaluate the Fitness$_{p}$ of the $p$th individual 
7       $p ← p + 1$ 
8     end 
9   $S ← \text{Select parents by Fitness}_{p}$; 
10  $P_{t+1} ← \text{Generate next generation individuals by environmental selection}$; 
11  Fitness$_{best} ← \text{The global-best fitness so far}$; 
12 end 
13 Select the best individual from $P_t$ and decode it to the corresponding CNN;
```

The proposed EAVL-COVID algorithm is mainly composed of two elaborate designs. First, to change the search solution space based on different tasks, EAVL-COVID generates variable-length chromosomes in the initialization step. Second, in EAVL-COVID, the crossover points are chosen randomly, which results in the generated children having chromosomes that are too long or too short after the crossover operation. The neural network, decoded by a long chromosome, may have a deep architecture. In contrast, the neural network, which is decoded by a short chromosome, cannot solve a difficult task with a plain architecture. As a result, the balance operation with the growth strategy and the shrink strategy is developed to solve unsuitable chromosomes.

Algorithm 1 describes the framework of the proposed EAVL-COVID algorithm, where our contributions and developments based on [52] are highlighted in bold and italics. First, the population is initialized based on the proposed variable-length gene encoding strategy. Specifically, the BN layers are probabilistically added after every convolutional layer (line 1). Second, the evolution begins to take effect until the maximum number of generations is satisfied (lines 3–11). Notably, the random immigrants strategy [57], which is widely used in EA-based NAS methods [26,53,58], is adopted to maintain the global diversity level of the population [59]. Finally, the best individual is selected and decoded to the corresponding CNN (line 16) for a specific task.

Population initialization

A CNN model usually consists of different blocks, such as convolutional layers, BN layers, pooling layers, and fully connected layers. In detail, unlike encoding binary numbers into chromosomes in EAs, the EAVL-COVID algorithm encodes all necessary parameters into four essential blocks presented in Table 1. Particularly, the goal of using the convolutional layers is to extract high-level features with the convolutional filters. With the increase in the number of convolutional layers, the ability to solve challenging tasks can be reinforced. However, the deeper the architecture is, the higher the risks of overfitting and computational consumption. In other words, the proposed variable-length chromosome strategy alerting the solution space can effectively and efficiently solve the particular problem with the appropriate number of convolutional layers.

The steps of initializing a chromosome by EAVL-COVID are described in Algorithm 2. Notably, the number of convolutional layers $n_c$, pooling layers $n_p$ and fully connected layers $n_f$ are chosen from the predefined ranges (line 2). Specifically, the convolutional layers and the pooling layers are chosen by the once coin-tossing probability (lines 3–22). If the convolutional layers are chosen, with or without a BN layer should be randomly determined (lines 11–19). After repeatedly choosing $n_c$ convolutional layers and $n_p$ pooling layers, $n_f$ fully connected layers are added to the end of the chromosome by encoding the necessary information, such
as the number of neurons, which are predefined in advance (lines 23–27).

After all those steps are finished, the EA VL-COVID algorithm obtains one integrated individual. With the same approach, a population with variable-length chromosome individuals is generated.

Parents selection and chromosome crossover

The steps for selecting parents and generating offspring are given as follows:

step 1: select parents through a fitness-based process, where fitter solutions are typically from the population $P_t$;
step 2: divide the chromosome of each parent into two parts, because the crossover operation is different;
step 3: use a crossover strategy on the selected parents to generate offspring.

First, EA VL-COVID uses a fitness-based method to select two parents, which means that the more suitable the parent is for the COVID-19 classification task, the higher the probability of being selected.

Second, each selected parent chromosome is divided into a head chromosome and a tail chromosome, because the crossover operation for each part is different. The head chromosome includes convolutional blocks (C), batch normalized convolutional blocks (BNC), and pooling blocks (P). The tail chromosome only involves fully connected blocks (F).

Third, the crossover strategy is implemented on the selected parents to generate offspring. Figure 3 shows the crossover operation intuitively. In particular, the crossover points are first randomly generated in head chromosomes. Then, the head chromosomes before the generated points are exchanged. After performing the crossover strategy on the head chromosome, all the blocks in the tail chromosome at the same positions perform a crossover strategy. Due to individuals possessing variable-length tail chromosomes, if the F in one parent mismatches those at the same positions as the other parents, those F will not participate in crossover processing. After the above steps are finished, offspring that have information on both parents are generated.

Balance operation

As the crossover points are chosen randomly, the lengths of all offspring chromosomes are diverse. If the chromosome is too long, a deep neural network encoded by this chromosome may be generated. Even if the complicated architecture may help accomplish a difficult task, it may also cause overfitting or consume too many computational resources. In contrast, if the chromosome is too short, the plain neural network may be unequipped with the ability to solve a difficult problem.
Fig. 3 Example to illustrate the crossover operation with variable-length chromosomes. In this example, parents are split into two parts (see a). The head-chromosome is made up of C, BNC and P. All F are all in the tail-chromosome and the first F in each parent are aligned at the same position. b Fully shows the crossover operation. For head-chromosome, a crossover point is random selected in each parents separately. After the head-chromosomes before crossover points are exchanged, the F in the same position are exchanged, respectively. c Describes the offspring generated by the selected parents through crossover operation.

To address those shortcomings caused by the variable-length chromosomes, the balance operation with the shrink strategy and the growth strategy, which replaces the mutation operation in original EAs, is proposed in our work. Figure 4 expressively describes the balance operation as an example.

More specifically, a shrink strategy is taken if \( l \geq \varphi \), where \( l \) is the difference between the length of the offspring’s head-chromosome \( \text{len}_{\text{head}}^{\text{head}} \) of \( \text{fspring} \) and the individual’s head-chromosome with global-best fitness \( \text{len}_{\text{best}}^{\text{head}} \).

\( \varphi \) is a hyperparameter for controlling the maximum depth of networks. First, the shrunken length \( \text{len}_{\text{shrink}} \) of the shrink strategy is determined by a probabilistic method. Given a normal distribution \( N(l, 1) \), each element \( i \in [1, \ldots, l] \) is given a value called \( \rho_i \), where

\[
\rho_i \sim N(i; l, 1).
\] (1)

After normalizing all \( \rho_i \), the choice probability of \( \text{len}_{\text{shrink}} \) is obtained and the choice probabilities determine \( \text{len}_{\text{shrink}} \). Then, a shrink point \( \text{index}_{\text{shrink}} \) is generated in the position range \([1, \text{len}_{\text{offspring head}} - \text{len}_{\text{shrink}}]\). A balanced offspring is generated with \( \text{index}_{\text{shrink}} \) and \( \text{len}_{\text{shrink}} \).

However, if \( \text{len}_{\text{offspring head}} < \varphi \), the short offspring will be performed with a growth strategy. Specifically, a growth point \( \text{index}_{\text{growth}} \) is generated within the position range \([1, \text{len}_{\text{offspring head}}]\). Because of the global-best fitness individual with the excellent genes solves a particular job well, a fragment with \( \varphi \) consecutive genes is selected from the global-best fitness individual. Then, the selected fragment is inserted into the short offspring at the generated position \( \text{index}_{\text{growth}} \).

In [52], even though variable-length chromosomes are generated in the population initialization phase, the chromosome length is fixed during evolution. However, the proposed balance operation with the shrink strategy and the growth strategy vary the length of chromosomes during evolution, which may exceed the predefined range. This means that the search space is dynamic, and an optimal CNN architecture could be principally found to address complex problems based on different data sets.

Environmental selection

To ensure the elitism and diversity of the evolutionary population, the next generation consists of the elites in the current population, the offspring generated by the crossover operation and the balance operation, and new individuals generated by initialization. The environmental selection is shown in Algorithm 3. The proposed EAVL-COVID algorithm chooses the top \( N_{\text{elite}} \) elites with the fitnesses in descending order from the current population to ensure the elitism. Then, \( N_{\text{new}} \) individuals are initialized to help the EA avoid local optimization and increase the opportunities for better results.
Fig. 4 Example shows the balance operation to deal with a long offspring and a short offspring generated by the proposed crossover operation. The numbers of units in head-chromosomes \( len_{\text{of} \, \text{spring}_{\text{head}}} \) are 18 and 5, respectively (a). b) Detailedly give the steps of the balance operation, where \( len_{\text{best}_{\text{head}}} = 10 \) and \( \phi = 6 \). Due to the long offspring \( len_{\text{of} \, \text{spring}_{\text{head}}} - len_{\text{best}_{\text{head}}} > \phi \), the shrink strategy chooses a shrink position \( index_{\text{shrink}} = 8 \) and the shrunken length \( len_{\text{shrink}} = 7 \); For the short offspring, where \( len_{\text{of} \, \text{spring}_{\text{head}}} < \phi \), a fragment with \( \phi \) consecutive genes is selected in global-best chromosome, followed by a growth point \( index_{\text{growth}} = 2 \). Finally, the selected fragment is inserted after the growth point. c) Gives the picture that the balanced offspring with both \( len_{\text{of} \, \text{spring}_{\text{head}}} = 11 \) after the balance operation.

(a) Generated a long and a short offspring because of the random choice crossover point strategy.

(b) Balance operation with the shrink strategy and the growth strategy.

(c) Balanced offspring.

### Experimental design

In this section, to examine the performance of the proposed algorithm on the COVID-19 classification task, the data set, peer competitors and parameter settings are provided, which are documented in “COVID-CT dataset”, “Peer competitors” and “Parameter settings”, respectively.

### COVID-CT data set

The COVID-CT data set \cite{60} has 349 CT images labelled as being positive for COVID-19 from 216 patients, which were collected from COVID-19-related papers and 463 negative CT images from four biomedical CT scan data sets. This data set was confirmed by a radiologist who performed the diagnosis and treatment of a large number of COVID-19 patients during the virus outbreak in Wuhan. Detailed information about the data set is shown in Table 2.

Because the original clinical findings of COVID-19 images were obtained from papers, the sizes of these images are diverse. In addition, many positive COVID-19 CT images are degraded, so the quality of those images is unpromising. Figure 5 illustrates examples of CT images from the COVID-CT data set. There are numerous invalid marks on the labelled positive COVID-19 lung images, which may render the diagnosis decisions less accurate. Moreover, the size of the COVID-CT data set, compared with other computer vision tasks in the deep learning area, is severely insufficient. As a result, the CT images must be preprocessed, but this process is extremely time consuming, and the architecture used for training must be elaborately designed.

### Peer competitors

To demonstrate the effectiveness of the proposed EAVL-COVID algorithm, we first experiment on randomly initialized networks with different backbones as baselines, and pre-trained them on ImageNet. The networks include VGG-16 \cite{9}, ResNet-18 \cite{10}, ResNet-56 \cite{10}, DenseNet-121 \cite{61}, DenseNet-169 \cite{61}, EfficientNet-b0 \cite{62}, and EfficientNet-b1 \cite{62}. The BN operation is used through all models and the binary cross-entropy serves as the loss function.

As a basic network, VGG-16 has good classification performance and adaptability to many data sets and provides meaningful performance for the COVID-19 classification task on the COVID-CT data set. As the depth of the network increases, the model obtains better representation ability. However, with the deeper architecture, the training accu-
Fig. 5 Examples of CT images from COVID-CT data set

(a) Examples of CT images that are negative for COVID-19

(b) Examples of good-quality CT images that are positive for COVID-19

(c) Examples of low-quality and high-noise CT images that are positive for COVID-19

Algorithm 2: Individual Initialization

Input: The maximal number of convolutional layers \( N_c \), the pooling layers \( N_p \) and the fully connected layers \( N_f \)

Output: An initialized individual \( P \)

1. \( P \leftarrow \emptyset \);
2. \( n_c, n_p, n_f \leftarrow \text{Randomly generate an integer between} [1, N_c], [0, N_p] \text{ and} [0, N_f] \);
3. \( \textbf{while} \ n_c + n_p \geq 0 \ \textbf{do} \)
   4. \( r \leftarrow \text{Randomly generate a number between} [0, 1] \);
   5. \( \textbf{if} \ r \leq 0.5 \ \textbf{then} \)
   6. \( l \leftarrow \text{Initialize a pooling layer} \);
   7. \( n_p \leftarrow n_p - 1 \)
   8. \( \textbf{end} \)
   9. \( \textbf{else} \)
   10. \( r_c \leftarrow \text{Randomly generate a number between} [0, 1] \);
   11. \( \textbf{if} \ r_c > 0 \ \textbf{then} \)
   12. \( l \leftarrow \text{Initialize a convolutional layer with BN operation} \);
   13. \( n_c \leftarrow n_c - 1 \)
   14. \( \textbf{else} \)
   15. \( l \leftarrow \text{Initialize a convolutional layer without BN operation} \);
   16. \( n_c \leftarrow n_c - 1 \)
   17. \( \textbf{end} \)
18. \( \textbf{end} \)
19. \( P \leftarrow P \cup l \);
20. \( \textbf{end} \)
21. \( \textbf{while} \ n_f > 0 \ \textbf{do} \)
22. \( l \leftarrow \text{Initialize a fully connected layer} \);
23. \( n_f \leftarrow n_f - 1 \);
24. \( P \leftarrow P \cup l \);
25. \( \textbf{end} \)
26. \( \textbf{return} \ P \)

Algorithm 3: Environmental Selection

Input: The current population \( P_t \), and the corresponding fitnesses \( acc \_list \), the number of individuals \( N \), the number of elites \( N \_elite \) and the number of new individuals \( N \_new \)

Output: The selected population \( P_{t+1} \)

1. \( P \_elite \leftarrow \text{Select individuals in} P_t \text{ with the top} N \_elite \text{fitnesses in} acc \_list \);
2. \( i \leftarrow 0; \)
3. \( P \_offspring \leftarrow \emptyset; \)
4. \( \textbf{while} \ i < \frac{(N - N \_elite - N \_new)/2}{\textbf{do}} \)
5. \( P \_generate \leftarrow \text{Select parents and generate offspring} \) (details see “Parents selection and chromosome crossover”);
6. \( P \_offspring \cup P \_generate \)
7. \( \textbf{end} \)
8. \( P \_new \leftarrow \text{Initialize} N \_new \text{individuals} \) (details see “Population initialization”);
9. \( P_{t+1} \leftarrow P \_elite \cup P \_offspring \cup P \_new \);
10. \( \textbf{return} \ P_{t+1} \)

Table 2 Detailed information about COVID-CT

|        | COVID-19 | Non-COVID-19 |
|--------|----------|-------------|
| Train set | 88       | 231         |
| Test set | 173      | 168         |
| Val set  | 88       | 64          |
| Total    | 349      | 463         |

Accuracy tends to be flat, and the training error becomes larger. ResNets use the skip connection to solve the optimization problems when the number of hidden layers is deepened, such as ResNet-18 and ResNet-56. DenseNet realizes feature reuse through the connection of features on the channel. These features allow DenseNet-121 or DenseNet-169 to
achieve better performance than ResNets with fewer parameters and computational costs. To pursue better accuracy and efficiency, EfficientNet balances all dimensions of network width, depth, and resolution during convolutional neural network scaling by neural architecture search. Unlike the EA used by the proposed EAVL-COVID algorithm, EfficientNet introduces a factorized hierarchical search space and solves the Pareto optimal solution of the multi-objective search problem by reinforcement learning [63].

**Parameter settings**

All the parameters are set based on the conventions in the EA communities and deep learning. The proposed EAVL-COVID algorithm is implemented in PyTorch, and each copy of the code runs on a computer equipped with two identical GTX2080Ti GPU cards.

First, in the EA initialization, the population size is set to be 50. The range of the convolutional layer depth with a 3 × 3 kernel is set to be [5, 8], and those convolutional layers have a 50% chance of performing a BN operation. The number of neurons in each convolutional layer is set to be [50, 100]. The maximal depth of the pooling layers whose kernel size and stride size are both set to be 2, and fully connected layers with [100, 500] neurons are set to be 5 and 4, which is believed to be sufficient to solve the benchmark data sets employed at the beginning of the classification task. In the crossover step, the probability of the crossover operation is set to be 0.9. In the balance step, the threshold value \( \phi \), which determines the choice of the growth strategy or the shrink strategy, is set to be 4. In the environmental selection step, 10 elites are preserved to the next generation, and 10 new individuals are generated by a gene encoding strategy. To weight the performance of each individual, the chromosomes are decoded into neural networks that are trained with the following process. In the training phase, the Adam optimizer is used with an initial learning rate of 1e−4 and a batch size of 16. Due to the time consumption of the proposed method, the training epoch of each individual is set to be 100, which is inadequate but workable during the EA procedure. Due to the stochastic nature of the proposed EAVL-COVID method, the chromosome with the best classification accuracy is generated after 20 generations. To achieve the truthful performance, the best chromosome is decoded and trained by another 1000 epochs.

**Experiment**

To perform the COVID-19 classification task on the COVID-CT data set, the proposed EAVL-COVID algorithm follows three steps. First, the generation number is set to 20. The evolution curve of the proposed EAVL-COVID algorithm is shown in Fig. 6. Second, after 15 generations, the chromosome with the global-best fitness is generated. Third, this chromosome is decoded to the convolutional neural network, which is used for the COVID-19 classification task. To investigate the character of the BN operation, the best chromosome is decoded into three architectures: ORIGIN, VARIANT#1 and the VARIANT#2. ORIGIN represents the architecture searched by the EAVL-COVID algorithm without any modification. The three architectures on the COVID-CT data set are illustrated in Table 3. As shown in Table 3, BN operations are not used in the three convolutional layers of the ORIGIN. They are conv-79 located at the 4th layer, conv-99 located at the 7th layer, and conv-58 located at the 9th layer. VARIANT#1 adds a BN operation after those three convolutional layers, where the ORIGIN is missing. In addition, VARIANT#2 removes all BN operations of the ORIGIN.

To evaluate the performance of the proposed EAVL-COVID algorithm, seven networks for the COVID-CT data set are chosen as the baseline. We first experiment on the randomly initialized networks and then the pre-trained networks on a large-scale data set. The quantitative results are documented in Table 4 and the best performance is highlighted in bold.

**Evaluations on randomly initialized networks**

To fairly evaluate the performance of EAVL-COVID and backbones on the COVID-CT data set, the Kaiming initialization strategy [64] is adopted to initialize the weights randomly. As seen in the performance (columns marked with “RAND.”) for different neural networks trained with random initialization in Table 4, the ORIGIN evolved by EAVL-COVID achieved 81.3% accuracy and obtained the best performance among all competitors. Amazingly, the evolution results of ORIGIN outperform the performance of
Table 3 ORIGIN searched by EAVL-COVID and its variants on the COVID-CT data set

| ORIGIN searched by EAVL-COVID and its variants |
|-----------------------------------------------|
| **ORIGIN** | **VARIANT#1** | **VARIANT#2** |
| maxpooling | maxpooling | maxpooling |
| *conv-91*  | *conv-91*  | conv-91  |
| *conv-69*  | *conv-69*  | conv-69  |
| *conv-51*  | *conv-51*  | conv-51  |
| conv-79    | *conv-79*  | conv-79  |
| *conv-72*  | *conv-72*  | conv-72  |
| *conv-53*  | *conv-53*  | conv-53  |
| conv-99    | *conv-99*  | conv-99  |
| maxpooling | maxpooling | maxpooling |
| *conv-77*  | *conv-77*  | conv-77  |
| conv-58    | *conv-58*  | conv-58  |
| maxpooling | maxpooling | maxpooling |
| FC-347     |             |           |
| FC-344     |             |           |
| FC-256     |             |           |
| FC-193     |             |           |
| softmax    |             |           |

*The convolutional layer followed a BN layer*

DenseNet-169 with skip connection operation. The following two reasons can explain the result. First, the original clinical findings of COVID-19 images were obtained from published papers. Many positive COVID-19 CT images are degraded with additional marks. This drawback of COVID-CT reduces the classification accuracy without preprocessing. The proposed variable-length chromosome strategy improves locality and significantly reduces redundancy in the representation. In conclusion, it is difficult for statistical scientists to design a suitable architecture without discreetly processing data. The second reason is that the COVID-CT is a data set with only 349 positive COVID-19 CT images. The architecture of the neural network deeply affects the accuracy of the classification task. Even though DenseNet introduces skip connections to avoid overfitting, the sophisticated architecture still fails in the classification task on the COVID-CT data set. Without unique domain knowledge designs, there are many obstacles to perfectly finishing a particular goal by transferring models that have excellent performance on other tasks in a short time. However, the proposed EAVL-COVID algorithm, which works automatically, with a dynamic search space and without any preprocessing or manual parameter settings, is perfectly suitable for this complicated task.

In addition, the BN layers improve the performance of image classification tasks. Compared with VARIANT#2 based on the COVID-CT data set, ORIGIN and VARIANT#1, which have BN operations, improve accuracy by 13.3 and 6.9%, respectively. Surprisingly, the experimental results show that the classification accuracy deteriorates by 6.4% if all convolutional layers are followed by BN layers.

**Evaluations on pre-trained networks**

In deep learning, many training tricks boost classification performance. The pre-train strategy is largely used for general knowledge discovery with no training phase or use of labels to generate high-level representations, which helps much research to achieve state-of-the-art results over various classification benchmarks. After evaluating the performance of different neural networks on random initialization, we further investigate the performance of networks pre-trained on ImageNet. Columns marks with “PRET.” in Table 4 show the classification accuracy on COVID-CT after training the networks on the ImageNet data set.

In certain cases, the pre-training strategy improves the task performance of the networks. For example, a 14.3% improvement in accuracy appears in VGG-16, which has the worst performance in the random initialization. After transfer learning, the performance of VGG-16 exceeds the performance of ResNet-18 and ResNet-50. The reason is apparent.

Table 4 Performance comparison between randomly initialized networks (RAND.) and ImageNet pretrained networks (PRET.)

|                | Parameter numbers | RAND. | PRET. |
|----------------|-------------------|-------|-------|
| BASELINE [65]  | VGG-16            | 138,357,544 | 0.632 | 0.773 |
|                | ResNet-18         | 11,689,512  | 0.652 | 0.734 |
|                | ResNet-50         | 25,557,032  | 0.687 | 0.775 |
|                | DenseNet-121      | 7,978,856   | 0.773 | 0.814 |
|                | DenseNet-169      | 14,149,480  | 0.807 | 0.842 |
|                | EfficientNet-b0   | 5,288,548   | 0.695 | 0.774 |
|                | EfficientNet-b1   | 7,794,184   | 0.726 | 0.797 |
| EAVL-COVID     | VARIANT#2         | 808,222     | 0.680 | 0.663 |
|                | VARIANT#1         | 810,565     | 0.749 | 0.738 |
|                | ORIGIN            | 810,093     | 0.813 | 0.815 |
Compared with sophisticated architectures designed for preventing overfitting, VGG-16 speeds up convergence and improves the ability to extract features through pre-trained weights. In contrast, the effect of the pre-training strategy on the ORIGIN is marginal. The reason is that the proposed EAVL-COVID algorithm has found a good solution for a specific task. The variable-length chromosome strategy reduces the architecture redundancy and avoids overfitting in large networks. Without any preprocessing operation, the ORIGIN’s performance can compete with DenseNet-169, whose accuracy was the highest at 84.2% after the pre-training strategy.

Similar results appear in the pre-trained network. Among VARIANT#1 and VARIANT#2, ORIGIN achieves the best performance on the COVID-CT data set. In conclusion, the BN operation may reduce the performance in some situations, so it is unnecessary to perform BN operation after every convolution operation. This interesting viewpoint was found by experimental results, and to the best of our knowledge, is summarized for the first time.

Ablation experiments

The idea of the variable-length strategy is hyper-heuristic optimization methods that construct a heuristic from several low-level heuristics to solve common problems in operations research. In searching for the best heuristic, the optimal number of low-level heuristics is unknown a priori, so a variable-length chromosome is designed to explore heuristics made from low-level heuristics. Similarly, a variable-length chromosome can be used to search the number of hidden layers, since the number of hidden layers in different data sets for optimal coverage performance is unknown a priori. With the proposed growth strategy and shrink strategy, the proposed variable-length chromosome representation adaptively alters the searching space and may find the optimal number of hidden layers based on different data sets.

To fully understand the effects of the variable-length strategy and investigate the usage rules of BN operation, three competitors are chosen to conduct ablation studies.

- VGG16BN: A classic CNN human-made architecture with BN operation.
- Random Search(RS): An NAS method that randomly searches for the CNN architecture in a similar searching space as the initial definition of the proposed EAVL-COVID algorithm, and the searching space is static during the whole process. Compared with EAVL-COVID, the RS does not determine whether to use BN. Basically, each convolutional layer is followed by the BN operation. To be fair, the RS generates 1000 architectures on each data set used.
- EVO-CNN: An EA-based NAS method evolutionarily searches for the CNN architecture in a searching space similar to the initial definition of the proposed EAVL-COVID algorithm, and the search space is static during the whole process. In addition, the BN operation is adopted by each convolutional layer.
- EAVL-COVID: The proposed EA-based NAS method searches for the CNN architecture in a dynamic searching space. The initially defined searching space is demonstrated in “Parameter settings”.

To investigate the advancements of the proposed EAVL-COVID algorithm with the variable-length chromosome strategy during the crossover and mutation stages, ablation experiments are performed on four data sets.

- COVID-CT: A small data set with low-quality and high-noise images, and there are two classes in the COVID-CT.
- CIFAR10: A classic data set in the computer vision community, and each class contains 10 times more images than the COVID-CT.
- STL10: A data set with high-resolution images, which means that the high dimensionality of data increases the computational complexity of algorithms owing to the effect of noise [66,67]. Compared with CIFAR10, each category contains fewer labelled training examples.
- CIFAR100: A data set with 100 classes that greatly exceed the number of classes in COVID-CT, CIFAR10, and STL10.

To fairly evaluate the performance of ablation experiments, only the Kaiming initialization strategy is adopted to initialize the weights randomly. Moreover, no preprocessing or post-processing techniques are adopted on the human-made architecture, the autosearched architectures and data sets.

The results of ablative experiments on different data sets are described in Table 5 and the best performance is highlighted in bold. It is clearly shown in Table 5 that the proposed EAVL-COVID algorithm achieves excellent performance on all chosen data sets. In detail, EAVL-COVID obtains significant advantages on the COVID-CT and CIFAR10 data sets. On COVID-CT, the accuracy of EAVL-COVID is 18.1% higher than that of the classical human-made architecture CNN. Moreover, the proposed EAVL-CNN also ranks first among the NAS methods. Compared with the performance of EAVL-CNN on COVID-CT, the accuracy rates of RS and EVO-CNN are less than 19.5 and 5.5%, respectively. A similar phenomenon can also be observed from the CIFAR10, CIFAR100, and STL10 data sets. Notably, the RS fails to achieve good performance on all data sets due to the ineffective search strategy and the limited number of generated architectures.
As shown in Table 5, EAVL-COVID generates different numbers of layers on different data sets, and the generated depth of the architectures may exceed the predefined value, i.e., the searching space is dynamic. Specifically, the number of hidden layers is set to \([5, 8]\), and the number of fully connected layers is \([1, 4]\). As a result, the maximal depth of the architectures searched by the NAS methods with static searching space is 12. However, the proposed EAVL-COVID algorithm can obtain CNNs exceeding predetermined depths. In the experiments, the depths of the architectures searched on COVID-CT, CIFAR100, and STL10 were 13, 15, and 13, respectively. Based on the results shown in Table 5, the conclusion can be made that the optimal priori about the depth of the architectures on different data sets can be automatically obtained with the variable-length strategy with the proposed crossover strategy and mutation strategy.

To investigate whether it is necessary to use the BN operation after each convolutional layer, more experiments are performed based on the architectures searched by the proposed EAVL-COVID algorithm on CIFAR10, CIFAR100, and STL10. First, the architecture searched by the EAVL-COVID is expanded into ORIGIN, VARIANT #1, and VARIANT #2. Second, three additional optimizers are selected to verify if the observation still holds true. Each architecture is conducted five times with the Kaiming initialization strategy and trained for 1000 epochs. The experimental results are documented in Table 6. The best performance of the architectures is highlighted in bold according to the optimizers and data sets. A similar conclusion about BN can also be found in Table 6. Compared with VARIANT #2 and VARIANT #1 based on the CIFAR10, the best overall test accuracies are all found by the ORIGIN. In detail, the ORIGIN achieves a test accuracy of \(0.885 \pm 0.0014\) with SGD, \(0.862 \pm 0.0046\) with Adadelta, \(0.863 \pm 0.0018\) with RMSprop, and \(0.881 \pm 0.0025\) with Adam. Moreover, the results on the CIFAR100 and the STL10 have the similar circumstances. Notably, a special situation appears in the CIFAR100 with SGD. Even though the VARIANT #1 achieves the best result, the performance of the ORIGIN is still competitive with a high standard deviation. In conclusion, it is not necessary to perform BN operations after every convolutional layer. To the best of our knowledge, this is the first time this interesting viewpoint has been experimentally identified.

To investigate whether it is necessary to use the BN operation after each convolutional layer, more experiments are performed based on the architectures searched by the proposed EAVL-COVID algorithm on CIFAR10, CIFAR100, and STL10. First, the architecture searched by the EAVL-COVID is expanded into ORIGIN, VARIANT #1, and VARIANT #2. Second, three additional optimizers are selected to verify if the observation still holds true. Each architecture is conducted five times with the Kaiming initialization strategy and trained for 1000 epochs. The experimental

| Table 5 | Performance of ablative experiments on different data sets |
| --- | --- | --- | --- | --- |
| COVID-CT | CIFAR10 | CIFAR100 | STL10 |
| Accuracy | # of layers | Accuracy | # of layers | Accuracy | # of layers | Accuracy | # of layers |
| VGG16BN | 0.632 | 16 | 0.611 | 16 | 0.602 | 16 | 0.589 | 16 |
| RS | 0.618 | 11 | 0.583 | 12 | 0.512 | 12 | 0.544 | 12 |
| EVO-CNN | 0.758 | 10 | 0.784 | 12 | 0.587 | 12 | 0.582 | 12 |
| EAVL-COVID | **0.813** | **13** | **0.899** | **11** | **0.648** | **15** | **0.590** | **13** |

| Table 6 | Performance of EAVL-CNN and its variants on different data sets |
| --- | --- | --- | --- | --- |
| Optimizer | Architecture | CIFAR10 | CIFAR100 | STL10 |
| SGD | VARIANT #2 | 0.615 ± 0.0027 | 0.525 ± 0.0468 | 0.534 ± 0.0127 |
| | VARIANT #1 | 0.678 ± 0.0018 | **0.628 ± 0.0219** | 0.560 ± 0.0153 |
| | ORIGIN | **0.885 ± 0.0014** | 0.624 ± 0.0441 | **0.576 ± 0.0237** |
| Adadelta | VARIANT #2 | 0.536 ± 0.0042 | 0.545 ± 0.331 | 0.537 ± 0.0094 |
| | ORIGIN | **0.862 ± 0.0046** | **0.640 ± 0.257** | **0.584 ± 0.0132** |
| RMSprop | VARIANT #2 | 0.572 ± 0.0027 | 0.553 ± 0.0354 | 0.526 ± 0.0157 |
| | ORIGIN | **0.863 ± 0.0018** | **0.655 ± 0.0223** | **0.581 ± 0.0095** |
| Adam | VARIANT #2 | 0.566 ± 0.0031 | 0.541 ± 0.0143 | 0.538 ± 0.0111 |
| | ORIGIN | **0.881 ± 0.0025** | **0.642 ± 0.0117** | **0.588 ± 0.0075** |
The model optimized by EA VL-COVID reached the highest accuracy. In the method-applied stage, the proposed algorithm significantly outperforms most competitors on the COVID-CT data set in terms of their best classification performance. The effectiveness of the EAVL-COVID algorithm is fully verified. Without any extra designs, the performance of the network searched by EAVL-COVID is still competitive with a high standard deviation. In conclusion, it is not necessary to perform BN operations after every convolutional layer. To the best of our knowledge, this is the first time this interesting viewpoint has been experimentally identified.

Conclusion and future work

The objective of the paper was to develop a new evolutionary approach to automatically evolve the CNN architecture for the COVID-19 classification task. This goal was successfully achieved by the proposed variable-length chromosome strategy with the growth operation and the shrink operation, which are suitable for classification tasks with a dynamic searching space. With the detailed investigation on random initialization networks and pre-trained networks, the experimental results showed that the proposed EAVL-COVID algorithm significantly outperforms most competitors on the COVID-CT data set in terms of their best classification performance. The effectiveness of the EAVL-COVID algorithm is fully verified. Without any extra designs, the performance of the network searched by EAVL-COVID is still competitive, which can save time and human resources. Furthermore, an important conclusion about BN operation is drawn in the related research area. The experimental results showed that it is not necessary to perform a BN operation after every convolution layer, because it may reduce the classification accuracy. In the method-applied stage, the proposed algorithm is successfully applied to the COVID-19 diagnosis task. The model optimized by EAVL-COVID reached the highest classification performance, 1.8% higher than the score obtained by DenseNet-169 with sophisticated skip connections and dense connections. In the future, we will invest efforts in combining the variable-length chromosome strategy and skip connection, which will widen the application area of EAVL-COVID. In addition, research on the working mechanism of the BN operation will be further included. Moreover, we will investigate the phenomenon that pre-training on ImageNet does not significantly improve the performance of the architecture searched by the evolutionary algorithm.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grants nos. U19A2078, 61971296, 61836011), and Sichuan Science and Technology Planning Project (Grants nos. 2021YFG0317, 2021YFG0301).

Declarations

Conflicts of interest To the best of our knowledge, the named authors have no conflict of interest, financial or otherwise. All authors have declared that: (i) no support, financial or otherwise, has been received from any organization that may have an interest in the submitted work; and (ii) there are no other relationships or activities that could appear to have influenced the submitted work.

Consent for publication Written informed consent for publication was obtained from all participants.

Availability of data and material All data generated or analyzed during this study are included in this published article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Covid-19 data compiled by Johns Hopkins university. https://coronavirus.jhu.edu/
2. Vaishya R, Javaid M, Khan IH, Haleem A (2020) Artificial intelligence (AI) applications for COVID-19 pandemic. Diabetes Metab Syndr Clin Res Rev
3. Ai T, Yang Z, Hou H, Zhan C, Chen C, Lv W, Tao Q, Sun Z, Xia L (2020) Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases. Radiology 200642
4. Salzmann L, Jiang H, Vezhnevets V, Poupyrev I, Zhang C, Wu Q, Wang C, Xin S, Ni L, Chen P, Bunke H, Gutierrez I, Guenter J, Suter B (2020) Large-scale multi-modality COVID-19 screening at Drumet Medical Imaging. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 840–849
5. Zhang F, Luo L, Sun X, Zhou Z, Li X, Yu Y, Wang Y (2019) Cascaded generative and discriminative learning for microcalcification detection in breast mammograms. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 12578–12586
6. Hu P, Wang X, Zhen L, Peng D (2019) Separated variational hashing networks for cross-modal retrieval. In: Proceedings of the 27th ACM international conference on multimedia. ACM, pp 1721–1729
7. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25(2)

8. Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR (2012) Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint. arXiv:1207.0580

9. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Comput Sci

10. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

11. Gozes O, Frid-Adar M, Greenspan H, Browning PD, Zhang H, Ji W, Bernheim A, Siegel E (2020) Rapid AI development cycle for the coronavirus (COVID-19) pandemic: initial results for automated detection & patient monitoring using deep learning CT image analysis. arXiv preprint. arXiv:2003.05037

12. Acien A, Morales A, Vera-Rodriguez R, Bartolome I, Fierrez J (2018) Measuring the gender and ethnicity bias in deep models for face recognition. In: Iberoamerican Congress on pattern recognition. Springer, Berlin, pp 584–593

13. Lu B, Chen J-C, Castillo CD, Chellappa R (2019) An experimental evaluation of covariates effects on unconstrained face verification. IEEE Trans Biom Behav Identity Sci 1(1):42–55

14. Cook CM, Howard JJ, Sirotin YB, Tipton JL, Vemury AR (2019) Demographic effects in facial recognition and their dependence on image acquisition: an evaluation of eleven commercial systems. IEEE Trans Biom Behav Identity Sci 1(1):32–41

15. Liu H, Simonyan K, Yang Y (2018) Darts: differentiable architecture search. arXiv preprint. arXiv:1806.09055

16. Yu K, Sciuto C, Jaggi M, Musat C, Salzmann M (2019) Evaluating the search phase of neural architecture search. arXiv preprint. arXiv:1902.08142

17. Li L, Talwalkar A (2020) Random search and reproducibility for neural network search. In: Uncertainty in artificial intelligence. PMLR, pp 367–377

18. Jin H, Song Q, Hu X (2018) Auto-keras: efficient neural architecture search with network morphism, vol 5. arXiv preprint. arXiv:1806.10282

19. Kandasamy K, Neiswanger W, Schneider J, Poczos B, Xing EP (2018) Neural architecture search with Bayesian optimisation and optimal transport. In: Advances in neural information processing systems, pp 2016–2025

20. Hutter F, Hoos HH, Leyton-Brown K (2011) Sequential model-based optimization for general algorithm configuration. In: International conference on learning and intelligent optimization. Springer, Berlin, pp 507–523

21. Zoph B, Le QV (2016) Neural architecture search with reinforcement learning. arXiv preprint. arXiv:1611.01578

22. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8697–8710

23. Zhong Z, Yan J, Wu W, Shao J, Liu C-L (2018) Practical blockwise neural network architecture generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2423–2432

24. Sun Y, Xue B, Zhang M, Yen GG (2018) An experimental study on hyper-parameter optimization for stacked auto-encoders. In: 2018 IEEE Congress on evolutionary computation (CEC). IEEE, pp 1–8

25. Tanaka T, Shin ozaki T, Watanabe S, Hori T (2016) Evolution strategy based neural network optimization and lstm language model for robust speech recognition. In: CHIME 2016 workshop, pp 32–35

26. Real E, Aggarwal A, Huang Y, Le QV (2019) Aging evolution for image classifier architecture search. In: AAAI conference on artificial intelligence

27. Sun Y, Xue B, Zhang M, Yen GG, Lv J (2020) Automatically designing CNN architectures using the genetic algorithm for image classification. IEEE Trans Cybern

28. Besnassi M, Naggaz N, Benettoua A (2020) Face detection based on evolutionary Haar filter. Pattern Anal Appl 23(1):309–330

29. Wistuba M (2018) Deep learning architecture search by neurocell-based evolution with function-preserving mutations. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, Berlin, pp 243–258

30. Real E, Aggarwal A, Huang Y, Le QV (2019) Regularized evolution for image classifier architecture search. In: Proceedings of the AAAI conference on artificial intelligence, vol 33, pp 4780–4789

31. Xiao X, Yan M, Basodi S, Ji C, Pan Y (2020) Efficient hyperparameter optimization in deep learning using a variable length genetic algorithm. arXiv preprint. arXiv:2006.12703

32. Hitomi N, Selva D (2018) Constellation optimization using an evolutionary algorithm with a variable-length chromosome. In: 2018 IEEE aerospace conference. IEEE, pp 1–12

33. Fathyanaos TN, Bachtair FA, Data M (2018) Feature selection using variable length chromosome genetic algorithm for sentiment analysis. In: 2018 International conference on sustainable information engineering and technology (SIET). IEEE, pp 27–32

34. Fuad MMM (2016) Variable-chromosome-length genetic algorithm for time series discretization. In: International conference on database and expert systems applications. Springer, Berlin, pp 418–425

35. Xiao AT, Tong YX, Zhang S (2020) False-negative of RT-PCR and prolonged nucleic acid conversion in COVID-19: rather than recurrence. J Med Virol

36. Sun L, Liu G, Song F, Shi N, Liu F, Li S, Li P, Zhang W, Jiang X, Zhang Y et al (2020) Combination of four clinical indicators predicts the severe/critical symptom of patients infected COVID-19. J Clin Virol 104431

37. Wu J, Zhang P, Zhang L, Meng W, Li J, Tong C, Li Y, Cai J, Yang Z, Zhu J et al (2020) Rapid and accurate identification of COVID-19 infection through machine learning based on clinical available blood test results. medRxiv

38. Wang L, Song X (2020) COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images. arXiv preprint. arXiv:2003.09871

39. Apostolopoulos ID, Mpesiana TA (2020) COVID-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 1

40. Zhang J, Xie Y, Li Y, Chen S, Xiu Y (2020) COVID-19 screening on chest X-ray images using deep learning based anomaly detection. arXiv preprint. arXiv:2003.12338

41. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

42. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. arXiv preprint. arXiv:1412.6980

43. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: ICML

44. Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. In: International conference on machine learning, pp 1139–1147

45. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint. arXiv:1502.03167

46. Santurkar S, Tsipras D, Ilyas A, Madry A (2018) How does batch normalization help optimization? In: Advances in neural information processing systems, pp 2483–2493

47. Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489
48. Wistuba M, Rawat A, Pedapati T (2019) A survey on neural architecture search. arXiv preprint. arXiv:1905.01392
49. Elsken T, Metzen J-H, Hutter F (2018) Neural architecture search: a survey. arXiv preprint. arXiv:1808.05377
50. Xie L, Yuille A (2017) Genetic CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1379–1388
51. Suganuma M, Shirakawa S, Nagao T (2017) A genetic programming approach to designing convolutional neural network architectures. In: Proceedings of the genetic and evolutionary computation conference. ACM, pp 497–504
52. Sun Y, Xue B, Zhang M, Yen GG (2019) Evolving deep convolutional neural networks for image classification. IEEE Trans Evolut Comput
53. Real E, Moore S, Selle A, Saxena S, Suematsu YL, Tan J, Le QV, Kurakin A (2017) Large-scale evolution of image classifiers. In: Proceedings of the 34th international conference on machine learning, vol 70, pp 2902–2911. JMLR.org
54. Liu H, Simonyan K, Vinyals O, Fernando C, Kavukcuoglu K (2017) Hierarchical representations for efficient architecture search. arXiv preprint. arXiv:1711.00436
55. Elsken T, Metzen J-H, Hutter F (2017) Simple and efficient architecture search for convolutional neural networks. arXiv preprint. arXiv:1711.04528
56. Arabas J, Bartnik Ł, Opara K (2011) DMEA—an algorithm that combines differential mutation with the fitness proportionate selection. In: 2011 IEEE symposium on differential evolution (SDE). IEEE, pp 1–8
57. Grefenstette JJ et al (1992) Genetic algorithms for changing environments. In: PPSN, vol 2. Citeseer, pp 137–144
58. Chen X, Sun Y, Zhang M, Peng D (2020) Evolving deep convolutional variational autoencoders for image classification. IEEE Trans Evolut Comput
59. Cobb HG, Grefenstette JJ (1993) Genetic algorithms for tracking changing environments. Technical report, Naval Research Lab, Washington DC
60. Zhao J, Zhang Y, He X, Xie P (2020) COVID-CT-dataset: a CT scan dataset about COVID-19. arXiv preprint. arXiv:2003.13865
61. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
62. Tan M, Le QV (2019) EfficientNet: rethinking model scaling for convolutional neural networks. arXiv preprint. arXiv:1905.11946
63. Tan M, Chen B, Pang R, Vasudevan V, Sandler M, Howard A, Le QV (2019) MnasNet: platform-aware neural architecture search for mobile. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2820–2828
64. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034
65. He X, Yang X, Zhang S, Zhao J, Zhang Y, Xing E, Xie P (2020) Sample-efficient deep learning for COVID-19 diagnosis based on CT scans. medrxiv
66. Chen J, Mao H, Zhang H, Yi Z (2018) Symmetric low-rank preserving projections for subspace learning. Neurocomputing 315(13):381–393
67. Chen J, Mao H, Wang Z, Zhang X (2021) Low-rank representation with adaptive dictionary learning for subspace clustering. Knowl Based Syst 223:1–12
68. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural computation 1(4):541–551

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.