International Journal of Neural Systems, (2023) 2350026 (15 pages)
© World Scientific Publishing Company
DOI: 10.1142/S0129065723500260

A Method based on Evolutionary Algorithms and Channel Attention Mechanism to Enhance Cycle Generative Adversarial Network Performance for Image Translation

Yu Xue and Yixia Zhang
School of Software
Nanjing University of Information Science and Technology
Nanjing 210044, P. R. China

Ferrante Neri*
NICE Research Group, Department of Computer Science
University of Surrey, Stag Hill Campus
Guildford GU2 7XH, UK
f.neri@surrey.ac.uk

Accepted 26 February 2023
Published Online 5 April 2023

A Generative Adversarial Network (GAN) can learn the relationship between two image domains and achieve unpaired image-to-image translation. One of the breakthroughs was Cycle-consistent Generative Adversarial Networks (CycleGAN), which is a popular method to transfer the content representations from the source domain to the target domain. Existing studies have gradually improved the performance of CycleGAN models by modifying the network structure or loss function of CycleGAN. However, these methods tend to suffer from training instability and the generators lack the ability to acquire the most discriminating features between the source and target domains, thus making the generated images of low fidelity and few texture details. To overcome these issues, this paper proposes a new method that combines Evolutionary Algorithms (EAs) and Attention Mechanisms to train GANs. Specifically, from an initial CycleGAN, binary vectors indicating the activation of the weights of the generators are progressively improved upon by means of an EA. At the end of this process, the best-performing configurations of generators can be retained for image generation. In addition, to address the issues of low fidelity and lack of texture details on generated images, we make use of the channel attention mechanism. The latter component allows the candidate generators to learn important features of real images and thus generate images with higher quality. The experiments demonstrate qualitatively and quantitatively that the proposed method, namely, Attention evolutionary GAN (AevoGAN) alleviates the training instability problems of CycleGAN training. In the test results, the proposed method can generate higher quality images and obtain better results than the CycleGAN training methods present in the literature, in terms of Inception Score (IS), Fréchet Inception Distance (FID) and Kernel Inception Distance (KID).

Keywords: Generative adversarial networks; channel attention mechanism; image-to-image translation; evolutionary computation.

1. Introduction

The Generative Adversarial Network (GAN) is one of the most important unsupervised learning models in Deep Learning.[1–11] GANs were originally developed as a powerful generative model that learns the distribution of real data and learns to generate realistic

[*] Corresponding author.

data. In general, the models learn to play off each other through (at least) two modules in the framework: the generator and the discriminator. The generator aims to generate high-quality samples to fool the discriminator, and the discriminator aims to distinguish whether the data are real or generated. Since the first proposal by Goodfellow *et al.*,[12] numerous GAN variants have been applied to various data types, such as speech,[13] images[14] and videos,[15] etc. Whereby, image-related variants of GANs have made great progress in many subtasks, such as image generation,[16] image inpainting,[17] super-resolution,[18,19] segmentation,[20] image-to-image translation,[21,22] etc.

Image-to-image translation, i.e. the task of taking images from one domain and transforming them into images that have the style of images from another domain, is the focus of this work. With the subtask set of image-to-image translation, one of the most widely known GAN variants is Cycle-consistent GAN (CycleGAN).[23] CycleGAN architecture consists of two GANs in a loop and is associated with a loss function that measures how much the transformed image is similar to the original image but with new features. Unfortunately, this translation process is usually unstable and often leads to some unsuccessful cases. Therefore, several studies on CycleGAN targeted the enhancement of the stability of the translations. For example, Zhang *et al.*[24] applied smooth regularization on its overall objective function to enforce smoothness–consistency between the source and target domains, halving the mean-squared error. In Geometry-consistent GAN (GcGAN),[25] their strategy solved the problem of failed geometric transformation of the target domain. Victor Schmidt *et al.*[26] used auxiliary tasks, i.e. rotation prediction, and generative colorization, to improve the visual performance. Xu *et al.*[27] introduce two adversarial training components for learning the spatial perturbation function to obtain the alignment of the distribution between the source and target domain. In addition, structure consistency constraint (SCC)[28] has been proposed to reduce the randomness of color transformation in the translation process. In terms of the limitation of intra-domain or deterministic inter-domain translation in image-to-image many approaches, Mao *et al.*[29] introduce a unified attribute space to enhance the visual quality of continuous translation results.

Although these methods improve upon the quality of the generated images they often generate either images with overly saturated colors compared to the input images or still have obvious visual deficiencies, such as incomplete translation, missing details and low fidelity. To overcome this difficulty, some researchers incorporated attention mechanisms[30] (a pre-processing algorithm to identify the important parts of the input) as a way to improve the quality of generated images. For example, Zhang *et al.*[31] placed the self-attention mechanism as a condition on the generators and showed its validity of extracting the key information on objective images. A number of studies[32,33] have also applied attention mechanisms to alleviate the above image-quality problems with limited datasets and for specific tasks.

Furthermore, these methods often suffer from training instability, that the fluctuation of loss function during the network training. The latter is a well-known issue in GANs' training, see e.g. Ref. 34. A popular way to mitigate this issue is to introduce redundancies by processing multiple candidate networks within the same training framework. Inspired by Evolutionary Computation (EC), Wang *et al.*[35] evolve a population of generators to play the adversarial game with the discriminator. Shu *et al.*[36] devised a mask-encoding on generators and applied an Evolutionary Algorithm (EA) to obtain generators with smaller model sizes and better generative results, their research focuses more on model compression. These approaches successfully address the problem of training instability. However, the problem of image quality tends to persist for these methods.

This paper proposes an approach, namely, Attention evolutionary GAN (AevoGAN), that combines the advantages of EC[37–46] and attention mechanisms to simultaneously overcome the problems of training instability and the deficiencies of the generated images. The following are the main contributions of this paper:

- An EA is applied to identify promising generators by compressing an initial CycleGAN structure. The EA compresses a CycleGAN structure by evolving a population of binary vectors whose 0 and 1 indicate active and inactive weights of the initial CycleGAN. Furthermore, this study proposes a novel loss function for the EA to search for a compressed CycleGAN which also has a better performance than the initial one.

- The channel attention (CA) mechanism is used to assist the generators to find and enhance the influential features of input images.

The remainder of this paper is organized in the following sections. Section 2 briefly introduces some related works about EAs and attention models combined with GANs. Section 3 describes the proposed AevoGAN in detail. Section 4 provides the experimental setup and shows the results of the experiments carried out in this study. Section 5 provides the conclusion to this work.

2. Related Work

Several studies have been performed in the past years to improve upon the performance of GANs.[47–53] In this section, we review relevant work closely related to this work, mainly in the areas of evolutionary GANs, unsupervised image-to-image translation, and attention-based GANs.

2.1. *Evolutionary GANs*

Many methods combining EC and GANs have been proposed and are here referred to as evolutionary GANs. Early studies on evolutionary GANs can be roughly divided into two categories: (1) variants based on loss functions; (2) variants based on network structure. The first category includes E-GAN,[35] which applies three variation operations to generator losses to evolve generators and subsequent work[54] modified the E-GAN framework to enhance its performance. Regarding the second category, Shu *et al.*[36] developed mask coding on generators using an EA to obtain generators with smaller model sizes and better generative results. Costa *et al.*[55] remove the randomly selected layers on the generator and discriminator, their work achieves a lower FID score while Mehta *et al.*[56] applied different mutation options to neural networks to generate images with better resolution.

In recent years evolutionary GANs embedded multi-objective problem formulations and neural architecture search (NAS) to offer attractive GANs. He *et al.*[57] proposed a multi-objective EA driven by GANs to improve the performance of various machine-learning models under the condition of limited training data. EAGAN[58] uses a two-stage EA based on a NAS framework to design GANs, and it uses multiple objectives to store nondominating architectures. Lin *et al.*[59] use different objective functions to train various candidate architectures in the generator, leading to more stabilized and better-performing GANs.

2.2. *Unsupervised image-to-image translation*

Image-to-image translation involves converting images from one domain into a new image that represents the original image but has the style (or characteristics) of images from another domain. For example, an image-to-image translation can transform photos into paintings according to a predetermined pictorial style. The unsupervised image-to-image translation is the task of performing image-to-image translation without ground truth image-to-image pairing. Recent studies of unsupervised image-to-image translation modify the network structure and loss function to improve performance. For example, GANILLA[60] proposes a new generator that uses additive connections between low and high-level features, including skip connections. NICE-GAN[61] changed the role of its discriminator by embedding the encoder into the discriminator. Zhao *et al.*[62] replaced the cycle-consistent loss with adversarial consistency loss to optimize the whole model and Park *et al.*[63] proposed patch-wise contrastive loss on mutual features to improve the performance of translation.

2.3. *Attention-based GANs*

Model networks often add attention mechanisms[64–66] to automatically learn and adjust the contribution of features in the input data. Zhang *et al.*[31] introduced the use of the attention mechanism in GANs. It is shown that the attention model helps the generator to generate more detailed images. Due to the effectiveness of the attention mechanism and its flexibility, it is portable on most models, Yang *et al.*[67] devised bidirectional attention GANs to effectively obtain the spatial characteristics of traffic flow. Moreover, the self-attention module has been applied on GANs to extract the internal damage of input data.[32] MA-GAN[33] combines multi-scaled convolution and CA to automatically learn and adjust the scale of residuals for better representation.

Besides the CA mechanism, the spatial attention (SA) mechanism, that is a module isolating important parts of an image using the inter-spatial relationship of features, has been proposed.[68] The SA mechanism has recently been applied, with success, to GANs for image translation.[69,70]

2.4. *CycleGAN loss function*

CycleGAN consists of two GANs that have the same network structures. Each of them is composed of a generator G_1, G_2, respectively, and a discriminator D_1, D_2, respectively. Let us consider the source images from two different styles $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_m\}$, e.g. a set of images of horses and a set of zebras, assuming that horses and zebras look alike except for their color patterns. In the following, when we write G_1, G_2, D_1 and D_2, we refer to the parameters of the corresponding network. Let us consider for the time being the GAN identified by G_1 and D_1. We may think about G_1 and D_1 as the following functions:

$$\begin{aligned} &G_1(x) : X \to Y;\\ &D_1(y) : Y \to (0, 1]. \end{aligned}$$

This means that G_1 transforms an image (which can be noise) into a new fake image while D_1 processes the fake image assigning a score to it. A score nearing 0 means that the image is likely to be fake while a score nearing 1 means that the image is assessed to be genuine.

The training of the $\mathcal{L}_{\mathrm{GAN}}(G_1, D_1)$ can then be formulated as the optimization of the following loss function[12]:

$$\begin{aligned} &\mathcal{L}_{\mathrm{GAN}}(G_1, D_1)\\ &\quad = \mathbb{E}_y[\log D_1(y)] + \mathbb{E}_x[\log(1 - D_1(G_1(x)))], \end{aligned} \tag{1}$$

where $\mathbb{E}_y$ and $\mathbb{E}_x$ represent the expectation that fake and genuine images have been correctly detected. The loss function in Eq. (1) has to be minimized for G_1 and maximized for D_1.

Since CycleGAN contains two generators and discriminators, the training objective function for CycleGAN is[23]

$$\begin{aligned} \mathcal{L}_{\mathrm{CycleGAN}} = {} & \mathcal{L}_{\mathrm{GAN}}(G_1, D_1) + \mathcal{L}_{\mathrm{GAN}}(G_2, D_2)\\ & + \lambda_{\mathrm{cyc}} \mathcal{L}_{\mathrm{cyc}}(G_1, G_2), \end{aligned} \tag{2}$$

where λ_{cyc} is a hyper-parameter and $\mathcal{L}_{\mathrm{cyc}}$ is a function called cycle-consistency loss whose minimization ensures that the generated images $G_1(x)$, $G_2(y)$ are as similar as possible to the source images x, y. The mathematical formulation of cycle-consistency loss is given by

$$\begin{aligned} \mathcal{L}_{\mathrm{cyc}}(G_1, G_2) = {} & \mathbb{E}_x[\|G_2(G_1(x)) - x\|_1]\\ & + \mathbb{E}_y[\|G_1(G_2(y)) - y\|_1], \end{aligned} \tag{3}$$

where the 1-norm operator $\| \cdot \|_1$ indicates the sum of the absolute values. The cycle-consistency loss calculates the differences between each image input to G_1, i.e. x and the corresponding image output by G_2 and analogously for the inputs to G_2. By following the classical example of CycleGAN, if we convert a zebra image to a horse image and then back to a zebra image, we may expect to have the original zebra image. If this is the case, then the cycle-consistency loss is zero. More details about the CycleGAN loss function are available in Ref. 23.

3. AevoGAN: Attention Evolutionary Generative Adversarial Network

This section describes the proposed AevoGAN including its encoding, the evolutionary framework and the use of the attention model.

3.1. *Encoding of CycleGAN*

For simplicity of notation, we indicate with G either the generators (either G_1 or G_2) and D either the discriminators (either D_1 or D_2) and refer to them as generator and discriminator, respectively. This simplification is done only because the mathematical description for (G_1, D_1) and (G_2, D_2) are analogous.

Let us assume that the generator G contains L convolution layers and let us denote its filter weights with $\{\mathbf{W}_i^G\}_{i=1}^L$ where $\mathbf{W}_i^G$ indicates the three-dimensional (3D) tensor containing the weights' matrices of the generator G at the convolution layer i.

The generator G is encoded by means of the binary data structure[71] of the same dimensions of $\{\mathbf{W}_i^G\}_{i=1}^L$ and indicated by $\mathbf{S}^G = \{\mathbf{S}_i^G\}_{i=1}^L$ where $\mathbf{S}_i^G$ contains the activation information of the weights within the ith convolution layer 0 means that the weight is not active (it is OFF) while 1 means that the weight is active (it is ON). Thus, obtaining sparse weights is directed to cancel the filters in the channels (red, green and blue, respectively). For each convolution layer, the weights of G are updated according to the formula

$$\widehat{\mathbf{W}_i^G} = \mathbf{S}_i^G \odot \mathbf{W}_i^G, \tag{4}$$

where $\odot$ indicates the element-by-element product, $\widehat{\mathbf{W}_i^G}$ is 3D tensor containing the updated weights of the generator G at the layer i.

The binary data structure $\mathbf{S}^G$ is arranged as a (long) binary vector representing the active and inactive weights of one of the CycleGAN generators, which, for the sake of simplicity, we will also indicate with $\mathbf{S}^G$.

3.2. *Evolutionary framework to design the compressed generators*

For a given CycleGAN with its two generators, G_1 and G_2, the following procedure has been applied to generate a new compressed CycleGAN with its two generators $\widehat{G_1}$ and $\widehat{G_2}$. The two generators $\widehat{G_1}$ and $\widehat{G_2}$ are processed separately by two optimization runs.

At the beginning of the optimization process, a population of K binary vectors of the type $\mathbf{S}^G$ are randomly sampled. From each $\mathbf{S}^G$, the corresponding generator $\widehat{G_1}$ (or $\widehat{G_2}$) is calculated by applying Eq. (4). For each element of the population, the sparse-weight network corresponding to it is trained on a set of training images sampled from the entire data set, and the fitness on the validation set is used as the evaluation criterion. To assess their quality, the corresponding sparse-weight (compressed) CycleGAN $(\widehat{G_1}, D_1, \widehat{G_2}, D_2)$ is trained on a subset of the training images.

For the compressed generator $\widehat{G_1}$, the following function is calculated

$$\mathcal{F}_{d1} = \left[\frac{1}{n}\sum_{i=1}^{n} \|D_1(G_1(x_i)) - D_1(\widehat{G_1}(x_i))\|_2^2\right]^{-1}, \tag{5}$$

while for the generator $\widehat{G_2}$ the following function is calculated

$$\mathcal{F}_{d2} = \left[\frac{1}{m}\sum_{i=1}^{m} \|D_2(G_2(y_i)) - D_2(\widehat{G_2}(y_i))\|_2^2\right]^{-1}, \tag{6}$$

where the 2-norm $\|\|_2$ is the square root of the squares of the elements of the vector therein. The functions $\mathcal{F}_{d1}$ and $\mathcal{F}_{d2}$ measure the deviation in performance of the candidate (compressed) generators $\widehat{G_1}$ and $\widehat{G_2}$ with respect to the original generators G_1 and G_2. Similar performance results into high values of $\mathcal{F}_{d1}$ and $\mathcal{F}_{d2}$. Since the performance in $\mathcal{F}_{d1}$ and $\mathcal{F}_{d2}$ is assessed by means of the discriminators we refer to it as "discriminators' fitness" and we indicate this fact with the subscript d.

In addition, the cycle-consistency modified from Eq. (4) is also taken into consideration. For $\widehat{G_1}$ and $\widehat{G_2}$, the following functions are calculated, respectively,

$$\mathcal{F}_{\text{cyc1}} = \lambda_{\text{cyc}}\left[\frac{1}{n}\sum_{i=1}^{n} \|G_2(\hat{G}_1(x_i)) - x_i\|_2^2\right]^{-1}, \tag{7}$$

$$\mathcal{F}_{\text{cyc2}} = \lambda_{\text{cyc}}\left[\frac{1}{m}\sum_{i=1}^{m} \|G_1(\hat{G}_2(y_i)) - y_i\|_2^2\right]^{-1}. \tag{8}$$

The functions $\mathcal{F}_{\text{cyc1}}$ and $\mathcal{F}_{\text{cyc2}}$ in Eqs. (7) and (8) signify how successful the generators are to transform an image and return back to the original one. As for $\mathcal{F}_{d1}$ and $\mathcal{F}_{d2}$, high performance of the candidate solutions correspond to high values of $\mathcal{F}_{\text{cyc1}}$ and $\mathcal{F}_{\text{cyc2}}$.

The overall fitness functions to evaluate each individual $\mathbf{S}^G$ in the corresponding population and to maximize are

$$\mathcal{F}_1 = \mathcal{F}_{d1} + \mathcal{F}_{\text{cyc1}} \tag{9}$$

and

$$\mathcal{F}_2 = \mathcal{F}_{d2} + \mathcal{F}_{\text{cyc2}}. \tag{10}$$

In the case of $\widehat{G_1}$, the fitness $\mathcal{F}_1$ (or $\mathcal{F}_2$ in the case of $\widehat{G_2}$) is calculated for all K candidate solutions and used to apply the fitness proportionate selection.[72] The kth individual of the population is associated with the probability p_1^k:

$$p_1^k = \frac{\mathcal{F}_1^k}{\sum_{k=1}^{K}\mathcal{F}_1^k}, \tag{11}$$

where $\mathcal{F}_1^k$ is the fitness of the kth candidate solution. Analogously for $\widehat{G_2}$, the probabilities $p_2^k = \frac{\mathcal{F}_2^k}{\sum_{k=1}^{K}\mathcal{F}_2^k}$ are calculated.

The parents are then selected according to the probabilities in Eq. (11) and undergo crossover and mutation to generate K offspring individuals. In this phase, we made use of binary encoding to effectively explore the search space with simple operators. We used the two-point crossover and a flip mutation ($0 \rightarrow 1$ or $1 \rightarrow 0$) applied to two randomly selected genes (binary bits).[72] We use the thresholds t_{cr} and

t_{mut} to apply crossover to a pair of parents and generate two offspring solutions or to apply mutation to one individual of the population, see Algorithm 1 for details. We chose this logic, on the basis of our preliminary results, to directly control the exploitation and exploration of the algorithm. We have observed that the logic outlined in Algorithm 1 is in this case more effective than that of a standard EA that applies mutation to the newly generated offspring. It must be remarked that similar implementations of EAs for GAN have also been recently proposed in the literature.[36,59]

Algorithm 1. Evolutionary algorithm of AevoGAN to produce an optimized $\widehat{G_1}$.

```
Require: Pre-trained generator $G_1$. Pre-defined hyper-parameters $K$, $T$, crossover threshold $t_{\text{cr}}$ and mutation threshold $t_{\text{mut}}$.
Initialize $K$ individuals of the type $\mathbf{S}^G$, each of them containing the information of $\widehat{G}_1$;
for $t$=1 to $T$ do
    Calculate the fitness of each individual in $\widehat{G}_1$ by Eq. (9): $\mathcal{F}_1\left(\widehat{G}_1^{(t)}\right) \leftarrow \mathcal{F}_{d1}^{(t)} + \mathcal{F}_{\text{cyc1}}^{(t)}$;
    Assign selection probabilities to all individuals by (Eq. (11));
    Update the best individual (elite) if the best individual of the population has a higher performance than the historical best: $\widehat{G}_1^{(t)} \leftarrow \widehat{G}_1^{(t-1)}$;
    if $k = 1$ to $K$ then
        Sample a random seed $r \in [0, 1]$;
        if $r < t_{\text{cr}}$ then
            According to the selection probabilities, select an individual to remain in the next generation;
            $k = k + 1$;
        else if $t_{\text{cr}} \leq r < t_{\text{mut}}$ then
            According to the selection probabilities, select two individuals and apply the crossover operator on the selected parents;
            $k = k + 2$;
        else
            According to the selection probabilities, select an individual and apply the mutation operator (flips two bits) on it;
            $k = k + 1$;
        end if
    end if
    Replace the population with the offspring population for the subsequent generation;
end for
Output: Best generator $\widehat{G}_1^{(T)}$.
```

An elitism of size 1 is used to preserve at each generation the best individual. Although this individual does not participate in the generation we chose to store in memory the best results to not lose them. The newly generated offspring solutions replace the population and are processed in the subsequent generation. This loop is repeated for T generations.

Algorithm 1 displays the pseudo-code of the evolutionary framework for $\widehat{G_1}$ where we use the generation index t. The pseudocode for $\widehat{G_2}$ would be analogous.

Figure 1 depicts the evolutionary process for identifying an optimized $\widehat{G_1}$. The evolutionary process for $\widehat{G_2}$ is analogous.

3.3. *Training of the CycleGAN with compressed generators*

The CycleGAN is then built with the best generators $\hat{G}_1$ and $\hat{G}_2$ produced by the EA in Algorithm 1 and the original discriminators D_1 and D_2. This newly designed CycleGAN undergoes adversarial training to modify the values of the active weights of $\hat{G}_1$ and $\hat{G}_2$. This training is performed by using the original $\mathcal{L}_{\text{CycleGAN}}$ shown in Eq. (2). In addition, we propose the introduction of the following identity loss function:

$$\mathcal{L}_{\text{iden}}(\hat{G}_1, \hat{G}_2) = \frac{1}{n}\sum_{i=1}^{n} |G_1(x) - \hat{G}_1(x)| + \frac{1}{m}\sum_{i=1}^{m} |G_2(y) - \hat{G}_2\ (y)|. \quad (12)$$

The meaning of the $\mathcal{L}_{\text{iden}}$ function is the distance between the original generator and its sparse version. The training of the resulting CycleGAN is performed by optimizing in adversarial training, i.e. minimizing for generator and maximizing for discriminator, with Adam,[73] the function

$$\mathcal{L}_{\text{AevoGAN}} = \mathcal{L}_{\text{CycleGAN}} + \lambda_{\text{iden}}\mathcal{L}_{\text{iden}}(\hat{G}_1, \hat{G}_2), \quad (13)$$

where λ_{iden} is a pre-defined hyper-parameter.

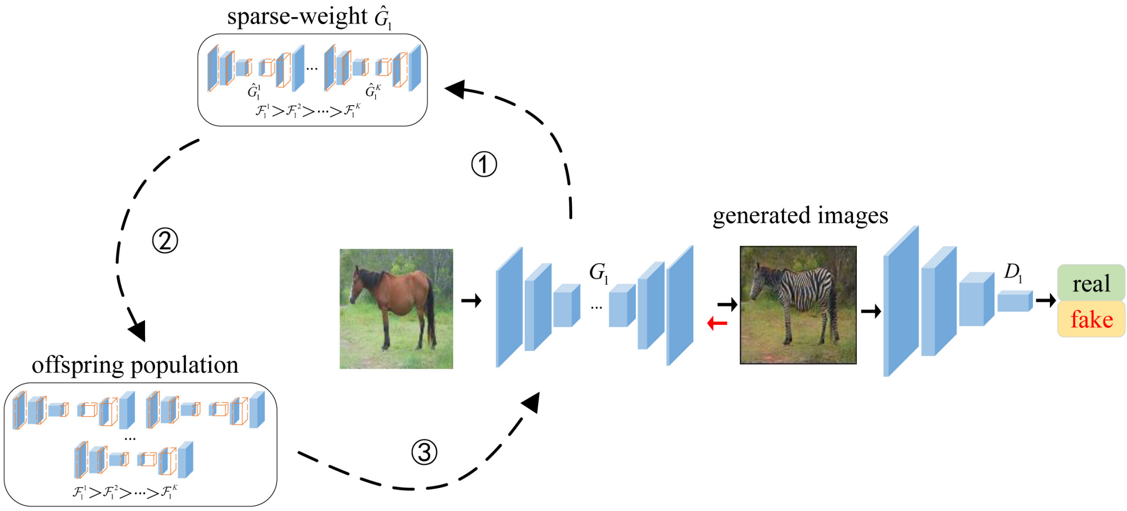

sparse-weight $\hat{G}_1$
$\hat{G}_1^1$
$\hat{G}_1^K$
$\mathcal{F}_1^1 > \mathcal{F}_1^2 > \cdots > \mathcal{F}_1^K$
①
②
③
generated images
G_1
D_1
real
fake
offspring population
$\mathcal{F}_1^1 > \mathcal{F}_1^2 > \cdots > \mathcal{F}_1^K$

Fig. 1. The evolutionary process to identify the generator $\widehat{G_1}$. Multiple generators $\widehat{G_1}$ are tested to replace G_1. The evolution over T generations is schematically represented as a single cycle producing a new generator. This cycle is composed of three sections: (1) from G_1 a population of sparse-weight $\widehat{G_1}$ is generated according to Eq. (4); (2) evolutionary operators are applied to produce candidate offspring solutions; (3) the most promising solutions are selected over the generations.

3.4. *Channel attention-based generators*

In deep neural networks,[10,74–81] many channels are usually generated by convolution operation, but some feature information in some channels has little influence. CA mechanism has emerged as an effective tool to capture the key features from a large feature space.[82] Leveraging this insight, we apply the CA mechanism[83] on the intermediate channels of the generators. In practice, we add the CA layer after the residual block that has the largest number of features. Then we perform max-pooling and average-pooling operations on these features to enhance the extracted features with significant effect in the channels. The effective feature extraction and enhancement process by CA is illustrated in Fig. 2.

For an input image data $F \in R^{C*H*W}$, the max pooling and average pooling layers change the feature map size from $C * H * W$ to $1 * 1 * C$. Then the two feature maps are put into a shared Multi-layer Perceptron (MLP) module and ReLU activation function. Finally, the output result of CA (i.e. F_c) is given as

$$\begin{aligned} F_c &= \sigma[\mathrm{MLP}(\mathrm{avg\_pool}(F)) + \mathrm{MLP}(\mathrm{max\_pool}(F))] \\ &= \sigma[\mathrm{MLP}(f_{\mathrm{avg}}) + \mathrm{MLP}(f_{\max})] \\ &= \sigma[\omega_2(\omega_1(f_{\mathrm{avg}})) + \omega_2(\omega_1(f_{\max}))] \\ &= \bar{f}_{\mathrm{avg}} + \bar{f}_{\max}, \end{aligned} \tag{14}$$

where f_{avg} and $f_{\max}$ are average and max pooling on input image F, respectively. MLP is a multi-layer perceptron, ω_1 contains the weights between the input layer and hidden layer of MLP, and ω_2 is the weights between the hidden layer and output layer, σ is the sigmoid activation function. By multiplying the output of CA with the input image F, we can get the feature map with the same size as the input

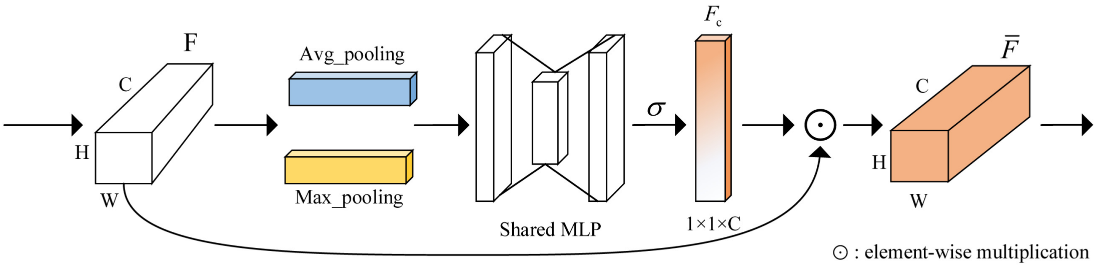


Fig. 2. The extraction and enhancement process of input feature.

feature $C * H * W$, i.e.

$$\bar{F} = F_c \odot F, \tag{15}$$

where $\odot$ denotes element-wise multiplication.

With the implementation of CA above, generators are able to generate images that retain more critical details, thus enhancing the overall visual performance. It is worth noting that the CA in the generator is not involved in the evolutionary process in our strategy, in case the key features from CA weights be substituted in the evolutionary process.

4. Experiments

To evaluate the proposed AevoGAN, we conduct experiments on the horse2zebra dataset[23] for image translation. In order to explore the potential effect of the SA mechanism on the proposed framework, we display the results of AevoGAN with and without SA. In this section, we compare CycleGAN, AevoGAN+ SA, and AevoGAN and perform an ablation study to evaluate the stability and performance of AevoGAN. The code is available at https://github.com/lizzhang-spec/AevoGAN.

4.1. *Implementation details*

We evaluate AevoGAN on horse2zebra dataset, which consists of 1187 horse and 1474 zebra images. Among these images, 10% of them are used as a test set. To guarantee a fair comparison, we used the network architecture of the original CycleGAN directly. To obtain better performance on generating 256×256 color images (thus each image is the composition of three 256×256 images in rgb), the CA mechanism is added to the layer with the largest number of channels in the generator convolution operation. The details of the generator are listed in Table 1. As for the discriminator, we also adopted the design in CycleGAN, which consists of five convolution layers and a fully connected network (FCN) layer. The output of the discriminator is a probability between 0 and 1 to indicate if the image is real or fake. It should be noted that the CA layer in the generator is not part of the evolutionary process so the important weights are not replaced during the evolution. In the evolutionary process of AevoGAN, the size of each individual in the number of bits is 640, the population size K is set to 32, and the maximum number of generations T is 100. The crossover threshold t_{cr} has been set equal to 0.2, mutation threshold t_{mu} is 0.9. With reference to Eqs. (2) and (13), λ_{cyc} and λ_{iden} are set equal to 10 and 5, respectively.

Table 1. The detailed design of generator.

Generator network	
Input: Real images of horses or zebras, $(3 \times 256 \times 256)$	
Operator	Number
Conv (7×7, stride 1)	×1
Conv (3×3, stride 2)	×2
Residual Block	×9
*Channel Attention	×1
Transposed Conv (3×3, stride 2)	×2
Conv (7×7, stride 1)	×1
Output: Generated images, $(3 \times 256 \times 256)$	

We conducted further experiments based on AevoGAN with the addition of the SA mechanism. Specifically, SA was used in the layer immediately after CA, it takes the output of CA with the same pooling operations (i.e. max-pooling and average-pooling) as CA to compress the channel dimension while fixing the spatial dimension. The resulting algorithm is indicated here as AevoGAN+SA.

4.2. *Experimental results and analysis*

Ablation Study. With the purpose of demonstrating the role of each novel component of AevoGAN, we have studied the performance of the method after the removal CA and the evolutionary process (Evo) separately. The generated images obtained by the methods in this ablation study are shown in Fig. 3. It can be easily observed that the generated results of AevoGAN without CA (AevoGAN-CA) and AevoGAN without evolutionary process (AvoGAN-Evo) are suffering from more defects. It can be observed that the head features of the animals in the four lower right images are incomplete or blurred, and the backgrounds in the upper right four images are replaced, or the animal's back or tail features are missing.

Hyperparameters. The cycle-consistency term λ_{cyc} is set as 10 as in the case of the original

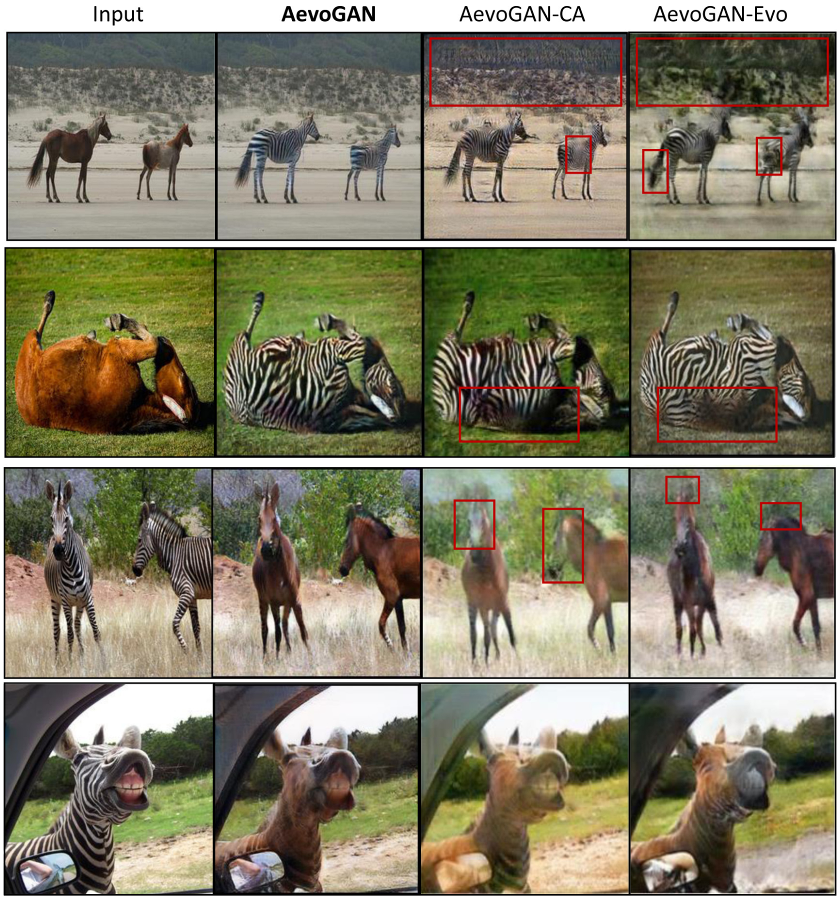


Fig. 3. Generated images of ablation study on horse2zebra datasets. The red rectangles emphasize the differences in image transformation quality, e.g. in terms of resolution and the presence/absence of details.

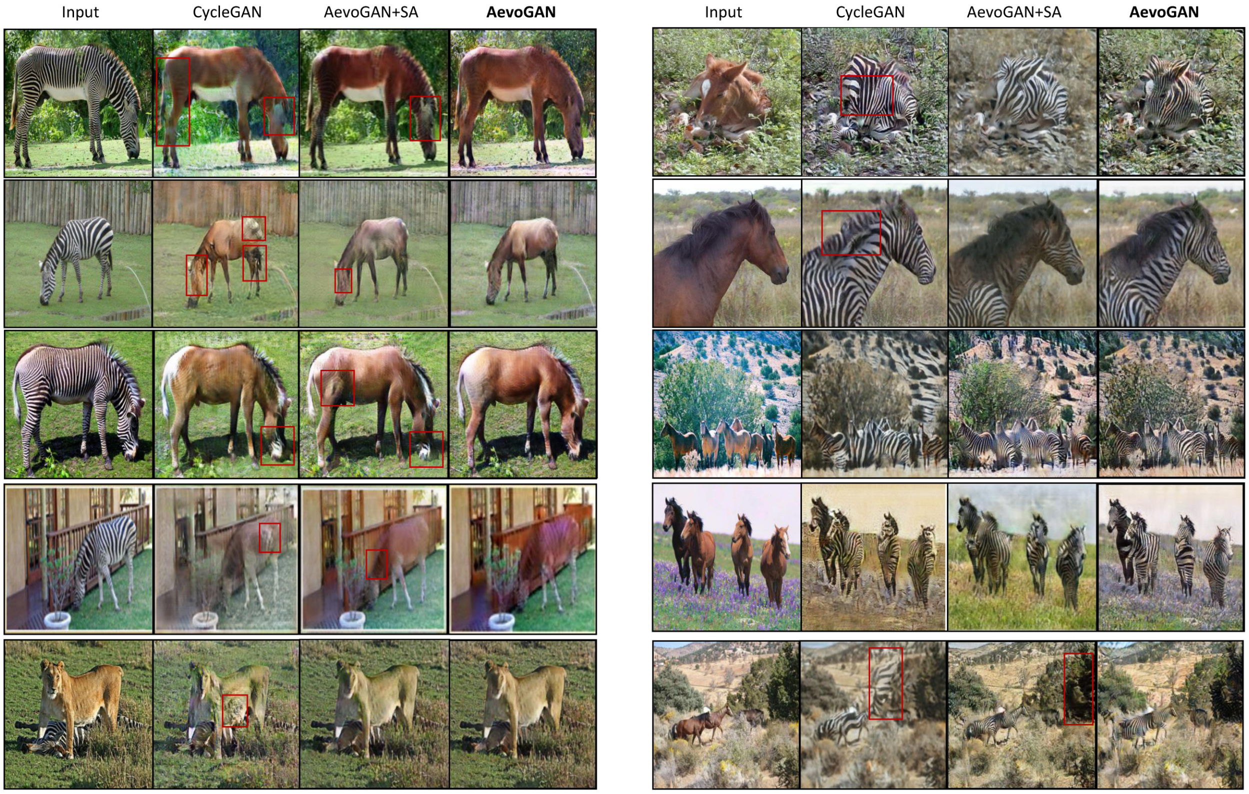
Table 2. Impact of λ_{iden} on horse2zebra datasets (FID comparison).

λ_{iden}	0.1	1	5	10
Horse → Zebra	82.85 ± 0.82	117.33 ± 0.64	$\mathbf{62.11 \pm 0.55}$	243.72 ± 0.76
Zebra → Horse	148.81 ± 1.5	153.15 ± 0.91	$\mathbf{134.37 \pm 1.12}$	244.48 ± 1.27

CycleGAN. In addition, we investigate the impact of the identity loss term λ_{iden} in Eq. (13). We observe the proposed approach achieves the best FID result when $\lambda_{\text{iden}} = 5$ (Table 2). The population size $K = 32$, crossover threshold $t_{\text{cr}} = 0.2$, mutation threshold $t_{\text{mu}} = 0.9$, and max generation $T = 100$ are selected.

Experiments on Horse2zebra Datasets. Figure 4 shows the comparison of the images generated by CycleGAN, AevoGAN+SA and the proposed AevoGAN. Clearly, the results of CycleGAN, and AevoGAN+SA show more defects in the areas highlighted by a red rectangle. It must be observed that the head features of the horse/zebra in the generated images are clearer on the AevoGAN method. Furthermore, in images with extensive objectives such as trees, walls, and other animals, AevoGAN shows superior performance.

Moreover, following the experiments on horse2zebra images, we adopted quantitative evaluation metrics to verify the validity of the proposed method. Specifically, Inception Score (IS)[84] and Fréchet



Zebra to Horse

Horse to Zebra

Fig. 4. (Color online) The results of experiments on horse2zebra datasets. The generated images by methods CycleGAN, AevoGAN + SA, and AevoGAN are presented sequentially from left to right. The red boxes highlight some details of their difference. The images generated by AevoGAN displayed in the last column appear of better quality than the others.

Table 3. Results of IS and FID on CycleGAN, AevoGAN+SA, AevoGAN. IS is a measure of the clarity and diversity of the generated images. A higher value is better. For FID metric, it reflects the distance between the generated image and the real image. A lower value is better.

Method	Horse → Zebra (IS)	Zebra → Horse (IS)
CycleGAN	1.26 ± 0.12	3.15 ± 0.04
CycleGAN +identity_loss	1.51 ± 0.03	3.11 ± 0.13
AevoGAN + SA	1.37 ± 0.05	3.03 ± 0.06
AevoGAN	$\mathbf{1.67 \pm 0.09}$	$\mathbf{3.37 \pm 0.11}$
Method	Horse → Zebra (FID)	Zebra → Horse (FID)
CycleGAN	65.15 ± 0.62	139.56 ± 0.91
CycleGAN +identity_loss	78.7 ± 1.5	137.83 ± 0.78
AevoGAN + SA	75.60 ± 0.82	142.37 ± 1.21
AevoGAN	$\mathbf{62.11 \pm 0.55}$	$\mathbf{134.37 \pm 1.12}$

inception distance (FID)[85] are employed to evaluate the images generated by CycleGAN, AevoGAN, and AevoGAN+SA. For the sake of completeness, we report the quantitative results for both the CycleGAN versions presented in Ref. 23, that is the standard CycleGAN and its modified version which also uses an identity loss during training. The latter version is here indicated as CycleGAN + identity_loss. The quantitative results are listed in Table 3 and clearly show that the proposed AevoGAN consistently achieves the best results.

To present the performance of the proposed AevoGAN against modern image translation methods representing the state-of-the-art, we have used Kernel Inception Distance (KID).[86] The latter is by far the most popular index for assessing the quality of image translations. Table 4 lists the KID results for AevoGAN, AevoGAN+SA, CycleGAN and five GAN-based algorithms recently proposed in the literature. We observe that the proposed AevoGAN outperforms the seven competitors included in this

Table 4. The result of KID ×100 ± std. ×100 on different methods. KID controls the distance between the generated and input image, for this metric, a lower value is better.

Method	Horse → Zebra	Zebra → Horse
UVCGAN[87]	4.72 ± 0.29	6.52 ± 0.34
UAIT[88]	6.93 ± 0.27	8.87 ± 0.26
U-GAT-IT[89]	7.06 ± 0.8	7.47 ± 0.71
CycleGAN[23]	10.25 ± 0.25	11.44 ± 0.38
UNIT[90]	11.22 ± 0.24	13.63 ± 0.34
DiscoGAN[91]	13.68 ± 0.28	16.60 ± 0.50
AevoGAN + SA	8.63 ± 0.13	10.26 ± 0.32
AevoGAN	$\mathbf{3.03 \pm 0.64}$	$\mathbf{5.48 \pm 0.42}$

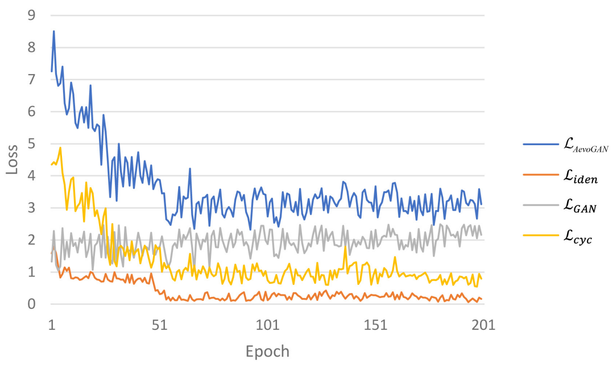


Fig. 5. The variations of the loss function during adversarial learning, see Eqs. (2) and (12), on AevoGAN.

study for both translations Horse → Zebra and Zebra → Horse. Thus, we can conclude that AevoGAN can help generate more satisfactory images.

Experimental Analysis. As introduced in Sec. 3, we used a binary data structure to encode the filters of the generator network, so as to retain the promising features and set unimportant filters to 0. Then we applied the GA only to the generators. The role of GA is to preserve the best generator in each domain according to their fitness. The role of the residual blocks in the generators is to embed features while the role of transposed convolution blocks is to generate the target domain images from the embedded features. Without this evolutionary process, the generated images show inferior performance as illustrated in Fig. 3.

To illustrate that the proposed method can stabilize the GAN training, we show the variation of the total loss function $\mathcal{L}_{\text{AevoGAN}}$, which includes the convergence loss of $\mathcal{L}_{\text{iden}}$, $\mathcal{L}_{\text{GAN}}$, $\mathcal{L}_{\text{cyc}}$, Fig. 5 displays the three trends separately. We observe that the proposed method ensures a stable convergence around the 50th epoch of adversarial learning.

In addition, a CA mechanism is also added to the generator network. The location of attention mechanisms added in the network model has been a question worthy of study. In this paper, we added the CA mechanism to the previous layer of the transposed convolution layer because we can enhance more features by CA on the layer with the highest number of features. As for the additional implementation of SA based on our method, we noticed that the feature enhancement on spatial dimension is kind of counterproductive to generate better images, as the results showed in Fig. 4. According to the experiments in Ref. 68, the combination of CA and SA mechanism is more suitable for the object detection task.

Runtime. The proposed method uses a binary data structure to encode active and active weights of the generator and then applies an evolutionary approach to preserve the most promising configurations. The whole training time of the evolutionary process costs 4 GPU days while the runtime of adversarial training between generators and discriminators is about 1 GPU day, taking a total of 5 GPU days. The original CycleGAN takes 3 GPU days to train. These figures refer to an RTX2080Ti. It must be remarked that the algorithm can be easily parallelized on standard multi-core machines, thus resulting in an actual calculation time of a few hours.

5. Conclusion and Future Work

A new evolutionary generative adversarial network (AevoGAN) for unpaired image-to-image translation is presented in this paper. To improve the generative performance, we used a binary data structure to encode the filters of generators and then applied an EA to evolve a population of generators by initialization, selection, evaluation, and variation, so as to preserve the best-performing generator in each translation domain. Experiments show that the implementation of the proposed approach stabilizes the adversarial training between generators and discriminators, and thus the preserved generators can generate images with higher quality. At the same time, to obtain important features on the real images, the generator networks in AevoGAN have a built-in CA mechanism, which is an auxiliary implementation to improve the model's performance. Then the evolutionary process and channel mechanism layers are combined to generate the target images. Experiments showed that the proposed AevoGAN outperforms CycleGAN and other modern algorithms in horse2zebra translation tasks. However, since our method is relatively time-consuming, future work will focus on mechanisms to reduce training time.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61876089, 61876185 and 61902281), the Natural Science Foundation of Jiangsu Province (No. BK20141005), and partially supported by the project of Distinguished Professors of Jiangsu Province.

References

1. Y. LeCun, Y. Bengio and G. Hinton, Deep learning, *Nature* **521**(7553) (2015) 436–444.
2. S.-H. Wang and Y.-D. Zhang, Advances and challenges of deep learning, *Recent Pat. Eng.* **17**(4) (2023) 1–2.
3. E. Macias-Garcia, D. Galeana-Perez, J. Medrano-Hermosillo and E. Bayro-Corrochano, Multi-stage deep learning perception system for mobile robots, *Integr. Comput.-Aided Eng.* **28**(2) (2021) 191–205.
4. J. Gasienica-Jozkowy, M. Knapik and B. Cyganek, An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance, *Integr. Comput.-Aided Eng.* **28**(3) (2021) 221–235.
5. M. J. Gómez-Silva, A. de la Escalera and J. M. Armingol, Back-propagation of the mahalanobis istance through a deep triplet learning model for person re-identification, *Integr. Comput.-Aided Eng.* **28**(3) (2021) 277–294.
6. M. H. Rafiei and H. Adeli, A novel machine learning model for estimation of sale prices of real estate units, *J. Constr. Eng. Manag.* **142**(2) (2016) 04015066.
7. M. H. Rafiei, W. H. Khushefati, R. Demirboga and H. Adeli, Supervised deep restricted Boltzmann machine for estimation of concrete, *ACI Mater. J.* **114**(2) (2017) 237.
8. G. B. Martins, J. P. Papa and H. Adeli, Deep learning techniques for recommender systems based on collaborative filtering, *Expert Syst.* **37**(6) (2020) e12647.
9. G. Mirzaei, A. Adeli and H. Adeli, Imaging and machine learning techniques for diagnosis of Alzheimer's disease, *Rev. Neurosci.* **27**(8) (2016) 857–870.
10. H. S. Nogay and H. Adeli, Machine learning (ML) for the diagnosis of autism spectrum disorder (ASD) using brain imaging, *Rev. Neurosci.* **31**(8) (2020) 825–841.
11. G. Mirzaei and H. Adeli, Resting state functional magnetic resonance imaging processing techniques in stroke studies, *Rev. Neurosci.* **27**(8) (2016) 871–885.
12. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial networks, *Commun. ACM* **63**(11) (2020) 139–144.
13. M. Patel, M. Purohit, J. Shah and H. A. Patil, CinC-GAN for effective F_0 prediction for whisper-to-normal speech conversion, in *2020 28th Eur. Signal Processing Conf. (EUSIPCO)* (IEEE, 2021), pp. 411–415.
14. A. Shoeibi, N. Ghassemi, J. Heras, M. Rezaei and J. M. Gorriz, Automatic diagnosis of myocarditis in cardiac magnetic images using CycleGAN and deep pretrained models, in *Int. Work-Conf. the Interplay Between Natural and Artificial Computation* (Springer International Publishing, Cham, 2022), pp. 145–155.
15. Z. Wang, X. Ji, J.-B. Huang, S. Satoh, X. Zhou and Y. Zheng, Neural global shutter: Learn to restore video from a rolling shutter camera with global reset feature, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, New Orleans, LA, USA, 2022), pp. 17794–17803.
16. K. S. Lee, N.-T. Tran and N.-M. Cheung, InfoMax-GAN: Improved adversarial image generation via information maximization and contrastive learning, in *Proc. IEEE/CVF Winter Conf. Appl. Computer Vision* (IEEE, Waikoloa, HI, USA, 2021), pp. 3942–3952.
17. H. Liu, Z. Wan, W. Huang, Y. Song, X. Han and J. Liao, PD-GAN: Probabilistic diverse GAN for image inpainting, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, Virtual, 2021), pp. 9371–9381.
18. C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet and M. Norouzi, Image super-resolution via iterative refinement, *IEEE Trans. Pattern Anal. Mach. Intell.* **45** (2022) 4713–4726.
19. J. Zhu, C. Tan, J. Yang, G. Yang and P. Lio', Arbitrary scale super-resolution for medical images, *Int. J. Neural Syst.* **31**(10) (2021) 2150037.
20. F. A. Saiz, G. Alfaro, I. Barandiaran and M. Graña, Generative adversarial networks to improve the robustness of visual defect segmentation by semantic networks in manufacturing components, *Appl. Sci.* **11**(14) (2021) 6368.
21. C. Ieracitano, N. Mammone, A. Paviglianiti and F. C. Morabito, A conditional generative adversarial network and transfer learning-oriented anomaly classification system for electrospun nanofibers, *Int. J. Neural Syst.* **32**(12) (2022) 2250054.

22. H. Wu, F. He, Y. Duan and X. Yan, Perceptual metric-guided human image generation, *Integr. Comput.-Aided Eng.* **29**(2) (2022) 141–151.
23. J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in *Proc. IEEE Int. Conf. Computer Vision* (IEEE, Venice, Italy, 2017), pp. 2223–2232.
24. R. Zhang, T. Pfister and J. Li, Harmonic unpaired image-to-image translation, in *Int. Conf. Learning Representations* (New Orleans, Louisiana, United States, 2018).
25. H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang and D. Tao, Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping, in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition* (IEEE, Long Beach, CA, USA, 2019), pp. 2427–2436.
26. V. Schmidt, M. N. Sreedhar, M. ElAraby and I. Rish, Towards lifelong self-supervision for unpaired image-to-image translation, arXiv:2004.00161.
27. Y. Xu, S. Xie, W. Wu, K. Zhang, M. Gong and K. Batmanghelich, Maximum spatial perturbation consistency for unpaired image-to-image translation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, New Orleans, LA, USA, 2022), pp. 18311–18320.
28. J. Guo, J. Li, H. Fu, M. Gong, K. Zhang and D. Tao, Alleviating semantics distortion in unsupervised low-level image-to-image translation via structure consistency constraint, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, New Orleans, LA, USA, 2022), pp. 18249–18259.
29. Q. Mao, H.-Y. Tseng, H.-Y. Lee, J.-B. Huang, S. Ma and M.-H. Yang, Continuous and diverse image-to-image translation via signed attribute vectors, *Int. J. Comput. Vis.* **130**(2) (2022) 517–549.
30. D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, in *ICLR* (San Diego, United States, 2015).
31. H. Zhang, I. Goodfellow, D. Metaxas and A. Odena, Self-attention generative adversarial networks, in *Inter. Conf. Mach. Learn.* (International Machine Learning Society (IMLS), Long Beach, United States, 2019), pp. 7354–7363.
32. R. Ali and Y.-J. Cha, Attention-based generative adversarial network with internal damage segmentation using thermography, *Autom. Constr.* **141** (2022) 104412.
33. S. Jia, Z. Wang, Q. Li, X. Jia and M. Xu, Multi-attention generative adversarial network for remote sensing image super resolution, *IEEE Trans. Geosci. Remote Sens.* **60** (2022) 5624715.
34. E. Becker, P. Pandit, S. Rangan and A. Fletcher, Instability and local minima in GAN training with kernel discriminators, in *Advances in Neural Information Processing Systems (NeurIPS)*, eds. A. H. Oh, A. Agarwal, D. Belgrave and K. Cho (New Orleans, Louisiana, United States of America, 2022).
35. C. Wang, C. Xu, X. Yao and D. Tao, Evolutionary generative adversarial networks, *IEEE Trans. Evol. Comput.* **23**(6) (2019) 921–934.
36. H. Shu, Y. Wang, X. Jia, K. Han, H. Chen, C. Xu, Q. Tian and C. Xu, Co-evolutionary compression for unpaired image translation, in *Proc. IEEE/CVF Int. Conf. Computer Vision* (IEEE, Seoul, Korea (South), 2019), pp. 3235–3244.
37. K. Zhou, S.-K. Oh, W. Pedrycz and J. Qiu, Data preprocessing strategy in constructing convolutional neural network classifier based on constrained particle swarm optimization with fuzzy penalty function, *Eng. Appl. Artif. Intell.* **117** (2023) 105580.
38. L. Mariot, S. Picek, D. Jakobovic and A. Leporati, Evolutionary algorithms for designing reversible cellular automata, *Genet. Program. Evolvable Mach.* **22**(4) (2021) 429–461.
39. Y. Xue, P. Jiang, F. Neri and J. Liang, A multi-objective evolutionary approach based on graph-in-graph for neural architecture search of convolutional neural networks, *Int. J. Neural Syst.* **31**(9) (2021) 2150035.
40. T. Wu, Q. Lyu and L. Pan, Evolution-communication spiking neural p systems, *Int. J. Neural Syst.* **31**(02) (2021) 2050064.
41. H. Adeli and S.-L. Hung, *Machine Learning: Neural Networks, Genetic Algorithms, and Fuzzy Systems* (John Wiley & Sons, 1994).
42. H. Kim and H. Adeli, Discrete cost optimization of composite floors using a floating-point genetic algorithm, *Eng. Optim.* **33**(4) (2001) 485–501.
43. M. Kociecki and H. Adeli, Shape optimization of free-form steel space-frame roof structures with complex geometries using evolutionary computing, *Eng. Appl. Artif. Intell.* **38** (2015) 168–182.
44. M. Kociecki and H. Adeli, Two-phase genetic algorithm for topology optimization of free-form steel space-frame roof structures with complex curvatures, *Eng. Appl. Artif. Intell.* **32** (2014) 218–227.
45. H. Adeli and S. Kumar, Concurrent structural optimization on massively parallel supercomputer, *J. Struct. Eng.* **121**(11) (1995) 1588–1597.
46. B. K. Oh, K. J. Kim, Y. Kim, H. S. Park and H. Adeli, Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings, *Appl. Soft Comput.* **58** (2017) 576–585.
47. D. Saranovic, M. Pavlovski, W. Power, I. Stojkovic and Z. Obradovic, Interception of automated adversarial drone swarms in partially observed environments, *Integr. Comput.-Aided Eng.* **28**(4) (2021) 335–348.
48. B. Zhang, Z. Xu, J. Zhang and G. Wu, A warning framework for avoiding vessel-bridge and vessel-vessel collisions based on generative adversarial and dual-

task networks, *Comput.-Aided Civ. Infrastruct. Eng.* **37**(5) (2022) 629–649.

49. W. Qian, Y. Xu and H. Li, A self-sparse generative adversarial network for autonomous early-stage design of architectural sketches, *Comput.-Aided Civ. Infrastruct. Eng.* **37**(5) (2022) 612–628.
50. Y. Gao, P. Zhai and K. M. Mosalam, Balanced semisupervised generative adversarial network for damage assessment from low-data imbalanced-class regime, *Comput.-Aided Civ. Infrastruct. Eng.* **36**(9) (2021) 1094–1113.
51. Y. Yu, C. Cai and Y. Liu, Probabilistic vehicle weight estimation using physics-constrained generative adversarial network, *Comput.-Aided Civ. Infrastruct. Eng.* **36**(6) (2021) 781–799.
52. K. Zhang, Z. He, L. Zheng, L. Zhao and L. Wu, A generative adversarial network for travel times imputation using trajectory data, *Comput.-Aided Civ. Infrastruct. Eng.* **36**(2) (2021) 197–212.
53. H. Maeda, T. Kashiyama, Y. Sekimoto, T. Seto and H. Omata, Generative adversarial network for road damage detection, *Comput.-Aided Civ. Infrastruct. Eng.* **36**(1) (2021) 47–60.
54. S. Chen, W. Wang, B. Xia, X. You, Q. Peng, Z. Cao and W. Ding, CDE-GAN: Cooperative dual evolution-based generative adversarial network, *IEEE Trans. Evol. Comput.* **25**(5) (2021) 986–1000.
55. V. Costa, N. Lourenço, J. Correia and P. Machado, COEGAN: Evaluating the coevolution effect in generative adversarial networks, in *Proc. Genetic and Evolutionary Computation Conf.* (Association for Computing Machinery, New York, NY, United States, 2019), pp. 374–382.
56. K. Mehta, Z. Kobti, K. Pfaff and S. Fox, Data augmentation using CA evolved GANs, in *2019 IEEE Symp. Computers and Communications (ISCC)* (IEEE, Barcelona, Spain, 2019), pp. 1087–1092.
57. C. He, S. Huang, R. Cheng, K. C. Tan and Y. Jin, Evolutionary multiobjective optimization driven by generative adversarial networks (GANs), *IEEE Trans. Cybern.* **51**(6) (2020) 3129–3142.
58. G. Ying, X. He, B. Gao, B. Han and X. Chu, EAGAN: Efficient two-stage evolutionary architecture search for GANs, in *Eur. Conf. Computer Vision* (Springer-Verlag, Berlin, Heidelberg, 2022), pp. 37–53.
59. Q. Lin, Z. Fang, Y. Chen, K. C. Tan and Y. Li, Evolutionary architectural search for generative adversarial networks, *IEEE Trans. Emerg. Top. Comput. Intell.* **6** (2022) 783–794.
60. S. Hicsonmez, N. Samet, E. Akbas and P. Duygulu, GANILLA: Generative adversarial networks for image to illustration translation, *Image Vis. Comput.* **95** (2020) 103886.
61. R. Chen, W. Huang, B. Huang, F. Sun and B. Fang, Reusing discriminators for encoding: Towards unsupervised image-to-image translation, in *Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition* (IEEE, Seattle, WA, USA, 2020), pp. 8168–8177.
62. Y. Zhao, R. Wu and H. Dong, Unpaired image-to-image translation using adversarial consistency loss, in *European Conf. Computer Vision* (Springer-Verlag, Berlin, Heidelberg, 2020), pp. 800–815.
63. T. Park, A. A. Efros, R. Zhang and J.-Y. Zhu, Contrastive learning for unpaired image-to-image translation, in *European Conf. Computer Vision* (Springer-Verlag, Berlin, Heidelberg, 2020), pp. 319–345.
64. Y. Mao, J. Jin, R. Xu, S. Li, Y. Miao and A. Cichocki, The influence of visual attention on the performance of a novel tactile P300 brain–computer interface with cheeks-stim paradigm, *Int. J. Neural Syst.* **31**(4) (2021) 2150004.
65. Y. Zhao, G. Zhang, C. Dong, Q. Yuan, F. Xu and Y. Zheng, Graph attention network with focal loss for seizure detection on electroencephalography signals, *Int. J. Neural Syst.* **31**(7) (2021) 2150027.
66. M. C. Maya-Piedrahita, P. M. Herrera-Gomez, L. Berrío-Mesa, D. A. Cárdenas-Peña and A. Orozco-Gutierrez, Supported diagnosis of attention deficit and hyperactivity disorder from EEG based on interpretable kernels for hidden Markov models, *Int. J. Neural Syst.* **32**(3) (2022) 2250008.
67. B. Yang, Y. Kang, Y. Yuan, X. Huang and H. Li, ST-LBAGAN: Spatio-temporal learnable bidirectional attention generative adversarial networks for missing traffic data imputation, *Knowl.-Based Syst.* **215** (2021) 106705.
68. S. Woo, J. Park, J.-Y. Lee and I. S. Kweon, CBAM: Convolutional block attention module, in *Proc. European Conf. Computer Vision (ECCV)* (Springer-Verlag, Berlin, Heidelberg, 2018), pp. 3–19.
69. H. Emami, M. M. Aliabadi, M. Dong and R. B. Chinnam, SPA-GAN: Spatial attention GAN for image-to-image translation, *CoRR* (2019) abs/1908.06616.
70. X. Xie, Y. Dong, Y. Li and S. Wang, AT-GAN: Attention transfer GAN for image-to-image translation, in *Proc. 2020 4th Int. Conf. Digital Signal Processing* (Association for Computing Machinery, 2020), pp. 102–106.
71. M. Ćurković, A. Ćurković and D. Vučina, Image binarization method for markers tracking in extreme light conditions, *Integr. Comput.-Aided Eng.* **29**(2) (2022) 175–188.
72. A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd edn. (Springer, 2015).
73. D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *ICLR (Poster)* (San Diego, United States, 2015), p. 13.
74. M. H. Rafiei and H. Adeli, NEEWs: A novel earthquake early warning model using neural dynamic classification and neural dynamic optimization, *Soil Dyn. Earthq. Eng.* **100** (2017) 417–427.

75. M. Rafiei and H. Adeli, Novel machine learning model for construction cost estimation taking into account economic variables and indices, *J. Constr. Eng. Manag.* **144**(12) (2018) 04018106.
76. A. Hassanpour, M. Moradikia, H. Adeli, S. R. Khayami and P. Shamsinejadbabaki, A novel end-to-end deep learning scheme for classifying multi-class motor imagery electroencephalography signals, *Expert Syst.* **36**(6) (2019) e12494.
77. H. S. Nogay and H. Adeli, Detection of epileptic seizure using pretrained deep convolutional neural network and transfer learning, *Eur. Neurol.* **83**(6) (2020) 602–614.
78. H. S. Nogay and H. Adeli, Diagnostic of autism spectrum disorder based on structural brain MRI images using, grid search optimization, and convolutional neural networks, *Biomed. Signal Process. Control* **79** (2023) 104234.
79. D. S. Jodas, T. Yojo, S. Brazolin, G. D. N. Velasco and J. P. Papa, Detection of trees on street-view images using a convolutional neural network, *Int. J. Neural Syst.* **32**(1) (2022) 2150042.
80. H. Sun, J. Jin, R. Xu and A. Cichocki, Feature selection combining filter and wrapper methods for motor-imagery based brain–computer interfaces, *Int. J. Neural Syst.* **31**(9) (2021) 2150040.
81. G. Mirzaei and H. Adeli, Segmentation and clustering in brain MRI imaging, *Rev. Neurosci.* **30**(1) (2018) 31–44.
82. Y. Xue and J. Qin, Partial connection based on channel attention for differentiable neural architecture search, *IEEE Trans. Industr. Inform.* (2022) early access. DOI: 10.1109/TII.2022.3184700.
83. M.-H. Guo, T.-X. Xu, J.-J. Liu, Z.-N. Liu, P.-T. Jiang, T.-J. Mu, S.-H. Zhang, R. R. Martin, M.-M. Cheng and S.-M. Hu, Attention mechanisms in computer vision: A survey, *Comput. Vis. Media* **8** (2022) 1–38.
84. T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, Improved techniques for training GANs, in *Advances in Neural Information Processing Systems*, Vol. 29 (Curran Associates Inc., Red Hook, NY, United States, 2016), pp. 2234–2242.
85. M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler and S. Hochreiter, GANs trained by a two time-scale update rule converge to a local nash equilibrium, in *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates Inc., Red Hook, NY, United States, 2017), pp. 6629–6640.
86. M. Bińkowski, D. J. Sutherland, M. Arbel and A. Gretton, Demystifying MMD GANs, arXiv:1801.01401.
87. D. Torbunov, Y. Huang, H. Yu, J. Huang, S. Yoo, M. Lin, B. Viren and Y. Ren, UVCGAN: Unet vision transformer cycle-consistent GAN for unpaired image-to-image translation, arXiv:2203.02557.
88. Y. Alami Mejjati, C. Richardt, J. Tompkin, D. Cosker and K. I. Kim, Unsupervised attention-guided image-to-image translation, in *Adv. in Neural Information Processing Systems*, Vol. 31 (Curran Associates Inc., Red Hook, NY, United States, 2018), pp. 3697–3707.
89. J. Kim, M. Kim, H. Kang and K. Lee, U-GAT-IT: Unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation, arXiv:1907.10830.
90. M.-Y. Liu, T. Breuel and J. Kautz, Unsupervised image-to-image translation networks, in *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates Inc., Red Hook, NY, United States, 2017), pp. 700–708.
91. T. Kim, M. Cha, H. Kim, J. K. Lee and J. Kim, Learning to discover cross-domain relations with generative adversarial networks, in *Int. Conf. Machine Learning* (2017), pp. 1857–1865.