Culturally Evolved GANs for Generating Fake Stroke Faces

4 authors:

Kaitav Mehta
University of Windsor
3 PUBLICATIONS 0 CITATIONS

Ziad Kobti
University of Windsor
104 PUBLICATIONS 617 CITATIONS

Kathryn Anne Pfaff
University of Windsor
35 PUBLICATIONS 212 CITATIONS

Susan M Fox-Wasylyshyn
University of Windsor
30 PUBLICATIONS 473 CITATIONS

Culturally Evolved GANs for generating Fake Stroke Faces

Kaitav Mehta
School of Computer Science
University of Windsor
Windsor, Canada
mehta128@uwindsor.ca

Ziad Kobti
School of Computer Science
University of Windsor
Windsor, Canada
kobti@uwindsor.ca

Kathryn Pfaff
Faculty of Nursing
University of Windsor
Windsor, Canada
kpfaff@uwindsor.ca

Susan Fox
Faculty of Nursing
University of Windsor
Windsor, Canada
sfox@uwindsor.ca

***Abstract*—Mining medical data images have great potential for exploring hidden patterns in the medical domain. Medical data are heterogeneous which involves images to a great extent like MRI, ECG or Stroke effects etc. Knowledge discovery from such data can improve the diagnostic technique. However, to make the machine learn from such datasets requires large data. In the low-data regime, machine learning algorithms work poorly. Data Augmentation alleviates this by using existing data more effectively, but standard data augmentation produces only limited alternative data. Recent developments in Deep Learning field is noteworthy when it comes to learning probability distribution of points through neural networks, and one of key part for such progress is because of Generative Adversarial Networks(GANs). In this paper, we propose an evolutionary training technique using a cultural algorithm(CA) for neuro-evolution of deep task oriented GANs to find the best architecture for the given dataset. This architecture will help in generating similar but completely new data images which can be further used for training diagnostic Neural Networks. We have compared our approach with the Genetic Algorithm(GA) based neuro-evolution of GANs and show that CA based neuro-evolution of GANs evolves architecture which can generate a higher number of stroke-face images with better resolution when there is low data of original stroke faces.**

***Index Terms*—Data Mining, Generative Adversarial Networks, Neuro-evolution, Cultural Algorithm, Machine Learning**

I. Introduction

Stroke, also known as cerebrovascular accident, is the second leading cause of death in the world [1]. A stroke occurs when the brain cells fails to receive necessary oxygen because of a blockage or rupture in a blood vessel that feeds the brain [2]. Early recognition and timely diagnosis are needed to reduce stroke-related disability and death. Facial drooping is a common sign of stroke [2], but it can also be associated with other health issues, and therefore overlooked by lay people and health professionals [2]. Currently, the gold standard for diagnosis of a stroke is computed tomography (CT) scanning [2] [3]. Unfortunately, access to CT scans is poor in low-income countries, and in many rural/remote areas [3] [4] [5], resulting in poorer survival rates [3] [6]. Novel approaches that more quickly and accurately recognize stroke are needed, particularly in counties and other settings where access to CT scans and specialized health care services is limited.

Generative Adversarial Networks (GANs) [7] are a machine learning approach capable of generating novel example outputs across a space of provided training examples. The first GAN was created by Ian Goodfellow. A GAN consists of two neural networks playing a zero-sum game with each other. The discriminator network tries to determine whether the information is real or fake. The other neural network, called a generator, tries to create data that the discriminator thinks is real. The generator never sees the genuine data, it must learn to create real samples by receiving feedback from the discriminator, this is called adversarial loss. The longer these two neural networks play this game, the more they sharpen each other's skills.

However, training these Neural Networks requires large datasets. In many realistic settings in a medical domain we need to achieve goals with limited datasets. In such cases DNN seems to fall short, overfitting on the training set and producing poor generalization on the test set [8]. To overcome this issue it is possible to generate more data from existing data by applying data augmentation techniques to the original datasets. But standard data augmentation produces only limited alternatives [8]. Here we propose a neuro-evolutionary GANs based techniques which alleviate the size of the original dataset.

The necessity to define hyperparameters of Deep Neural Network(DNN) is an issue which is not only related to GANs but also neural network in general. The topology and hyperparameters are chosen empirically, thus spending human time on a repetitive task such as fine-tuning. Ideally, one would want to have an automated method to generate the right architecture for any given task. One approach to generate these architectures is through the use of evolutionary algorithms [9]. Recently researchers have started evolving GANs architecture using different black-box optimization techniques like Genetic Algorithm and Coevolutionary Algorithm. Neural networks are created to mimic the human brain capabilities, we humans became the top predator when we started evolving culturally [10] and hence our motivation for using CA.

Our contribution in this paper is of two-fold first we use knowledge-based evolutionary technique i.e Cultural Algorithm to generate GANs architecture and secondly, we aim to generate targeted images in the low-data regime. According to our research, no other paper has used CA to evolve GAN architectures. We will test our proposed approach by generating stroke faces, an area where there is limited original data. This approach can be extended to generate image sets such as CT Scan Image datasets [11]. These fake generated images can be used to improve the accuracy of diagnostic Neural classifiers [8], such as those that are currently used for the recognition and diagnosis of stroke.

II. Background and Related Works

Evolving deep Neural Network using nature-inspired algorithm has been steadily increasing in recent years. We review presentational contributions which are related to our work.

A. Neuro-evolution of ANNs

Over the last few years, evolutionary algorithms have achieved considerable success in solving deep learning problems and to minimize human participation in designing deep Networks and automatically optimize hyper-parameters and generate deep Neural network architecture using evolutionary search. [12] [13] [14] [15].

Researchers have focused on using Genetic Algorithm for evolving deep neural networks to solve different problems [16]. Many other nature-inspired evolutionary algorithms have also been used like coevolutionary [17], cultural algorithm [18] and other, search optimization algorithms to evolve Neural Network Architecture in different dimensions to carry out various task. Evolutionary approaches were also able to find better architecture than state of the art human crafted architecture [19].

B. Evolutionary GANs

In E-GAN [20] the authors have proposed an evolutionary training of GANs in which three different loss functions of Generators keep evolving the weights of the networks through generations during the training, and they consider that Discriminator is not evolved and it was assumed to be optimal classifier during the training phase. In Towards Distributed Coevolutionary GANs [21] authors have proposed how to overcome common GANs issues like mode collapse, training instability using coevolutionary spatial approach. In Deep Interactive Evolution [22], authors have combined GANs and interactive evolutionary computation where they have kept latent vector under evolutionary control, allowing controllable and high-quality image generation. Recently Unai et al. [23] tried to evolve whole GAN architecture loss-functions and Generator-Discriminator synchronization parameters using Genetic Algorithm.

C. Data Augmentation using GANs

The impact of data mining using a machine learning technique for medical diagnostic is noteworthy [24]. Many researchers have used GAN for bringing diversity to medical data. Hoo-chang et al. generated MRI images with brain tumor using GANs [11]. They also compared the tumor segmentation on generated data with original data and demonstrated value of generative models as anonymization tool. Another research by Maayan et al. improved the classification of liver lesion by augmenting data with GANs [25]. It is worthy to read a broad review of GANs application for medical image analysis [26]. GANs does not produce completely new data but produces new data with different properties, which can capture many different aspects of original data and all of those different aspects can be captured by a classifier to improve diagnostic accuracy.

III. Neuro-evolved GANs using CA

A. Evolvability of GAN components

The GAN architecture is made up of different components e.g(Neural networks, loss functions), we define the GAN components that can be modified to influence its behaviour.

- Architecture of the generator and discriminator. Which include number of hidden layers, Activation functions, Initialization of weights.
- Loss function used to evaluate the model during training
- Training epochs of the generator and discriminator in one generation(D-G loops)
- Gradient-based optimization technique used to search for the model (e.g decaying learning rate)
- Content of the training batch

The maximum number of hidden layers, the maximum size of any layer are parameters of the algorithm. The number of loops of G and D for the training are modifiable parameters. GANs are allowed to have different weight initialization and activation functions. Table 1 shows all possible assignments for the modifiable components of the algorithm which we have used for our experiments. However one can add custom components like convolution kernel size, stride size, other different loss functions etc.

B. CA for evolving GANs

The search process used by standard Evolutionary Algorithms(EA) is unbiased, using little or no domain knowledge to guide the search process. The performance of EAs can be improved if domain knowledge is used to bias the search process. Domain knowledge is useful to reduce the search space by promoting the desirable parts of the solutions space and pruning undesirable parts.

Culture is the total of learned behaviour of a group of people that is considered to be the tradition of that people and is transmitted from generations to generations [27], in our case we are transferring specification among GANs.

In CA we maintain two search spaces: the population space, and a belief space (to represent cultural component). The belief space models cultural information about the population, while population space represents individuals. Both the population space and belief space evolve in parallel, with both influencing one another.

1) Description of individual GAN Component: In population space, there is a list of individual GAN which represents genetic components. The CA used to evolve the GANs uses a list-based encoding and genetic operators that operate on these lists. Each GAN is encoded by a list of two parameters, one for each network. Each network parameter list includes

TABLE I

FEASIBLE ASSIGNMENTS FOR GAN COMPONENTS MODIFIED BY CA AND MUTATION OPERATIONS

Weights-Init	Activation Fn	Loss Fn	Mutation Op
Normal Xavier	Relu LeakyRelu Sigmoid Tanh	Binary Cross Entropy Mean Absolute Error L1Loss	No. of layer change Activation change Weight change Loss function change D-G loops

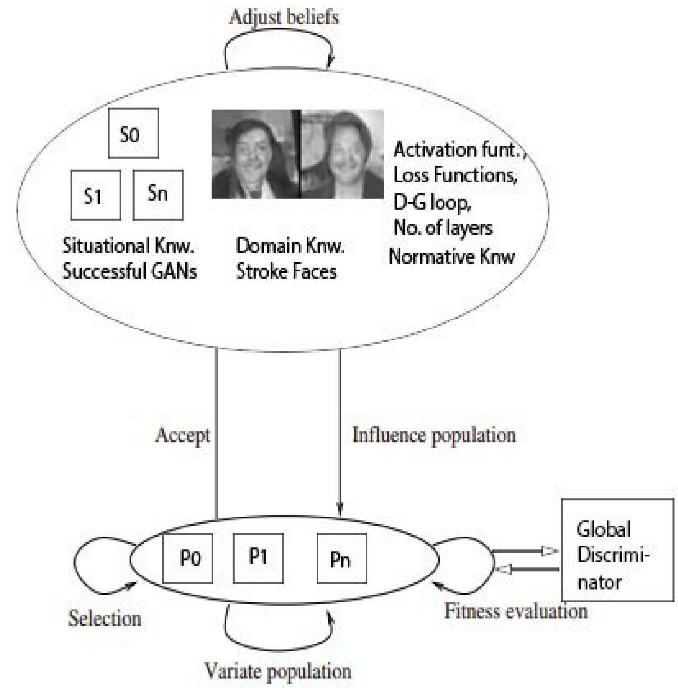

Fig. 1. Population and Belief Space of CA GAN.

the number of hidden layers, output feature in each layer, activation function for each layer, D/G loop. Also with the two list parameter, there is a loss function associated with each individual. In generator genotype, all hidden layers are composed of transpose convolutional section followed by batch normalization and activation function. While in discriminator genotype, all hidden layers are strided convolutional, followed by batch normalization and activation function, these hidden layers design is inspired by DCGAN [28]. The specification of the hidden layers (e.g weights and bias) will be trained by gradient descent method and will not be part of the evolution. During the evaluation step, the GAN is built according to the specification encoded by the solution, it is trained, and its quality is assessed by a predefined fitness function.

2) CA Belief Space and Knowledge Component: In Belief space, we will use three knowledge components i.e Situational knowledge, Normative knowledge and Domain knowledge.

A situational knowledge component keeps track of the best solutions found at each generation. We keep the fittest population as our situtational knowledge and influence the future generation using the individuals from the situational knowledge by crossover.

The second knowledge component we use is Normative knowledge, which provides guidelines for mutational adjustments to the individuals. Table 1 act as our Normative knowledge where the individuals mutate according to it using mutational operations. Also, we provide a range of maximum no of layers in neural networks, the maximum number of D-G loops during the start of the program which is also considered as Normative knowledge.

The last belief space is domain knowledge, where we store the information about the domain problem where CA is applied. In our domain knowledge, we have original stroke face dataset and we keep adding GANs generated stroke faces during the evolutionary process. We keep stroke faces as our domain knowledge because the more stroke faces we have for training the better and higher fake stroke faces GANs will generate. We influence population space by domain knowledge during the training process of GANs by wavering training batch of GANs. After each generation the domain knowledge is increased, an external classifier selects stroke faces from the generated images and adds them to domain space.

3) Crossover and Mutation: Given two parents P_1 = (G_1,D_1) and P_2 = (G_2,D_2), the crossover operator creates two offspring O_1 = (G_1,D_2) and O_2 = (G_2,D_1). Crossover preservers integrity of each network. The mutation step consists of the application of one operator among the set of possible mutations stated in Table 1, to only one of the network. The mutations have a different scope of applications, activation change modifies the activation function in layers of neural network, D-G loop change modifies the number of training epochs in one of the networks, while loss change alters the GAN loss function, number of hidden layers and output features of hidden layer.

The pseudo-code of the CA is shown in Algorithm(1)

C. Evaluation

The evaluation is the operation of measuring the quality of individuals. To determine evolutionary direction i.e individual selection, we devise an evaluation function to measure the performance of the evolved individuals. In this evaluation, we just focus on the generators quality. We feed generator produced images into the Global Discriminator D and observe the average value of the output, which is named as quality fitness score(F_q) [20].

$$F_q = \mathbb{E}_z[D(G(z))] \tag{1}$$

Algorithm 1 Cultural Algorithm(CA) for evolving GANs

Set g $\Leftarrow$ 0. Create population P_0 by generating N random GAN description.
Initialize B_0 (Belief Space)
while Termination criteria is met **do**
  Train by influencing from domain knowledge $B(_g)$ and Evaluate P_g using fitness function
  From P_g select a population P_g^S of K $\leq$ N solutions according to a selection method
  Adjust belief space $B(_g)$ by accepting better solutions
  Create a mating pool from P_g^S and $B(_g)$
  Create offspring set O_g by applying crossover with probability P_x
  Apply mutation with probability $P_m = 1 - P_x$. Choice of mutation operator is made uniformly at random.
  Create P_{g+1} by selecting best solutions from $\{P_g, O_g\}$
  g $\Leftarrow$ g + 1
end while
return Top Individual

Note that the global Discriminator D is constantly upgraded to be optimal during the training process, reflecting the quality of generators at each evolutionary step. For simplicity of experiments, we are not considering the diversity of Generators samples. Overall, a relatively low fitness score leads to higher training efficiency and better generative performance.

IV. Experiments

A. Experimental Setup

Table II describes the parameters used in the CAGAN experiments reported in this paper. We have designed a set of Python classes that implement GAN architecture when a list of description is provided. The network architecture is defined using pytorch library [29]. We modify the original architecture of DCGAN [28] using the components as stated in Table 1. The loss functions and activation functions are used from the library of pytorch. The CA was implemented by extending the GA based DEAP library [30]. The input data for the GAN will be a batch of gray-scale images, after training the Generators will generate images which have stroke face images. We will use our fitness function to evaluate the generator performance by the global discriminator and select the individuals according to their fitness for further generations. For both the algorithms we have tested with a population size of 5. If the architecture of the GAN collapses i.e if it keeps giving constant loss value then we remove those individuals and generate new ones. We have performed all our experiments on Nvidia GeForce GTX 745, to run for 50 generations for 64 x 64 images it cost around 8 hours on a single GPU.

TABLE II
EXPIRIMENTAL PARAMETERS

Evolutionary parameters	Value
Number of generations	50
Population Size	5
Crossover probability	0.1
Mutation probability(1-Cx)	0.9
Genome limit	5
D-G Loop	3
Selection	3 Best
Celeb Faces	13000
Augmented Stroke Faces	1000
Batch Size	64
Batches per Generation	108
Optimizer	Adam
Initial Learning Rate	1e-2

Fig. 2. Sample training dataset which also includes stroke faces

B. Datasets and Data augmentation

To perform our experiments we have taken 13k image dataset of celebrity human face from Kaggle and converted it into 64x64 size and also we have gray-scaled these images [31], so it takes less computation time as our goal was not to show the high-quality image generation rather it was to prove CA works better than GA when there is less data for a particular task oriented image generation. Additionally, we have taken 150 open source stroke faces which were available to use for experimentation and changed their dimension to 64 x 64 with gray-scale. We then augment those stroke face images by adding rotation, translation and noise and increased its size to 1000 images. Figure 2 shows the sample batch for training purpose of GANs which has normal faces as well as stroke faces.

C. GA vs CA experimentation and comparison

We are inspired by the GA algorithm model as stated in [23], and we implemented our GA model consisting of Multi-level perceptrons(MLP) using the pytorch library and used the evolving component for GANs as stated in Table 1. We implemented the CA algorithm as stated in Algorithm 1, and used the same evolving components as we have used for GA. The dataset as described above is used for both the algorithms, just the difference is in CA we waver the batch naturally as it is part of our domain knowledge. In both the experiments we start with the random generation of the architecture of GANs, the neural weights of the Generators and Discriminators are deleted after each generation, but the neural weights of a global discriminator are not deleted. We have run both the algorithms

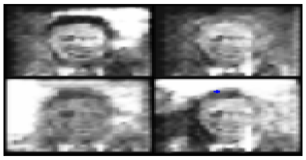

Fig. 3. Results of best GA generated architecture for image generation after 50 gen and running it for 5 epochs.

Fig. 4. Results of best CA generated architecture for image generation after 50 gen and running it for 5 epochs.

for 50 generations and in each generation for 5 epochs, the loss of global discriminator was fixed in a small range after 50 generations for CA. We select the best architecture from GA and CA using fitness results of evolved GANs, and compared the results after 50 generations as shown in figure 3 and 4. Further to test it more precisely we select the best individual from both the algorithms and run it till both the individual converges but the training is kept as stated in the respective algorithms, and we non-automatically count how much stroke faces are seen in the generated batch and compare those results as shown in figure 5 and 6.

To compare the image quality and diversity of both the techniques best architectures we considered using GAN-train and GAN-test evaluation method as described in [32]. GAN-train learns a classifier on GAN generated images and measure the performance on real test images. This evaluates the diversity and realism of GAN images. GAN-test learns a classifier on real images and evaluates it on GAN images. This measure shows how realistic GAN images are. Table 2 shows the accuracy comparison of the best GA-GAN and CA-GAN, after running both algorithms for 5 times and averaging the GAN-train and GAN-test score. Higher the accuracy, comparatively better the GAN model. Also from the results graphs in fig 5 we can see that almost 15% more stroke faces are generated from CA evolved GAN when compared to GA. This significant result gap happens because belief knowledge is incorporated into the search process which reduces the search space by removing undesired part of solution space moreover the quality of generated images with CA is better as CA used

TABLE III
AVERAGE GAN-TRAIN AND GAN-TEST ACCURACY PERCENTAGE OF THE BEST GA-GAN AND CA-GAN(HIGHER IS BETTER)

Model	GAN-train	GAN-test
real images	91%	-
Best CA-GAN	59%	53%
Best GA-GAN	46%	41%

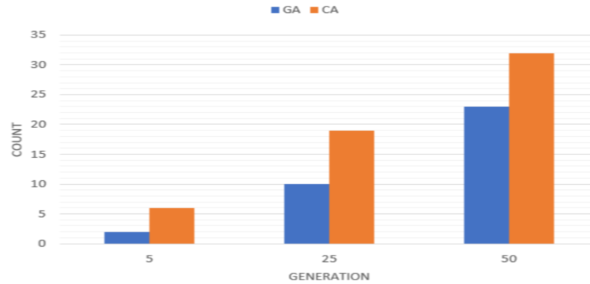

Fig. 5. Average stroke face images count comparison from the last batch of best evolved architecture after running it for 50 epochs

Fig. 6. Generated images from best CA evolved architecture after 50 generation and running it for 50 epochs

Convolutional layers while GA uses MLP. We demonstrate that CA can produce better results in image generation from GAN when there is less dataset for a particular task(i.e generating stroke faces) when compared to GA.

V. CONCLUSION

In this paper, we have proposed a method for improving the neuro-evolutionary ability of GANs using CA approach. While we have focused on the non-standard dataset of the face, a straight forward question is to determine whether CA evolved GANs can generate unique images when there is a smaller dataset.

Moreover, we presented the importance of using a classical evolutionary algorithm in Deep learning rather than just GA and achieved significant results by using the right evolutionary technique for the relevant problem. We also showed that it is effective to use CA when there is a smaller dataset for a particular task(stroke face images) when compared to GA. Our method does not outperform existing state of the art hard crafted GANs, but it can generate comparable results.

A. Future Work

In the future work, we can extend this approach to test the transferability of neuro-evolved GANs, from gray-scale to RGB and test for higher resolution of images. We also want to improve the complexity of neuro-evolution of Generator and Discriminator and compare it with human-designed GANs.

Although our work is exploratory, future work could involve testing the approach with medical CT scan datasets to evaluate the predictive capability of this approach on stroke recognition.

REFERENCES

[1] W. H. Organization, "The top 10 causes of death," 2016.

[2] "Canadian stroke best practices," 2018.

[3] A. L. Berkowitz, "Managing acute stroke in low-resource settings," *Bulletin of the World Health Organization*, vol. 94, no. 7, p. 554, 2016.

[4] M. O. Owolabi, O. Arulogun, S. Melikam, A. M. Adeoye, S. Akarolo-Anthony, R. Akinyemi, D. Arnett, H. Tiwari, M. Gebregziabher, C. Jenkins, *et al.*, "The burden of stroke in africa: a glance at the present and a glimpse into the future," *Cardiovascular Journal of Africa*, vol. 26, no. 2 H3Africa Suppl, p. S27, 2015.

[5] "Global atlas of medical devices," May 2017.

[6] V. L. Feigin, C. M. Lawes, D. A. Bennett, S. L. Barker-Collo, and V. Parag, "Worldwide stroke incidence and early case fatality reported in 56 population-based studies: a systematic review," *The Lancet Neurology*, vol. 8, no. 4, pp. 355–369, 2009.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[8] A. Antoniou, A. Storkey, and H. Edwards, "Augmenting image classifiers using data augmentation generative adversarial networks," in *International Conference on Artificial Neural Networks*, pp. 594–603, Springer, 2018.

[9] Estenben, "Using evolutionary automl to discover neural network architectures," Mar 2018.

[10] G. M. Feinman and L. R. Manzanilla, *Cultural evolution: Contemporary viewpoints*. Springer Science & Business Media, 2000.

[11] H.-C. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, and M. Michalski, "Medical image synthesis for data augmentation and anonymization using generative adversarial networks," in *International Workshop on Simulation and Synthesis in Medical Imaging*, pp. 1–11, Springer, 2018.

[12] R. Miikkulainen, J. Liang, E. Meyerson, A. Rawal, D. Fink, O. Francon, B. Raju, H. Shahrzad, A. Navruzyan, N. Duffy, *et al.*, "Evolving deep neural networks," in *Artificial Intelligence in the Age of Neural Networks and Brain Computing*, pp. 293–312, Elsevier, 2019.

[13] E. Real, S. Moore, A. Selle, S. Saxena, Y. L. Suematsu, J. Tan, Q. V. Le, and A. Kurakin, "Large-scale evolution of image classifiers," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2902–2911, JMLR. org, 2017.

[14] B. Shabash and K. C. Wiese, "Evonn: a customizable evolutionary neural network with heterogenous activation functions," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1449–1456, ACM, 2018.

[15] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evolutionary computation*, vol. 10, no. 2, pp. 99–127, 2002.

[16] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 497–504, ACM, 2017.

[17] V. Costa, N. Lourenço, and P. Machado, "Coevolution of generative adversarial networks," in *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*, pp. 473–487, Springer, 2019.

[18] P. H. Mcquesten, *Cultural enhancement of neuroevolution*. PhD thesis, 2002.

[19] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," *arXiv preprint arXiv:1802.01548*, 2018.

[20] C. Wang, C. Xu, X. Yao, and D. Tao, "Evolutionary generative adversarial networks," *IEEE Transactions on Evolutionary Computation*, 2019.

[21] T. Schmiedlechner, A. Al-Dujaili, E. Hemberg, and U.-M. O'Reilly, "Towards distributed coevolutionary gans," *arXiv preprint arXiv:1807.08194*, 2018.

[22] P. Bontrager, W. Lin, J. Togelius, and S. Risi, "Deep interactive evolution," in *International Conference on Computational Intelligence in Music, Sound, Art and Design*, pp. 267–282, Springer, 2018.

[23] U. Garciarena, R. Santana, and A. Mendiburu, "Evolved gans for generating pareto set approximations," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 434–441, ACM, 2018.

[24] S. K. Wasan, V. Bhatnagar, and H. Kaur, "The impact of data mining techniques on medical diagnostics," *Data Science Journal*, vol. 5, pp. 119–126, 2006.

[25] M. Frid-Adar, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, "Synthetic data augmentation using gan for improved liver lesion classification," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pp. 289–293, IEEE, 2018.

[26] S. Kazeminia, C. Baur, A. Kuijper, B. van Ginneken, N. Navab, S. Albarqouni, and A. Mukhopadhyay, "Gans for medical image analysis," *arXiv preprint arXiv:1809.06222*, 2018.

[27] A. P. Engelbrecht, *Computational Intelligence : an Introduction*. Wiley., 2007.

[28] N. Inkawhich, "Dcgan tutorial," 2018.

[29] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[30] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *Journal of Machine Learning Research*, vol. 13, pp. 2171–2175, jul 2012.

[31] V. P, "Face images," 2019.

[32] K. Shmelkov, C. Schmid, and K. Alahari, "How good is my gan?," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 213–229, 2018.