p-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2024. № 1
e-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2024. № 1

UDC 004.93

METHOD OF GENERATIVE-ADVERSARIAL NETWORKS SEARCHING ARCHITECTURES FOR BIOMEDICAL IMAGES SYNTHESIS

Berezsky O. M. – Dr. Sc., Professor, Professor of the Department of Automated Control Systems, Lviv Polytechnic National University, Lviv, Ukraine.

Liashchynskyi P. B. – Post-graduate student of the Department of Automated Control Systems, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. The article examines the problem of automatic design of architectures of generative-adversarial networks. Generative-adversarial networks are used for image synthesis. This is especially true for the synthesis of biomedical images – cytological and histological, which are used to make a diagnosis in oncology. The synthesized images are used to train convolutional neural networks. Convolutional neural networks are currently among the most accurate classifiers of biomedical images.

Objective. The aim of the work is to develop an automatic method for searching for architectures of generative-adversarial networks based on a genetic algorithm.

Method. The developed method consists of the stage of searching for the architecture of the generator with a fixed discriminator and the stage of searching for the architecture of the discriminator with the best generator.

At the first stage, a fixed discriminator architecture is defined and a generator is searched for. Accordingly, after the first step, the architecture of the best generator is obtained, i.e. the model with the lowest FID value.

At the second stage, the best generator architecture was used and a search for the discriminator architecture was carried out. At each cycle of the optimization algorithm, a population of discriminators is created. After the second step, the architecture of the generative-adversarial network is obtained.

Results. Cytological images of breast cancer on the Zenodo platform were used to conduct the experiments. As a result of the study, an automatic method for searching for architectures of generatively adversarial networks has been developed. On the basis of computer experiments, the architecture of a generative adversarial network for the synthesis of cytological images was obtained. The total time of the experiment was ~39.5 GPU hours. As a result, 16,000 images were synthesized (4000 for each class). To assess the quality of synthesized images, the FID metric was used.The results of the experiments showed that the developed architecture is the best. The network's FID value is 3.39. This result is the best compared to well-known generative adversarial networks.

Conclusions. The article develops a method for searching for architectures of generative-adversarial networks for the problems of synthesis of biomedical images. In addition, a software module for the synthesis of biomedical images has been developed, which can be used to train CNN.

KEYWORDS: generative adversarial network, biomedical images, cytological images, search for neural network architectures, genetic algorithms, FID metrics, computer systems for automatic diagnostics.

ABBREVIATIONS

ATTN is a shorthand for Self-Attention;

GAN is a generative adversarial network;

CNN is a convolutional neural network;

CAD is a computer-aided diagnosis;

NAS is a neural architecture search;

AutoGAN is neural architecture search for GAN;

CIFAR-10 dataset – Canadian Institute for Advanced Research, 10 classes is a subset of the tiny images dataset and consists of 60000 32x32 color images;

AWS is a Amazon Web Services;

ELU is exponential linear unit activation function;

Zenodo is a general-purpose open repository developed under the European OpenAIRE program and operated by CERN;

ReLU is a rectified linear unit;

GA is a genetic algorithm;

Aging Evolution GA (AGA) is an evolutionary algorithm for neural architecture search.

Batch Norm is batch normalization;

ELU (exponential linear unit) is an activation function for neural networks;

Self-Attention operates by transforming the input sequence into three vectors: query, key, and value;

Hinge loss function is measure the distance of data points from the decision boundary;

Nvidia A6000 GPU is the videocard of the company Nvidia;

IS metric is a metric (Inception Score) based on the Google Inception V3 image classification model;

FID is Fréchet inception distance;

PyTorch is an open source machine learning framework Python Torch;

AWS S3 is a Amazon Web Services Simple Storage Service;

RAM is a Random Access Memory;

vCPU is a virtual Central processing unit;

GPU is a graphical processing unit;

TFLOPS is a Tera FLoating-point OPerations per Second;

Adam optimizer is an adaptive moment stochastic gradient descent method;

H, W, and C are image height, width, and number of channels respectively.

NOMENCLATURE

I_t is a training set of images;

I_G is a set of generated images;;

DOI 10.15588/1607-3274-2024-1-10

G is a generator;
D is a discriminator;
L is a set of discriminator layers;
Q is a set of generator layers;
n is a number of discriminator layers;
m is a number of generator layers;
i is an index of discriminator layers;
j is an index of generator layers;
$CELL_D$ is a cell of discriminator;
$CELL_G$ is a cell of generator;
o is a number of nodes in the generator cell;
p is a number of nodes in the discriminator cell;
A_G is a generator architecture;
A_D is a discriminator architecture;
O_G is a set of generator operations;
O_D is a set of discriminator operations;
P_G is a the set of parameters of generator operations;
P_D is a set of parameters of discriminator operations;
$V(G, D)$ is the loss function for the generator and discriminator;
q is the power of the training sample set;
r is the power of the set of the generated sample;
$P_{I_t}(x)$ is the density of the distribution function of the training sample;
$P(z)$ is the density of the generator's noise distribution function;
$E(x)$ is the expected value of a random variable x.
M_F is a FID metric;
$\langle C \rangle$ is a set of convolution functions;
$\langle K \rangle$ is a set of activation functions;
$\langle U \rangle$ is a set of operations of the Upsample block;
$\langle W \rangle$ is a set of operations of the generator cell $CELL_G$;
$\langle Z \rangle$ is a set of operations of the discriminator $CELL_D$;
$\langle T \rangle$ is a set of operations of the Downsample block;
S is a Self Attention operation.

INTRODUCTION

Image synthesis is a popular trend in artificial intelligence. A separate class of images are biomedical images. A biomedical image is a structural and functional image of human and animal organs, designed to diagnose diseases and study the anatomical and physiological picture of the body.

A subclass of biomedical images are cytological, histological and immunohistochemical images. These images are used to make a diagnosis in oncology. The widespread use of deep neural networks for image classification has led to the problem of datasets. To achieve the required accuracy of biomedical image classification, it is necessary to provide powerful datasets.

Hence, there is a contradiction between achieving high accuracy in biomedical image classification and providing powerful datasets to train convolutional neural networks. To resolve this contradiction, it is necessary to develop a method and means of generating biomedical images. One of the modern approaches to solve this problem is the use of generative adversarial networks. Since their inception in 2014, generative adversarial networks have become the primary tool for synthesizing high-quality and diverse images [1–5]

Synthesized images help improve the training of machine learning models (in particular, classifiers) by extending existing training datasets, which are often low-power [6].

The complex process of synthesizing GAN architectures manually makes them difficult to use. Manual design of GAN architecture requires a deep understanding of machine learning principles and knowledge of the specifics of the subject area. This process is time-consuming and often requires developers to make a number of complex decisions. For example, developers have to choose between different types of layers, activation functions, and optimization techniques that have a big impact on the performance of the neural network.

One of the main challenges of manual architecture design is scalability. The more complex biomedical imaging becomes, the more complex GAN architectures become. Manual architecture design depends on the experience and intuition of the developer.

Thus, the creation of an automatic method for searching for GAN network architectures is an urgent task. The automatic method of searching for GAN architectures will allow you to quickly and thoroughly explore a large search space (types of layers, activation functions, etc.), finding the best network configurations. This method will reduce the synthesis time of GAN architectures and increase the efficiency of the process.

The object of research is the process of biomedical images synthesis.

The subject of the research is the synthesis of generative-adversarial network architectures.

The aim of the work is to develop an automatic method for searching the architectures of generative-adversarial networks based on a genetic algorithm. This will make it possible to synthesize GAN network architectures in automatic mode for the synthesis of biomedical images.

1 PROBLEM STATEMENT

Let be given a training sample of images I_t with a cardinality of q. Generating a set of images I_G with a cardinality of r, and $r \gg q$. To generate I_G we use GAN. The GAN consists of a generator and a discriminator. In addition, the architecture of the generator and discriminator A_G and A_D, respectively, is given.

The discriminator architecture is described as follows:

$$A_D = \left\{ L_i, i = \overline{1,n} \right\},$$

and the architecture of the generator as follows:

DOI 10.15588/1607-3274-2024-1-10

$$A_G = \left\{ Q_j, j = \overline{1,m} \right\}.$$

The set of discriminator operations is represented as follows:

$$O_D = \left\{ \langle C \rangle; \langle K \rangle; \langle Z \rangle; \langle T \rangle \right\},$$

and the set of generator operations is as follows:

$$O_G = \left\{ \langle C \rangle; \langle K \rangle; \langle U \rangle; S \right\}.$$

Then it is necessary to carry out a two-level optimization of the discriminator and generator architectures, i.e.:

$$A_D = \arg\min_{O_D, P_D} M_F \left(P_D, O_D, I_t, I_G \right),$$
$$A_G = \arg\min_{O_G, P_G} M_F \left(P_G, O_G, I_t, I_G \right).$$

In this case, the loss function of the discriminator and the generator will be defined as follows:

$$V(G,D) = \min_G \max_D F(G,D) =$$
$$= E_{x \sim P_{I_t}(x)} \left(\log D(x) \right) + E_{z \sim p(z)} \left(\log(1 - D(G(z))) \right).$$

2 REVIEW OF THE LITERATURE

In deep learning, the development of generative adversarial networks has been a big step for image synthesis. Generative adversarial networks have a great potential for the synthesis of realistic images.

In the early days of GAN research, the main goal was to synthesize realistic images from random noise. The work of Radford, et al. gave a push to the improvement of the quality of synthesized images and the stability of GAN learning [7]. This work laid the groundwork for future research, as it proved GAN's ability to handle complex image distributions.

In biomedicine, GANs are used to augment training data, which is essential to address the scarcity of annotated medical images. The authors [8] presented an innovative approach to the synthesis of retinal images using the generative model method. Based on generative models, realistic retinal images were synthesized and used to train diagnostic algorithms. Likewise, the authors Frid-Adar et al. used generative adversarial networks to expand the liver lesion image datasets that were used by the classifier [9]. Thus, the authors demonstrated the practical utility of GAN for expanding datasets and improving the accuracy of neural networks in biomedical image classification tasks.

The complexity of designing neural network architectures has led to the development of automatic search for NAS neural network architectures. Zof and Le were among the first to introduce the concept of automatic architecture search for CNN. This concept has been adapted by researchers to automate the design of GAN architectures [10]. In their work, the authors applied the reinforcement learning method to explore the search space for architectures. This work shows the potential of NAS to reduce the human factor in the design of new neural network architectures. Since the field of NAS was primarily focused only on convolutional neural networks, NAS for generative networks is a relatively new area of research. Early work in NAS for GAN focused on finding efficient architectures that could generate high-quality images with reduced computing resources.

A fundamental paper in this area is [11], which develops a method for automatically searching for GAN architectures. The authors applied the reinforcement learning method to optimize the GAN architecture [12]. The AutoGAN framework has demonstrated that automated architecture design can compete with human-made models. To evaluate the method, the authors used the FID metric, the final value of which is 12.42. The architecture search time is 48 GPU hours. The main disadvantages of the work are: limited search space, fixed discriminator, unconditional image generation.

The development of the preliminary research is the work [13]. In the paper, the authors applied a genetic algorithm to optimize the neural network architecture and expanded the search space with operations such as skip connection, a convolution with different kernel sizes (1, 3, and 5, respectively) [14]. The advantage of this work is the search for the architecture of the generator and discriminator. The search for architecture is divided into two stages – the first stage is the search for the architecture of the generator (the discriminator is fixed), and the second stage is the search for the discriminator. According to the authors, this approach has significantly stabilized the training of the GAN network. The experiments took 1.2 GPU days and showed a FID of 9.91 on the CIFAR-10 dataset. The disadvantage of the work is the unconditional generation of images.

In terms of optimizing GAN architectures, researchers have identified several key factors that affect the performance of networks. Brock et al. introduced the concept of scaling GANs, showing that larger models and batch sizes can lead to improved image quality [15]. This concept is essential for automatically searching for GAN architectures. Search algorithms must take into account the trade-off between model complexity and computational capability.

In addition, the evaluation of biomedical images generated by GAN networks presents particular challenges. When evaluating synthesized images, it is necessary to take into account not only visual accuracy, but also the preservation of diagnostic features. Automatic search for GAN architectures should result in models that are statistically powerful and clinically relevant.

Despite the progress made, significant gaps remain in the literature. The analyzed articles in the field of NAS for GAN do not focus on the synthesis of biomedical images and use an open dataset – CIFAR. This can be

DOI 10.15588/1607-3274-2024-1-10

explained by the difficulty of obtaining the datasets of biomedical images themselves, which in turn further emphasizes the relevance of their synthesis. Also, the considered methods have a limited search space, do not use the most modern operations (ELU, Self-Attention) in the search space, do not synthesize images by tags (use unconditional generation).

The article [16] developed a method for the synthesis and classification of breast cancer histological images. At the same time, well-known GAN structures are used.

In this paper, we propose an automatic method for searching for architectures of generatively adversarial networks for the synthesis of biomedical images.

3 MATERIALS AND METHODS

Defining the search space is the first important step in the development of generative adversarial network architectures. In this step, you define the range of possible types of layers and their combinations that will be used in the process of building the model.

The search space in the developed method is cell-based. A cell consists of a certain number of nodes and operations between them. An example of a cell is shown in Figure 1.

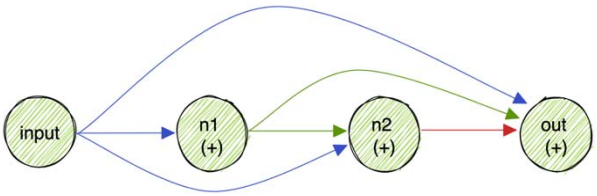


Figure 1 – Cell Structure

Figure 1 shows a cell with four nodes. The arrows represent the type of operation between the two nodes. Nodes *n1, n2, n3* signify the addition operation.

One of the disadvantages of classical neural network architectures is that all layers follow each other sequentially. This option of connection gives rise to the problem of gradient attenuation, which significantly impairs the training of the neural network [17]. In this method, each successive node in the cell can receive input from all the previous nodes, depending on the selected operation between them.

We analyzed modern architectures of GAN networks and identified a set of layers that are most often used in research. These include the following operations: kernel convolution operation 1×1, 3×3 and 5×5, max- and average-pooling operations, self-attention mechanism. Also, the generator and discriminator architectures in most GANs consist of repeating convolutional blocks, and the input noise vector is transformed into a matrix of size 4×4. The number of such blocks varies depending on the resolution of the generated image. Accordingly, if the generator synthesizes an image with a resolution of 64×64, then it consists of 4 convolutional blocks. The full set of possible operations we have chosen for the cell is listed below.

Zero. This operation replaces the node's output with a matrix of zeros. It can be used to remove a node from a cell architecture to simplify it.

Skip connection. This operation can be thought of as a function that receives and returns data. Using such an operation allows you to directly connect the output of one node in a cell to the input of another.

Convolutional block. batch normalization is often used along with convolution operations. Therefore, this layer uses three sequential operations: convolutions, activation functions, and batch normalization. We used convolution kernels 1×1, 3×3, and 5×5. As an activation function, ELU is used – an improved modification of ReLU.

Pooling. We use pooling only in the discriminator model. The two types of pooling are max pooling with a 3×3 kernel and step 1×1 and average pooling with a 3×3 kernel and step 1.

The Self-Attention operation is placed outside the cells. The integration of the Self-Attention mechanism into CNN allows to focus on the most informative features, which can be useful for object detection, segmentation and image recognition [18].

The search space for the generator and discriminator is shown in Figure 2.

In the developed method, we applied the Conditional GAN strategy, which allows to synthesize images of classes. Accordingly, in the generator model, we use the Conditional Batch Normalization operation in each UpSample block. In the discriminator model, we use an Embeddig layer in combination with a fully connected layer. The sum of the outputs of these two layers is the output of the discriminator model.

The generator consists of one deconvolution layer to convert the input noise vector to dimension 4×4×1024, four cells, four blocks to double the resolution, and two convolutional layers at the network output. The final resolution of the synthesized image is 64×64×3.

The main element in the generator's search space is the cell, which consists of *o* nodes. The available operations in a cell are defined as follows: convolution by the kernel 1×1, 3×3, 5×5; separable convolution by kernel 3×3; zero; skip connection. The cell architecture remains the same for the entire generator model. Accordingly, the search space of the generator is reduced to determining the number of nodes in the cell, selecting operations between nodes, as well as selecting layers after which you need to apply the Self-Attention operation through the residual connection (shown in the figure by *ATTN (represents Self-Attention)* arrows).

The generator's search space is encoded as {*N, C, [ATTN]}*. Let's review in details these parameters.

DOI 10.15588/1607-3274-2024-1-10

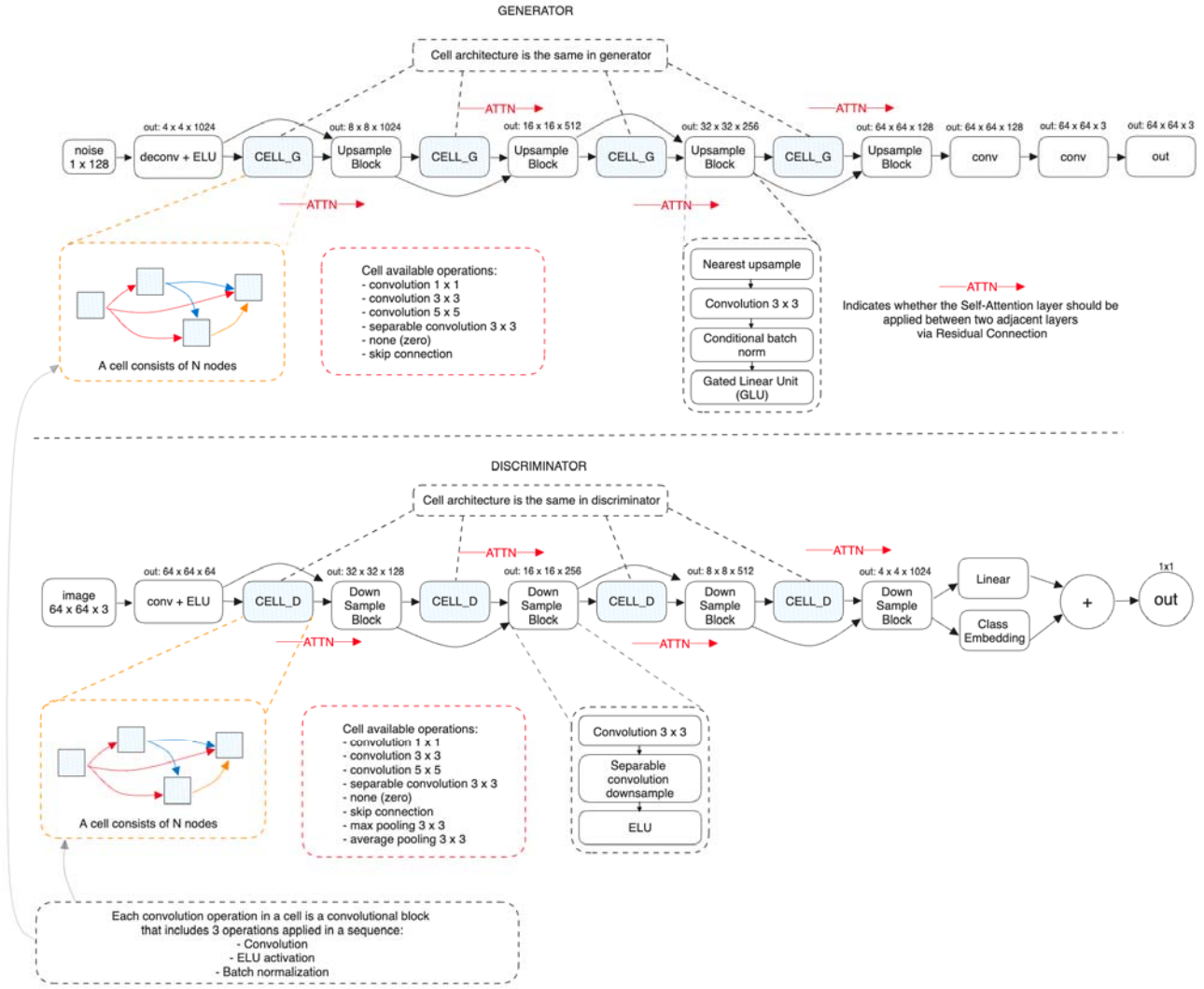


Figure 2 – Generator and Discriminator Search Spaces

The *N* parameter is the number of nodes in the cell. The minimum value is 3, the maximum value is 5. This decision was made due to limited computing resources. In addition, our study uses mostly small images, specifically 64×64 pixel in size. This aspect has a significant impact on network architectures, as such images usually require fewer layers. Therefore, the decision to limit the number of layers in the model is also due to the nature of the data.

Parameter *C* is the architecture of the cell in which the operations between nodes are encoded. It is represented by a tape *xxx–xx–x*, the length of which varies depending on the number of nodes in the cell. For example, for a cell with a number of nodes of 3, the encoded tape might look like 12–6. The first fraction separated by a hyphen represents the operations between the first node and all subsequent ones, the second – between the second node and all subsequent ones. Respectively 1 means kernel convolution operation 1×1 between the first and second nodes (see Fig. 2), 2 is the kernel convolution operation 3×3 between the first and second nodes. The number 6 means the skip connection operation between the second and third nodes.

The *ATTN* parameter represents the cell numbers after which you want to apply Self-Attention operations.

According to the described encoding and the above example, the encoding of the generator architecture is {3, 12–6, [1, 2]}. As you can see from the example, the Self-Attention operation is applied after the first and second cells.

The discriminator consists of one convolution layer, four cells, four blocks for halving the resolution, and one linear and embedding layer.

The main element in the discriminator search space is the cell, which consists of *p* nodes. Unlike the generator, the available set of operations in the discriminator cell is extended by two operations: the maximum pooling by the kernel 3×3 and the average pooling by the kernel 3×3.

For the dimensionality reduction operation, the Separable Downsample Convolution operation is used. The authors of the paper state that this is the most optimal method of reducing dimensionality in convolutional networks at present [19].

The cell architecture also remains the same for the entire discriminator model. The discriminator search

DOI 10.15588/1607-3274-2024-1-10

space boils down to the same steps as for the generator. These steps are as follows: determining the number of nodes in a cell, selecting operations between nodes, selecting layers. After these steps, you need to apply the Self-Attention operation through the residual connection (shown by the ATTN arrows in the figure).

The discriminator's search space is encoded in the same way as the generator's: *{N, C, [ATTN]}*.

The developed method consists of the following stages:

1. Search for a generator architecture with a fixed discriminator.

2. Search for a discriminator architecture with the best generator.

The general framework of the method is shown in Figure 3.

At the first stage, we define a fixed discriminator architecture and search for only the generator. The discriminator architecture is manually defined as follows: four nodes in a cell, the cell architecture is encoded as 235–66–7 (convolution by kernel 3×3 and 5×5, zero, skip connection, skip connection, max pooling by kernel 3×3), Self-Attention mechanism is applied only after the last cell.

At each cycle of the optimization algorithm, we initialize the population with random generator architectures. Next, we create a generator and discriminator pair for each generator in the population. The resulting pairs of GAN networks are trained and evaluated at the end of each cycle using FID metrics [20, 21]. At the end of the cycle, we select the generator with the lowest FID value and copy its weights. Further in the next cycle, we initialize all generators with copied weights. Accordingly, after the first step, we get the architecture of the best generator, i.e. the model with the lowest FID value.

In the second step, we use the best generator architecture and search for the discriminator architecture. At each cycle of the optimization algorithm, we create a population of discriminators and initialize it with random architectures. Then, by analogy with the first step, we create pairs of generators and discriminators. The resulting pairs of GAN networks are trained independently of each other and evaluated using the FID metric. At the end of each cycle, we copy the weights of the best discriminator and initialize all the discriminators in the next cycle with them. After the second step, we get the architecture of the best discriminator, and, accordingly, the best architecture of the GAN network.

At both stages, the input images of the discriminatormodel are subjected to the technique of Differentiable Augmentations with the application of a random color and translation policy. This technique helps to further expand the training dataset directly during the learning process and is especially effective on small datasets [22].

We used a modified Aging Evolution GA (AGA) algorithm to optimize the generative adversarial network architecture [23]. The AGA maintains a population of potential solutions, where each solution represents a unique GAN architecture.

A variation in the genetic algorithm known as Aging Evolution GA adds an aging component to increase population diversity and prevent early convergence.

The main steps of this algorithm are:

1. Train and evaluate each architecture in the population by calculating the FID metric.

2. Selection of a subset of architectures from the population. The selection process is based on a random strategy.

3. Selection of the parent architecture and application of the genetic operator (mutation) to it to create a new architecture.

4. Evaluate the new architecture by training it and computing the FID metric and adding the architecture to the population.

5. Remove the oldest architecture from the population.

At the first step of the algorithm, the population is initialized by random architectures to achieve a given population size – *population_size.* Simultaneously with initialization, architectures are trained with the calculation of the FID metric value.

Further, evolution occurs cyclically. The number of cycles is also set by the user (*cycles* parameter*).* On each of the cycles, a given number of architectures is randomly selected – sample_size. Then, among these architectures, the one with the lowest FID value is selected. This architecture is called the parent architecture.

Based on the parent architecture, a new child architecture is created by applying a mutation. In this algorithm, the mutation changes the architecture randomly. Next, the mutated architecture learns.

The mutation is applied to each element in the encoded architecture with a probability of mutation_prob. For example, the network architecture is set to tape *107590*. In order to mutate, you need to go through each element of the tape in a loop and change it randomly with a given probability *of mutation_prob.*

Accordingly, after applying a mutation, we get a new child architecture that is added to the population. At the same time, the oldest architecture is being removed from the population. The algorithm then returns the architecture with the lowest FID value.

DOI 10.15588/1607-3274-2024-1-10

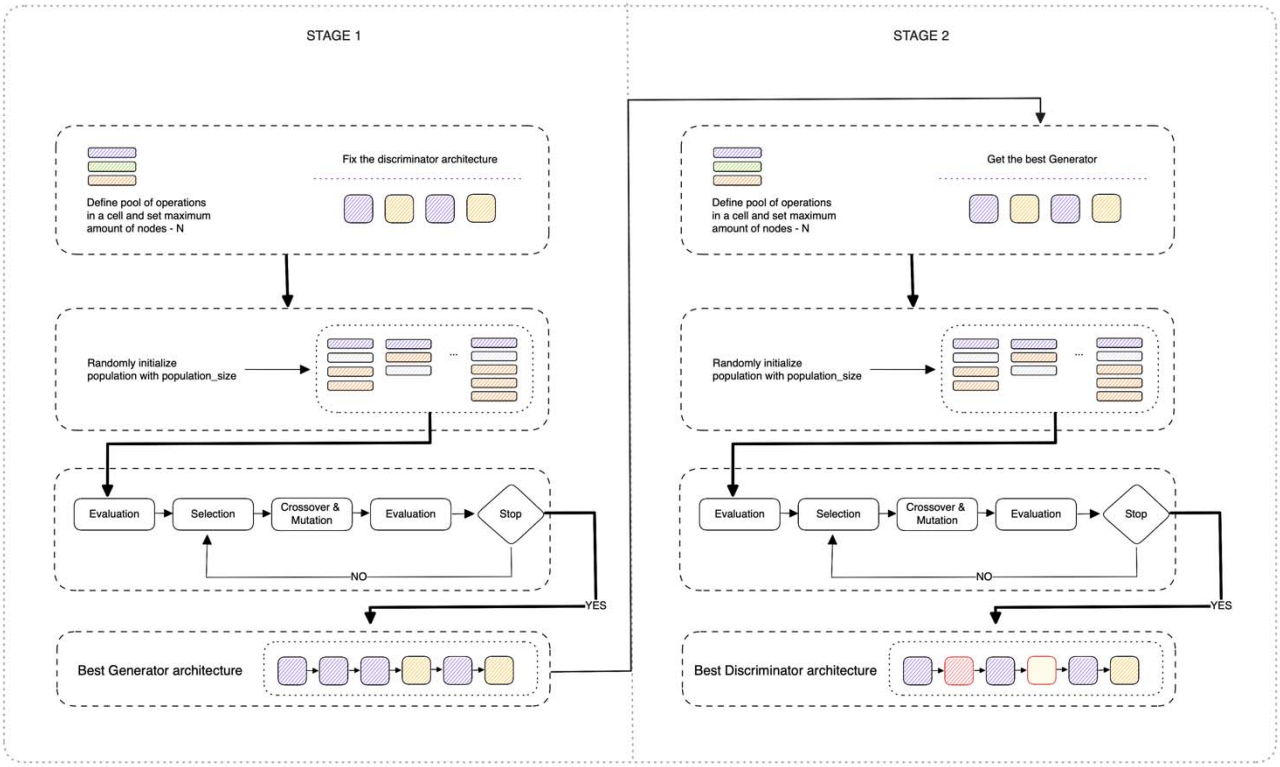


Figure 3 – General GAN Synthesis Method Framework

4 EXPERIMENTS

The experiments were based on the Python 3.10 programming language, the PyTorch 2.0 framework, and a virtual machine with the following configuration: 32 GB RAM, 7 vCPU, Nvidia A6000 GPU, 48 GB RAM.

The software module consists of two main Python scripts: train.py and generate.py. The train.py script is designed to build and train generative adversarial network (GAN) architectures. It allows users to define basic parameters for GAN search, including noise vector dimension, activation functions, and optimization parameters. After training, the architecture that gives the best results is chosen. The trained model is stored in Amazon S3 storage, ensuring that the model is saved and can be accessed or downloaded as needed.

The second script, generate.py, runs after the GAN model has been trained and configured. This script is responsible for generating new images using the best architecture found during the learning phase. It can be run as a command-line tool where the user specifies the path to the model and the preferred directory to output the generated images. The script loads the trained GAN model and uses it to synthesize new images that are expected to demonstrate the learned distribution of the training data. The results can be used for further analysis or as input for other stages of research.

In this study, we used cytological images to test the method. A cytological image is a microscopic image of individual cells or cell formations obtained by cytological examination.

Cytological images of breast cancer on the Zenodo platform were used for the experiments [24]. This dataset is designed to test and configure automatic biomedical image processing systems. The structure of the dataset is as follows:

1) files of cytological images and the indicated diagnosis (size 3264×2448).

2) files of histological and immunohistochemical images of sections of breast tissue (size 2048×1536) and the indicated diagnosis.

Examples of cytological images are shown in Figure 4 (one image per class).

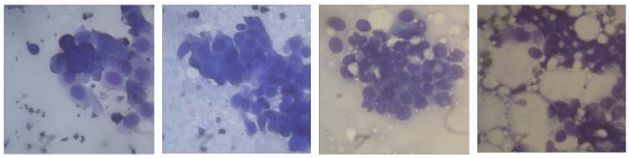


Figure 4 – Example of images from the dataset

For the experiments, the image was converted to a resolution of 64×64 pixels. Since the initial number of images in the sample is quite small (about 100 images per class), it was expanded to 700 images per class by applying affine distortions (random rotation, displacement, twisting, etc.) [25].

For both stages of the search, we used the Hinge loss function [26] and the Adam optimizer (betas = 0.5, 0.999) [27]. We also applied the Two Time-scale Update Rule [28]. Accordingly, the learning rate of the generator is 0.0001, and the discriminator is 0.0004.

DOI 10.15588/1607-3274-2024-1-10

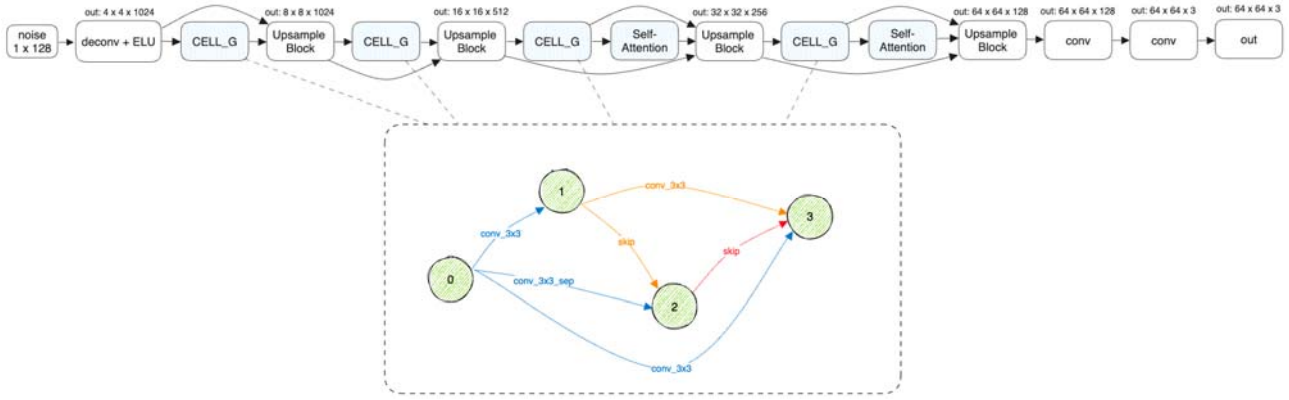


Figure 5 – Synthesized Generator Architecture

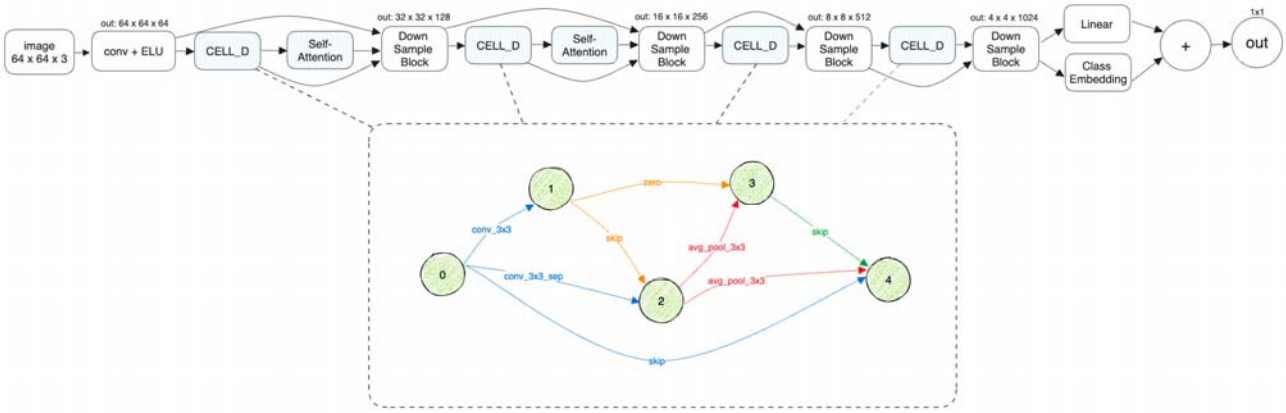


Figure 6 – Synthesized Discriminator Architecture

For all convolutional, deconvolutional, and linear layers in both models, we applied the spectral normalization technique, which allows us to stabilize the learning of the GAN network [29].

The total number of cycles of the optimization algorithm at each stage is 25, and the number of epochs of training for each model in the population is 30. The population size (population_size) is 100. The number of randomly selected candidates for further selection of the parent architecture for mutation (sample_size) is 25. The probability of mutation (mutation_prob) is 0.05. Batch size (batch_size) for the generator and discriminator is the same and is 128 images.

To evaluate architectures, the FID metric was used, which is calculated after each architecture is trained.

In total, the first and second stages took 15.6 and 10.3 GPU hours, respectively.

Upon completion of both phases, the resulting GAN network was trained from scratch for 100,000 iterations. It took another ~13.6 GPU hours. That is, the total time spent from finding architectures to obtaining a fully trained GAN network is 39.5 GPU hours.

As a result of the experiments, 4000 images with a resolution of 64×64 for each class from the educational dataset were synthesized. Accordingly, the total number of synthesized images is 16,000.

5 RESULTS

The architecture of the found generator and discriminator is shown in Figures 5 and 6 and in Tables 1–4.

As you can see from the figures, the number of nodes in the generator and discriminator cells is 4 and 5, respectively. There are two skip connection operations in the generator cell, and there are 3 in the discriminator cell. There is also a zero operation in the discriminator cell, which is not present in the generator. The Self-Attention operation is applied 2 times in both the generator and the discriminator. However, in the generator, this operation is placed towards the end of the network, And in the discriminator, on the contrary, it is closer to the beginning.

The FID metric value for the found GAN network architecture is 3.39, and the IS metric value is 3.95.

Examples of comparison of synthesized images with the original ones for each class are shown in Figures 7–10. The images are selected randomly.

DOI 10.15588/1607-3274-2024-1-10

Table 1 – Generator Structure

Layer name	Params	Output shape
L1: Input	Gaussian noise	1×128
L2: Transposed Conv + ELU activation	Kernel = 4, stride = 1, padding = 0	4×4×1024
L3: $CELL_G$	Nodes = 4	4×4×1024
L4: L2 + L3		4×4×1024
L5: Upsample	Scale = 2	8×8×1024
L6: $CELL_G$	Nodes = 4	8×8×1024
L7: L5 + L6		8×8×1024
L8: Upsample	Scale = 2	16×16×512
L9: $CELL_G$	Nodes = 4	16×16×512
L10: Self Attention	Input channels = 512	16×16×512
L11: L8 + L10 + L9		16×16×512
L11: Upsample	Scale = 2	32×32×256
L12: $CELL_G$	Nodes = 4	32×32×256
L13: Self Attention	Input channels = 256	32×32×256
L14: L11 + L13 + L12		32×32×256
L15: Upsample	Scale = 2	64×64×128
L16: Convolution	Kernel = 3, stride = 1, padding = 1	64×64×128
L17: Convolution	Kernel = 3, stride = 1, padding = 1	64×64×3
L18: Output		64×64×3

Table 2 – Generator $CELL_G$ Structure

Layer name	Params	
L0: Input		
L1: Conv → ELU → Batch Norm	Kernel = 3, stride = 1, padding = 1	
L2: L1 + Conv 3×3 → Conv 1×1 → ELU → Batch Norm	Conv 3x3 = (Kernel = 3, stride = 1, padding = 1), Conv 1x1 = (Kernel = 1, stride = 1, padding = 0)	
L3: L2 + Conv (L1) + Conv (L0)	Kernel = 3, stride = 1, padding = 1	
L0: Input		
L1: Conv → ELU → Batch Norm	Kernel = 3, stride = 1, padding = 1	
Upsample block structure		
Layer name	Params	Output shape
L0: Input		H × W × C
L1: Upsample	Scale = 2, mode = nearest	(H × 2) × (W × 2) × C
L2: Convolution	Kernel = 3, stride = 1, padding = 1	(H × 2) × (W × 2) × C
L3: Conditional Batch Norm	Number of classes = 4	(H × 2) × (W × 2) × C
L4: Gated Linear Unit (GLU)	Dimension = 1	(H × 2) × (W × 2) × (C / 2)

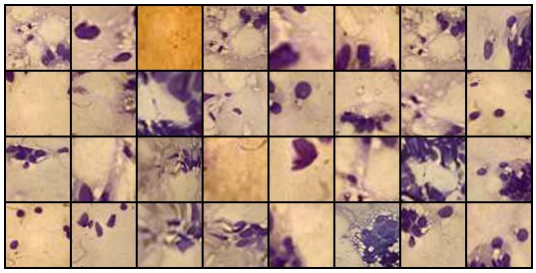
Table 3 – Discriminator Structure

Layer name	Params	Output shape
L1: Input	Image	64×64×3
L2: Conv + ELU activation	Kernel = 3, stride = 1, padding = 1	64×64×64
L3: $CELL_D$	Nodes = 5	64×64×64
L4: Self Attention	Input channels = 64	64×64×64
L5: L2 + L4 + L3		64×64×64
L6: Downsample	Scale = 2	32×32×128
L7: $CELL_D$	Nodes = 5	32×32×128
L8: Self Attention	Input channels = 64	32×32×128
L9: L6 + L8 + L7		32×32×128
L10: Downsample	Scale = 2	16×16×256
L11: $CELL_D$	Nodes = 5	16×16×256
L12: L10 + L11		16×16×256
L13: Downsample	Scale = 2	8×8×512
L14: $CELL_D$	Nodes = 5	8×8×512
L15: L13 + L14		8×8×512
L16: Downsample	Scale = 2	4×4×1024
L17: Linear(Sum(L16))		1×1
L18: Sum(Multiply(Sum(L16), Embedding))	Number of classes = 4	1×1
L19: L17 + L18		1×1
L20: Output		1×1

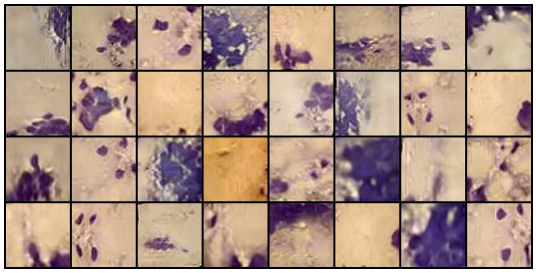
DOI 10.15588/1607-3274-2024-1-10

Table 4 – Discriminator $CELL_D$ Structure

Layer name	Params	
L0: Input		
L1: Conv → ELU → Batch Norm	Kernel = 3, stride = 1, padding = 1	
L2: L1 + Conv 3×3 → Conv 1×1 → ELU → Batch Norm	(Kernel = 3, stride = 1, padding = 1), (Kernel = 1, stride = 1, padding = 0)	
L3: AvgPool 3× 3 (L2)	Kernel = 3, stride = 1	
L4: L0 + L3 + AvgPool 3× 3 (L2)	Kernel = 3, stride = 1	
Downsample block structure		
Layer name	Params	Output shape
L0: Input		H × W × C
L2: Convolution	Kernel = 3, stride = 1, padding = 1	H × W × (C × 2)
L3: Pixel Rearrange → Convolution	Kernel = 1, stride = 1, padding = 0	(H / 2) × (W / 2) × (C × 2)
L4: Exponential Linear Unit (ELU)		(H / 2) × (W / 2) × (C × 2)

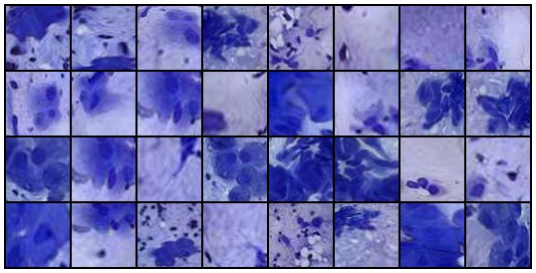


a

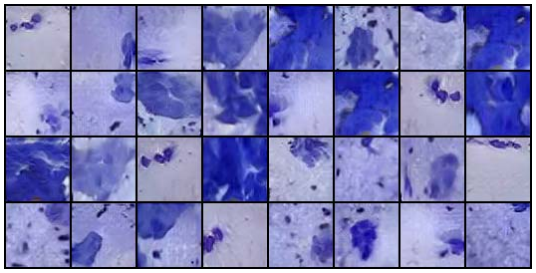


b

Figure 7 – Original (a) and Synthesized (b) Images, Class 1

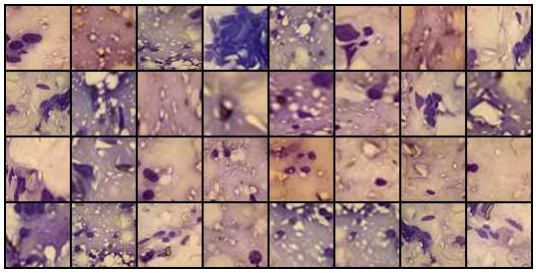


a

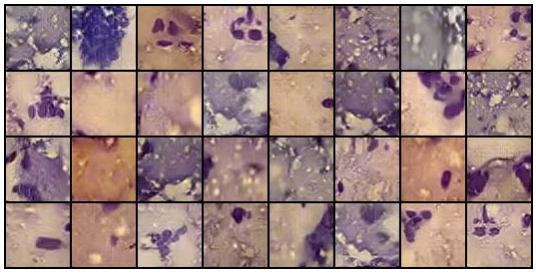


b

Figure 9 – Original (a) and Synthesized (b) Images, Class 3

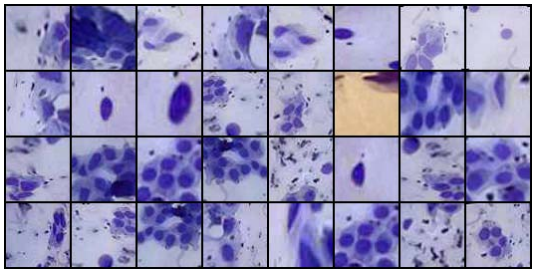


a

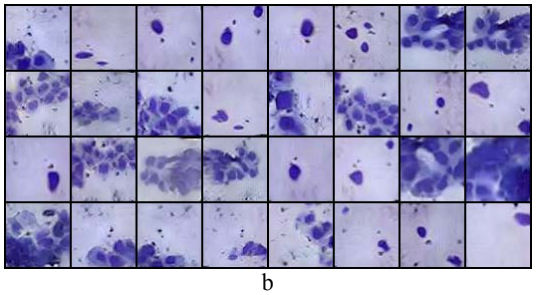


b

Figure 8 – Original (a) and Synthesized (b) Images, Class 2



a



b

Figure 10 – Original (a) and Synthesized (b) Images, Class 4

DOI 10.15588/1607-3274-2024-1-10

We also compared different GAN architectures for the synthesis of cytological images. The results of the comparison are shown in Table 5.

Table 3 – Comparison of GANs by FID metric using same images

Method	FID
DCGAN	12.67
WGAN	12.72
WGAN-GP	19.09
BGAN	10.03
BEGAN	15.32
Our method GA-GAN	3.39

6 DISCUSSION

As a result of the study, the architecture of the generative-adversarial network for the synthesis of cytological images was obtained (Fig. 5, Fig. 6). To assess the quality of synthesized images, the FID metric was used.

Table 3 shows that the network architecture designed by our method showed the best results compared to other GAN network architectures for the same images.

Unlike the above architectures, our method uses the Self-Attention mechanism in the generator and discriminator, which allowed us to improve the quality of synthesized images. Also, our method supports the mechanism of image synthesis by labels (conditional generation), which is not relevant for the above architectures and approaches.

Figures 7–10 show a comparison of original pairs and synthesized images for each class from the original and synthesized dataset. The synthesized images are difficult to visually distinguish from the original ones, which further indicates the power of the resulting network.

We did not test the method on higher-resolution images, as this would have led to an increase in search time. Therefore, the limitation of our research is the relatively low resolution of the synthesized images – 64×64 pixels. In order to synthesize images of higher resolution, you need to increase the number of cells in the generator and discriminator.

We also conducted experiments on only one subclass of biomedical images – cytological images. Accordingly, a further direction of research may be testing and adaptation of the developed method to other classes and resolutions of biomedical images.

CONCLUSIONS

As a result of the study,the automatic method for searching for architectures of generatively adversarial networks for the tasks of synthesis of cytological images was developed.

Architectural search space is defined in terms of cells, which consist of a set of nodes and operations between them. The architectural features of the cell allow you to expand the search space and reduce the likelihood of a gradient attenuation problem.

The developed method consists of two stages: the search for the architecture of the generator with a fixed discriminator and the search for the architecture of the discriminator paired with the fixed best generator.

As a result of computer experiments, the architecture of a generatively competitive network for the synthesis of cytological images was obtained. The total time of the experiment was ~39.5 GPU hours. As a result, 16,000 images were synthesized (4000 for each class).

Comparison of the synthesized architecture with other architectures of generative-adversarial networks, using the same training dataset, is carried out on the basis of the FID metric. The results showed that the designed architecture is the best. The FID value of the developed network (3.39) is two and a half times better than the FID metric of the above architectures.

The scientific novelty is the development of a method for finding generative-adversarial network architectures for the synthesis of biomedical images.

The practical significance is the development of a software module for the synthesis of biomedical images that can be used to train CNN.

The authors of the article have many years of experience in the development of biomedical image analysis systems [30–34].

A software module for the synthesis of biomedical images will be integrated into image analysis systems.

Prospects for further research is the development of a CAD system for the classification and synthesis of biomedical images.

REFERENCES

1. Goodfellow I., Pouget-Abadie J., Mirza M. et al. Generative Adversarial Nets, *Advances in Neural Information Processing Systems*, 2014, №27.
2. Salimans T., Goodfellow I., Zaremba W. et al. Improved Techniques for Training GANs [Electronic resource], *arXiv.org,* 2016. Access mode: https://arxiv.org/abs/1606.03498.
3. Karras T., Aila T., Laine S. et al. Progressive Growing of GANs for Improved Quality, Stability, and Variation [Electronic resource], *arXiv.org*, 2017. Access mode: https://arxiv.org/abs/1710.10196.
4. Analyzing and Improving the Image Quality of StyleGAN [Electronic resource], *Conference on Computer Vision and Pattern Recognition (CVPR) : proceedings, Seattle,* 2020,P. 8107–8116. Access mode: https://doi.org/10.1109/CVPR42600.2020.00813.
5. Arjovsky M., Chintala S. Wasserstein Generative Adversarial Networks [Electronic resource], *International Conference on Machine Learning : proceedings*, 2017. Access mode: https://leon.bottou.org/publications/pdf/icml-2017.pdf.
6. Düzyel O., Kuntalp M. Data Augmentation with GAN increases the Perfor-mance of Arrhythmia Classification for an Unbalanced Dataset [Electronic resource], arXiv.org. – 2023. Access mode: https://doi.org/10.48550/arXiv.2302.13855.

DOI 10.15588/1607-3274-2024-1-10

7. Radford A., Metz L. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks [Electronic resource], arXiv.org, 2016. Access mode: https://doi.org/10.48550/arXiv.2302.13855.
8. Costa Pedro et al. End-to-End Adversarial Retinal Image Synthesis [Electronic resource], *IEEE Transactions on Medical Imaging*, 2018, Vol. 37, № 3, pp. 781–791. Access mode: https://doi.org/10.1109/tmi.2017.2759102.
9. Frid-Adar Maayan et al. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification [Electronic resource], *Neurocomputing.* 2018, Vol. 321, pp. 321–331. Access mode: https://doi.org/10.1016/j.neucom.2018.09.013.
10. Zoph B., Le Q. Neural Architecture Search with Reinforcement Learning [Electronic resource], *arXiv.org,* 2017. Access mode: https://doi.org/10.48550/arXiv.1611.01578.
11. Gong Xinyu et al. AutoGAN: Neural Architecture Search for Generative Adversarial Networks [Electronic resource], *2019 IEEE/CVF International Conference on Computer Vision (ICCV).* Seoul, Korea (South), 2019. Access mode: https://doi.org/10.1109/iccv.2019.00332.
12. Baker B., Gupta O. Designing Neural Network Architectures using Rein-forcement Learning [Electronic resource], *arXiv.org,* 2017. Access mode: https://doi.org/10.48550/arXiv.1611.02167.
13. Ying G., He X., Gao B. et al. Efficient Two-stage Evolutionary Architecture Search for GANs [Electronic resource], *arXiv.org,* 2020. Access mode: https://doi.org/10.48550/arXiv.2111.15097.
14. Gao Chen et al. AdversarialNAS: Adversarial Neural Architecture Search for GANs [Electronic resource], *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA, 2020. Access mode: https://doi.org/10.1109/cvpr42600.2020.00572.
15. Brock A., Donahue J., Simonyan K. Large Scale GAN Training for High Fidelity Natural Image Synthesis [Electronic resource], *Semantic Schoola,* 2018. Access mode: https://www.semanticscholar.org/paper/Large-Scale-GAN-Training-for-High-Fidelity-Natural-Brock-Donahue/22aab110058ebbd198edb1f1e7b4f69fb13c0613.
16. Berezsky O. M., Liashchynskyi P. B., Pitsun O. Y. et al. Deep network-based method and software for small sample biomedical image generation and classification, *Radio Electronics, Computer Science, Control*, 2024, № 4, P. 76.
17. He Kaiming et al. Deep Residual Learning for Image Recognition [Electronic resource], *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA, 2016. Access mode: https://doi.org/10.1109/cvpr.2016.90.
18. Vaswani A., Shazeer N., Parmar N. et al. Attention is All you Need [Electronic resource], *Semantic Schoolar,* 2017. Access mode: https://www.semanticscholar.org/paper/Attention-is-All-you-Need-Vaswani-Shazeer/204e3073870fae3d05bcbc2f6a8e263d9b72e776.
19. Sunkara R., Luo T. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects [Electronic resource], Machine Learning and Knowledge Discovery in Databases. Cham, 2023, pp. 443–459. Access mode: https://doi.org/10.1007/978-3-031-26409-2_27.
20. Barratt S., Sharma R. A Note on the Inception Score. [Electronic resource], *arXiv.org,* 2018. Access mode: https://arxiv.org/abs/1801.01973.
21. Borji A. Pros and cons of GAN evaluation measures [Electronic resource], *Computer Vision and Image Understanding*, 2019, Vol. 179, pp. 41–65. Access mode: https://doi.org/10.1016/j.cviu.2018.10.009.
22. Zhao S., Liu Z., Lin J. et al. Differentiable Augmentation for Data-Efficient GAN Training [Electronic resource], arXiv.org, 2020. Access mode: https://arxiv.org/abs/2006.10738.
23. Real Esteban et al. Regularized Evolution for Image Classifier Architecture Search [Electronic resource], *Proceedings of the AAAI Conference on Artificial Intelligence,* 2019, Vol. 33, pp. 4780–4789. Access mode: https://doi.org/10.1609/aaai.v33i01.33014780.
24. Cytological and histological images of breast cancer. (n.d.) [Electronic resource] // Zenodo. – 2023. – Access mode: https://doi.org/10.5281/zenodo.7890874.
25. Liashchynskyi P. Rudi – Python library [Electronic resource], GitHub, 2018. Access mode: https://github.com/liashchynskyi/rudi.
26. Ye J., Lim J. Geometric GAN [Electronic resource], *arXiv.org,* 2017. – Access mode: https://arxiv.org/abs/1705.02894.
27. Kingma D., Ba J. Adam: A Method for Stochastic Optimization [Electronic resource], *arXiv.org,* 2017. Access mode: https://arxiv.org/abs/1412.6980.
28. Bynagari N. B. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium [Electronic resource], *Asian Journal of Applied Science and Engineering.* 2019, Vol. 8, № 1, pp. 25–34. Access mode: https://doi.org/10.18034/ajase.v8i1.9.
29. Miyato T., Kataoka T., Koyama M. Spectral Normalization for Generative Adversarial Networks [Electronic resource], *arXiv.org,* 2018. Access mode: https://arxiv.org/abs/1802.05957.
30. Berezsky O. M. Liashchynskyi P. B. Comparison of generative adversarial networks architectures for biomedical images synthesis [Electronic resource], *Applied Aspects of Information Technology*, 2021, Vol. 4, № 3, pp. 250–260. Access mode: https://doi.org/10.15276/aait.03.2021.4.
31. Berezsky O. M., Berezska K. M., Melnyk G. M. et al. Design of computer systems for biomedical image analysis, *Proceedings of the X th International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics» CADSM 2009*. Lviv, Publishing House Vezha&CoC, pp. 186–191.
32. Berezsky O. Verbovyy S., Datsko T. The intelligent system for diagnosing breast cancers based on image analysis [Electronic resource], *2015 Information Technologies in Innovation Business Conference (ITIB).* Kharkiv, Ukraine, 7–9 October 2015, 2015. Access mode: https://doi.org/10.1109/itib.2015.7355067.
33. Berezsky Oleh et al. Fuzzy system diagnosing of precancerous and cancerous conditions of the breast [Electronic resource, *2016 XIth International Scientific and Technical Conference "Computer Sciences and Information Technologies (CSIT).* Lviv, Ukraine, 6–10 September 2016. 2016. Access mode: https://doi.org/10.1109/stc-csit.2016.7589906.
34. Berezsky O., Verbovyy S., Pitsun O. Hybrid Intelligent Information Technology for Biomedical Image Processing [Electronic resource], *IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies,* 2018. Access mode: https://ieeexplore.ieee.org/document/8526711.
35. Berezsky O., Dubchak L., Batryn T.et al. Fuzzy System For Breast Disease Diagnosing Based On Image Analysis [Elec-

DOI 10.15588/1607-3274-2024-1-10

tronic resource], *Proceedings of the II International Workshop Informatics & Data-Driven Medicine (IDDM 2019). –* 2019. – Access mode: http://ceur-ws.org/Vol-2488/paper6.pdf.

Received 19.01.2024.
Accepted 28.02.2024.

УДК 004.93

МЕТОД ПОШУКУ АРХІТЕКТУР ГЕНЕРАТИВНО-ЗМАГАЛЬНИХ МЕРЕЖ ДЛЯ СИНТЕЗУ БІОМЕДИЧНИХ ЗОБРАЖЕНЬ

Березький О. М. – д-р техн. наук, професор, професор кафедри автоматизованих систем управління НУ «Львівська політехніка», Львів, Україна.
Лящинський П. Б. – аспірант кафедри автоматизованих систем управління НУ «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. У статті досліджено проблему автоматичного проектування архітектур генеративно-змагальних мереж. Генеративно-змагальні мережі використовуються для синтезу зображень. Особливо це актуально для синтезу біомедичних зображень –цитологічних і гістологічних, які використовуються для постановки діагнозу в онкології. Синтезовані зображення використовуються для навчання згорткових нейронних мереж. Згорткові нейронні мережі є одними із найточніших класифікаторів біомедичних зображень на сьогодні..

Мета роботи – це розробка автоматичного методу для пошуку архітектур генеративно-змагальних мереж на основі генетичного алгоритму.

Метод. Розроблений метод складається з етапу пошуку архітектури генератора з фіксованим дискримінатором і етапу пошуку архітектури дискримінатора із найкращим генератором. На першому етапі визначається фіксована архітектура дискримінатора та здійснюється пошук генератора. Відповідно після першого кроку отримується архітектура найкращого генератора, тобто модель із найнижчим значенням FID.

На другому етапі використано найкращу архітектуру генератора та проводено пошук аріхтектури дискримінатора. На кожному циклі алгоритму оптимізації створюється популяція дискримінаторів. Після другого кроку отримується аріхтектура генеративно-змагальної мережі.

Результати. Для проведення експериментів використано цитологічні зображення раку молочної залози на платформі Zenodo. В результаті дослідження розроблено автоматичний метод пошуку архітектур генеративно змагальних мереж.В результаті комп'ютерних експериментів отримано архітектуру генеративно змагальної мережі для синтезу цитологічних зображень. Загальний час експерименту склав ~39.5 GPU годин. В результаті синтезовано 16 000 зображень (по 4000 на кожен клас). Для оцінки якості синтезованих зображень використано метрику FID . Результати експериментів показали, що розроблена архітектура є найкращою. Значення FID мережі становить 3.39. Цей результат є найкращим, порівняно з відомими генеративно-змагальними мережами.

Висновки. У статті розроблено метод пошуку архітектур генеративно-змагальних мереж для задач синтезу біомедичних зображень. Крім цього розроблено програмний модуль для синтезу біомедичних зображень, який може бути використаний для навчання CNN.

КЛЮЧОВІ СЛОВА: генеративно-змагальна мережа, біомедичні зображення, цитологічні зображення, пошук архітектур нейронних мереж, генетичні алгоритми, метрика FID, комп'ютерні системи автоматичної діагностики.

ЛІТЕРАТУРА

1. Generative Adversarial Nets / [I. Goodfellow, J. Pouget-Abadie, M. Mirza et al.] // Advances in Neural Information Processing Systems. – 2014. – № 27.
2. Improved Techniques for Training GANs [Electronic resource] / [T. Salimans, I. Goodfellow, W. Zaremba et al.] // arXiv.org. – 2016. – Access mode: https://arxiv.org/abs/1606.03498.
3. Progressive Growing of GANs for Improved Quality, Stability, and Variation [Electronic resource] / [T. Karras, T. Aila, S. Laine et al.] // arXiv.org. – 2017. – Access mode: https://arxiv.org/abs/1710.10196.
4. Analyzing and Improving the Image Quality of StyleGAN [Electronic resource] // Conference on Computer Vision and Pattern Recognition (CVPR) : proceedings, Seattle. – 2020. – P. 8107–8116. – Access mode: https://doi.org/10.1109/CVPR42600.2020.00813.
5. Arjovsky M. Wasserstein Generative Adversarial Networks [Electronic resource] / M. Arjovsky, S. Chintala // International Conference on Machine Learning : proceedings. – 2017. – Access mode: https://leon.bottou.org/publications/pdf/icml-2017.pdf.
6. Düzyel O. Data Augmentation with GAN increases the Perfor-mance of Arrhythmia Classification for an Unbalanced Dataset [Electronic resource] / O. Düzyel, M. Kuntalp // arXiv.org. – 2023. – Access mode: https://doi.org/10.48550/arXiv.2302.13855.
7. Radford A. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks [Electronic resource] / A. Radford, L. Metz // arXiv.org. – 2016. – Access mode: https://doi.org/10.48550/arXiv.2302.13855.
8. End-to-End Adversarial Retinal Image Synthesis [Electronic resource] / [Pedro Costa et al.] // IEEE Transactions on Medical Imaging. – 2018. – Vol. 37, № 3. – P. 781–791. – Access mode: https://doi.org/10.1109/tmi.2017.2759102.
9. GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification [Electronic resource] / [Maayan Frid-Adar et al.] // Neurocomputing. – 2018. – Vol. 321. – P. 321–331. – Access mode: https://doi.org/10.1016/j.neucom.2018.09.013.
10. Zoph B. Neural Architecture Search with Reinforcement Learn-ing [Electronic resource] / B. Zoph, Q. Le // arXiv.org. – 2017. – Access mode: https://doi.org/10.48550/arXiv.1611.01578.

DOI 10.15588/1607-3274-2024-1-10

11. AutoGAN: Neural Architecture Search for Generative Adversarial Networks [Electronic resource] / [Xinyu Gong et al.] // 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South). – 2019. – Access mode: https://doi.org/10.1109/iccv.2019.00332.
12. Baker B. Designing Neural Network Architectures using Rein-forcement Learning [Electronic resource] / B. Baker, O. Gupta // arXiv.org. – 2017. – Access mode: https://doi.org/10.48550/arXiv.1611.02167.
13. Efficient Two-stage Evolutionary Architecture Search for GANs [Electronic resource] / [G. Ying, X. He, B. Gao et al.] // arXiv.org. – 2020. – Access mode: https://doi.org/10.48550/arXiv.2111.15097.
14. AdversarialNAS: Adversarial Neural Architecture Search for GANs [Electronic resource] / [Chen Gao et al.] // 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA. – 2020. – Access mode: https://doi.org/10.1109/cvpr42600.2020.00572.
15. Brock A. Large Scale GAN Training for High Fidelity Natural Image Synthesis [Electronic resource] / A. Brock, J. Donahue, K. Simonyan // Semantic Schoolar. – 2018. – Access mode: https://www.semanticscholar.org/paper/Large-Scale-GAN-Training-for-High-Fidelity-Natural-Brock-Donahue/22aab110058ebbd198edb1f1e7b4f69fb13c0613.
16. Deep network-based method and software for small sample biomedical image generation and classification / [O. M. Berezsky, P. B. Liashchynskyi, O. Y. Pitsun et al.] // Radio Electronics, Computer Science, Control. – 2024. – № 4. – P. 76.
17. Deep Residual Learning for Image Recognition [Electronic resource] / [Kaiming He et al.] // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA. – 2016. – Access mode: https://doi.org/10.1109/cvpr.2016.90.
18. Attention is All you Need [Electronic resource] / [A. Vaswani, N. Shazeer, N. Parmar et al.] // Semantic Schoolar. – 2017. – Access mode: https://www.semanticscholar.org/paper/Attention-is-All-you-Need-Vaswani-Shazeer/204e3073870fae3d05bcbc2f6a8e263d9b72e776.
19. Sunkara R. No More Strided Convolutions or Pooling: A New CNN Building Block for Low-Resolution Images and Small Objects [Electronic resource] / Raja Sunkara, Tie Luo // Machine Learning and Knowledge Discovery in Databases. – Cham, 2023. – P. 443–459. – Access mode: https://doi.org/10.1007/978-3-031-26409-2_27.
20. Barratt S. A Note on the Inception Score. [Electronic resource] / S. Barratt, R. Sharma // arXiv.org. – 2018. – Access mode: https://arxiv.org/abs/1801.01973.
21. Borji A. Pros and cons of GAN evaluation measures [Electronic resource] / Ali Borji // Computer Vision and Image Understanding. – 2019. – Vol. 179. – P. 41–65. – Access mode: https://doi.org/10.1016/j.cviu.2018.10.009.
22. Differentiable Augmentation for Data-Efficient GAN Training [Electronic resource] / [S. Zhao, Z. Liu, J. Lin et al.] // arXiv.org. – 2020. – Access mode: https://arxiv.org/abs/2006.10738.
23. Regularized Evolution for Image Classifier Architecture Search [Electronic resource] / [Esteban Real et al.] // Proceedings of the AAAI Conference on Artificial Intelligence. – 2019. – Vol. 33. – P. 4780–4789. – Access mode: https://doi.org/10.1609/aaai.v33i01.33014780.
24. Cytological and histological images of breast cancer. (n.d.) [Electronic resource] // Zenodo. – 2023. – Access mode: https://doi.org/10.5281/zenodo.7890874.
25. Liashchynskyi P. Rudi – Python library [Electronic resource] / Petro Liashchynskyi // GitHub. – 2018. – Access mode: https://github.com/liashchynskyi/rudi.
26. Ye J. Geometric GAN [Electronic resource] / J. Ye, J. Lim // arXiv.org. – 2017. – Access mode: https://arxiv.org/abs/1705.02894.
27. Kingma D. Adam: A Method for Stochastic Optimization [Electronic resource] / D. Kingma, J. Ba // arXiv.org. – 2017. – Access mode: https://arxiv.org/abs/1412.6980.
28. Bynagari N. B. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium [Electronic resource] / Naresh Babu Bynagari // Asian Journal of Applied Science and Engineering. – 2019. – Vol. 8, № 1. – P. 25–34. – Access mode: https://doi.org/10.18034/ajase.v8i1.9.
29. Miyato T. Spectral Normalization for Generative Adversarial Networks [Electronic resource] / T. Miyato, T. Kataoka, M. Koyama // arXiv.org. – 2018. – Access mode: https://arxiv.org/abs/1802.05957.
30. Berezsky O. M. Comparison of generative adversarial networks architectures for biomedical images synthesis [Electronic resource] / Oleh M. Berezsky, Petro B. Liashchynskyi // Applied Aspects of Information Technology. – 2021. – Vol. 4, № 3. – P. 250–260. – Access mode: https://doi.org/10.15276/aait.03.2021.4.
31. Design of computer systems for biomedical image analysis / [O. M. Berezsky, K. M. Berezska, G. M. Melnyk et al.] // Proceedings of the X th International Conference «The Experience of Designing and Application of CAD Systems in Microelectronics» CADSM 2009. – Lviv : Publishing House Vezha&CoC. – P. 186–191.
32. Berezsky O. The intelligent system for diagnosing breast cancers based on image analysis [Electronic resource] / Oleh Berezsky, Serhiy Verbovyy, Tamara Datsko // 2015 Information Technologies in Innovation Business Conference (ITIB), Kharkiv, Ukraine, 7–9 October 2015. – 2015. – Access mode: https://doi.org/10.1109/itib.2015.7355067.
33. Fuzzy system diagnosing of precancerous and cancerous conditions of the breast [Electronic resource] / [Oleh Berezsky et al.] // 2016 XIth International Scientific and Technical Conference “Computer Sciences and Information Technologies (CSIT), Lviv, Ukraine, 6–10 September 2016. – 2016. – Access mode: https://doi.org/10.1109/stc-csit.2016.7589906.
34. Berezsky O. Hybrid Intelligent Information Technology for Bio-medical Image Processing [Electronic resource] / O. Berezsky, S. Verbovyy, O. Pitsun // IEEE 13th International Scientific and Technical Conference on Computer Sciences and Information Technologies. – 2018. – Access mode: https://ieeexplore.ieee.org/document/8526711.
35. Fuzzy System For Breast Disease Diagnosing Based On Image Analysis [Electronic resource] / [O. Berezsky, L. Dubchak, T. Batryn et al.] // Proceedings of the II International Workshop Informatics & Data-Driven Medicine (IDDM 2019). – 2019. – Access mode: http://ceur-ws.org/Vol-2488/paper6.pdf.

DOI 10.15588/1607-3274-2024-1-10