INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 17, Issue: 5, Month: October, Year: 2022
Article Number: 4886, https://doi.org/10.15837/ijccc.2022.5.4886

Evolutionary Computation Paradigm to Determine Deep Neural Networks Architectures

R.C. Ivanescu, S. Belciug, A. Nascu, M.S. Serbanescu, D.G. Iliescu

Renato Constantin Ivanescu
Department of Computers and Information Technologies, University of Craiova, Romania
A.I. Cuza, No 13, 200585, Craiova, Romania
constantin.ivanescu@edu.ucv.ro

Smaranda Belciug
Department of Computer Science, University of Craiova, Romania
A.I. Cuza, No 13, 200585, Craiova, Romania
*Corresponding author: sbelciug@inf.ucv.ro

Andrei Nascu
Department of Computer Science, University of Craiova, Romania
A.I. Cuza, No 13, 200585, Craiova, Romania
andreinascu3@gmail.com

Mircea-Sebastian Serbanescu
1. Department of Computer Science, University of Craiova, Romania
A.I. Cuza, No 13, 200585, Craiova, Romania
2. Department No 2, University of Medicine and Pharmacy, Craiova
Petru Rares, No 2, 200349, Craiova, Romania
mircea_serbanescu@yahoo.com

Dominic Gabriel Iliescu
1. Department of Computer Science, University of Craiova, Romania
A.I. Cuza, No 13, 200585, Craiova, Romania
2. Department No 2, University of Medicine and Pharmacy, Craiova
Petru Rares, No 2, 200349, Craiova, Romania
dominic.iliescu@umfcv.ro

Abstract

Image classification is usually done using deep learning algorithms. Deep learning architectures are set deterministically. The aim of this paper is to propose an evolutionary computation paradigm that optimises a deep learning neural network's architecture. A set of chromosomes are randomly generated, after which selection, recombination, and mutation are applied. At each generation the fittest chromosomes are kept. The best chromosome from the last generation determines the deep learning architecture. We have tested our method on a second trimester fetal morphology database. The proposed model is statistically compared with DenseNet201 and ResNet50, proving its competitiveness.

Keywords: deep learning, evolutionary computation, statistical analysis, image classification, fetal morphology

1 Introduction

The first cause of fetal death, infant morbidity and mortality is represented by Congenital Anomalies (CAs), [1]. Within the first 28 days after birth, more than 295 000 new-borns die because of CAs, placing Romania on top with one of the highest death rates in EU, (https://data.unicef.org/country/rou - accessed July 1, 2022). According to the EUROCAT (European Union Congenital Anomalies) report [2], each year 2.5% births within the EU will have CAs, while worldwide the numbers go up to 6%, that is 7.9 million infants, [3]. Even if some CAs can be treated, still 3.2 million children are disabled for life. Detecting CAs at an early stage of pregnancy facilitates life-saving treatments and even stop the progression of disabilities. Discovering CAs prenatally improves neonatal care for both term and preterm infants, [4].

CAs are diagnosed prenatally while performing the second trimester morphology scan (MF). Through MS the structure and functionality of the organs are evaluated. MS are not easy to interpret, sonographers been far from a correct detection of CAs. An experienced sonographer (more than 2000 MS examinations) has the detection rate of only 52%, while a sonographer with less experience has a detection rate of 32.5%, [5].

A study that regards the pre- and post-natal discrepancy of CAs performed manually reported a sensitivity that ranged from 27.5% to 96%, [6]. Multiple factors influence the performance of the sonographer, from which we mention: experience, time pressure, fatigue, fetal unintentional movement, and mother's physical features.

An *Advanced Intelligent Decision Support System* (AIDSS) has the capacity to gather, analyse data, communicate with other systems, learn from experience and adapt according to new cases. Our aim is to foster a cross-fertilization of MS, image processing and Artificial Intelligence (AI), which will result in a AIDSS that will assist medical practice and discovery. Relatively little work has been published in this field. Fujitsu, Cancer Translational Research Team, and the Department of Obstetrics and Gynaecology Showa University School of Medicine used deep learning to signal CAs in the fetal heart, [7] [8]. A fully convolutional neural network for the segmentation of the 3D fetal brain was proposed by Namburete et al., [9]. In [10], deep learning together with sequential forward feature selection techniques and support vector machines was used to obtain a segmentation of the fetal lungs and brain from magnetic resonance imaging and ultrasounds. In [11] deep learning was used to segment the fetal head. In [12], a differential evolution deep neural network was used to determine the view planes from a MS.

Finding the best architecture of a deep neural network has become a quest in the last decade. In 2019, 300 studies have been published in this area, [13]. One way to optimize the network's architecture is through neuroevolution, [14]. Neuroevolution proved to have similar performance as the gradient descent algorithm when applied on the loss function in the case of Gaussian white noise, [15]. Obviously, this view is not shared by everyone. In [16], it is mentioned that one generation of neuroevolution is not sufficient for comparison with the gradient descent.

Neuroevolution has been applied to the evolution of deep encoder, [17]. In other cases, we can use neuroevolution to estimate the performances of different deep neural networks, [18]. In this study, the authors succeeded in reducing the computation time of the training phase from 33 GPU days to 10 GPU days. The performance of the deep neural network remained unchanged. Xie and Yuille use a genetic algorithm to evolve the topology, hyperparameters, and synaptic weights of the convolutional neural network, [19].

Another approach of neuroevolution is the AmoebaNet-A, which has been obtained through a modified evolutionary computation algorithm. The authors modified the tournament selection using the age of chromosomes to favour the offspring over the parents, [20].

In this paper, we are interested in automatically determining the best architecture of a deep neural network for classifying the views of the anatomical structures from the fetal abdomen. We propose to use evolutionary computation to determine the best architecture of a deep neural network for this task. For benchmarking we have compared the novel model's results with the results obtained by two other state-of-the-art deep learning networks, ResNet50 and DenseNet201.

The paper is organized as follows: section 2 the design and implementation of our approach. In section 3 we present the dataset, the manner in which we have designed the experiments, and the parameter setting. In section 4 we present the results and their discussion. The paper ends with the conclusions in section 5.

2 Model design

2.1 Deep learning neural networks' architectures

Deep learning (DL) neural networks are a special class of neural networks, [21]. Unlike the classical neural networks, [22], where we encounter only one type of hidden layers, in a DL network we have three such types: convolutional, pooling, and fully connected layers. In the convolutional layers we have kernels or filters that scan the input images and produce a feature map through convolutions. After several convolution layers, it is customary to add a pooling layer that down samples the feature map. In this manner in the DL spatial invariance is produced. The fully connected layer resembles to the layer from the classical neural networks, where the input of the layer is connected to all the neurons. The architecture of a DL ends with a fully connected layer.

The most commonly used activation function in a DL is the rectified linear unit (ReLU):

$$f(x) = \{0, x < 0 \ x, x \geqslant 0\}.$$

Other activation functions that can be used are the sigmoid:

$$f(x) = \frac{1}{1+exp(-a \cdot x)},$$

or the hyperbolic function:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}.$$

The fully connected layer has as activation function the softmax, that is the generalized logistic regression. Softmax takes a score from R^n and output a vector that contains probabilities, $p = (p_1..p_n)$, where $p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$.

State-of-the-art DL have their architectures set deterministically, which makes them rigid. Optimizing their architecture through neuroevolution might be the key to this problem.

2.2 Evolutionary computation approach

Evolutionary computation (EC) is a perfect tool for optimization problems. The algorithms are modelled after the biological processes encountered in life. These types of algorithms consist of a population of chromosomes, different selection mechanisms that are based on a fitness function, a reproduction operator that produces new offspring, and a mutation mechanism applied onto the offspring, [23] [24]. Initially, the population of chromosomes is randomly initialised. These chromosomes cover all the intel about the potential candidate solution for the optimization problem, whether it concerns minimization or maximization. At each iteration, considering the evaluation of the cost function we choose the best chromosomes to form the next generation. The reproduction operator is used to restock the population, whereas the mutation operator is used for variation, technically speaking escaping from the local minima or maxima. At each iteration all the chromosomes are evaluated using the fitness function. The algorithm stops if the a given criterion is met.

The choice of the criterion depends on the user on multiple facts such as, [25]:

- Does the fitness function improve as generations pass or not ?
- Do we have a predetermined threshold for the population diversity ?
- Do we have a predetermined number of generations ?
- Do we have a predetermined number of fitness evaluations ?

The fitness function $f(x_i), i = 1, 2, ..., n$ measures the skill of each chromosome to compete against the rest of the population in a given environment. The fitness function is nonnegative.

EC uses the chromosomes to explore in parallel different potential solutions for the problem at hand. Let us presume that the population C contains N chromosomes. Each chromosome contains M genes/features and is mathematically written as ${X_i}^C = ({X_{i1}}^C, {X_{i2}}^C, ..., {X_{iM}}^C)$, where *i=1, 2, . . . ,N*. We initialize the population randomly from the upper (${X_{i,n}}^U$) and lower bound (${X_{i,n}}^L$) of the search interval for each gene of the chromosome.

$$X_{i,n} = {X_{i,n}}^L + rand() \cdot ({X_{i,n}}^U - {X_{i,n}}^L), i = 1, 2, ..., M, n = 1, 2, ...N.$$

The general scheme for an EC algorithm is [26] [27]:
1. Randomly initialize the chromosome population
2. Obtain the fitness score of each chromosome
3. Repeat the following steps until the stopping criterion is reached:
    2.1 Apply the selection operator to choose the parents
    2.2 Apply the crossover operator to produce the offspring
    2.3 Apply the mutation operator on the offspring to produce variety
    2.4 Apply the fitness function on the population
    2.5 Select according to the fitness the chromosomes that will form the next generation

2.3 Our idea

In what follows the EC/DL algorithm for optimizing the architecture of a DL is presented. The first step in the optimization process is to determine the genes of a chromosome, that will in the end determine the best DL architecture. A chromosome is encoded in a fixed-length integer vector. For us to build a performant DL architecture, we needed to determine the number γ of convolutional hidden layers and the number of neurons in each convolutional layers $nH_i, i = 1, ..., \gamma$. After each convolution, we added a max pooling layer of 2 by 2. The depth of the kernel matched the color channel number 3 (red, green, blue). We have set the recombination probability, p_r, to 0.6, and the mutation probability, p_m to 0.25. A potential solution will be an integer vector of the following form: $x_i = (\gamma, nH_j, j = 1, ..., \gamma, activationfunction, optimizer), i = 1, 2, ..., N$. All the potential architecture ended with a dense layer.

Briefly, we summarize the EC/DL algorithm:

1. Input: the image dataset, the number of generations nG, the number of chromosomes in a generation, N, and the chromosomes $x_i = (\gamma, nH_j, j = 1, ..., \gamma, activationfunction, optimizer), i = 1, 2, ..., N$.

2. Create the initial population: generate randomly the chromosomes. Use each chromosome i to build a DL having γ_i convolutional and pooling layers, nH_i, one activation function chosen randomly from the three mentioned before, and one optimizer. Train the DLs and record the chromosome's fitness score as the accuracy obtained on the validation dataset of the DL having the respective architecture.

3. Selection. Select the parents for the recombination process

4. Recombination. Apply the recombination operator.

5. Mutation. Apply the mutation operator.

6. Selection. Select the chromosomes that will form the next generation.

7. Repeat steps 3-6 until the stopping criterion is met (the predetermined number of generations nG has been reached).

8. Output. Return the best performing chromosome from the last generation. This chromosome will represent the best DL architecture for the problem at hand.

3 Dataset and design of experiments

3.1 Dataset

This is a prospective cohort study which is deployed in a tertiary maternity hospital (University Emergency County Hospital Craiova, Romania). The data analysis and processing was performed at the Department of Computer Science, University of Craiova, Romania. All the eligible participants were pregnant women admitted at the Prenatal Unit for the second trimester morphology scan. The patients have been recruited consecutively.

During the standard consultation, the doctor provided information about the conducted research to all eligible patients and invited them to take part of the study. All patients that have agreed and fulfilled the inclusion criteria gave a written informed consent. An extremely important issue that needs to be addressed is privacy-preserving and explainable AI for parents with children that had been diagnosed with CAs [28]. The doctors make sure that the patients understand the study's implication.

The ultrasounds had been performed by obstetricians with minimum 2 years experience, and with training in transabdominal obstetrical ultrasound. All images were acquired using Voluson 730 Pro (GE Medical Systems, Zipf, Austria) and Logic e (GE Healthcare, China US machines with 2-5-MHz, 4-8-MHz, and 5-9-MHz curvilinear transducers.

When this study has been performed, the dataset contained 970 images from 100 participants. The dataset regards the abdominal plane and has 10 classes: 3 vessels plus bladder (113 images), gallbladder (70 images), sagittal cord insertion (47 images), transverse cord insertion (103 images), anterior abdominal wall (20 images), anteroposterior kidney plane (96 images), biometry plane (205 images), intestinal sagittal plane (83 images), kidney sagittal plane (207 images), bladder plane (22 images).

Figure 1 presents a sample image from every decision class.

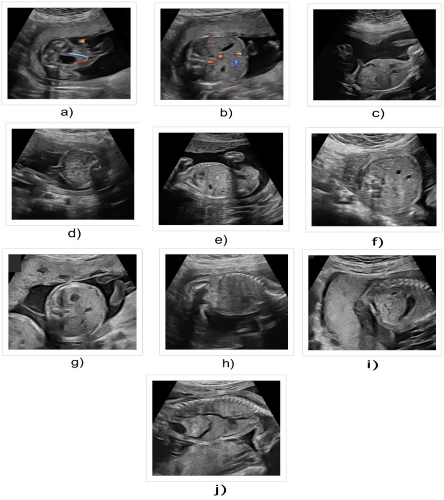

Figure 1: Fetal morphology abdomen: (a) 3 vessels plus bladder; (b) gallbladder; (c) sagittal cord insertion; (d) transverse cord insertion; (e) anterior abdominal wall; (f) anteroposterior kidney plane; (g) biometry plane; (h) intestinal sagittal plane; (i) kidney sagittal plane; (j) bladder plane.

It can be seen from the distribution of the images in the decision classes, that the dataset is unbalanced. This situation is encountered in real clinical scenarios. The images were anonymized and pre-processed to eliminate the text. For the pre-processing we have used CV2 and Keras-OCR. At first the text was detected from the images using Optical Character Recognition (OCR), followed by fill the missing parts using inpainting. In some special cases, the algorithm produced false positives (detecting text when it was not the case), hence after the automatically process of removing the text, we have rechecked manually every image to verify whether there are false positives or not. When a text had been identified a bounding box contained the coordinates was obtained. A mask was applied to each box, and the algorithm started to inpaint it, in order to produce a whole image.

Because DL algorithm need a larger dataset, we have applied augmentation. Hence, we have applied minor changes to our data: flips, rotations, translations, and adding Gaussian noise. The purpose for these changes is to trick the DL into thinking it is presented with multiple distinct images. After the augmentation process, the dataset used for testing our novel approach contained 4815 images: 3 vessels plus bladder (560 images), gallbladder (350 images), sagittal cord insertion (235 images), transverse cord insertion (510 images), anterior abdominal wall (105 images), anteroposterior kidney plane (480 images), biometry plane (1020 images), intestinal sagittal plane (420 images), kidney sagittal plane (1030 images), bladder plane (105 images).

3.2 Design of experiments. Parameters settings.

An important part of this study was the benchmarking process for comparing our novel approach's performance with the performances of two other state-of-the-art DL algorithms DenseNet201 and ResNet50. The validation method chose for each algorithm was the 10-fold cross validation. Each model was executed in 100 independent runs of a complete cross-validation cycle. The statistical power achieved was greater than 95% having a type I error α=0.05 for every test applied. For each model we have computed the average validation accuracy (AVA), standard deviation (SD), precision, recall and F1-score. We have used Precision Recall AUC instead of the AUROC, because the dataset was unbalanced, hence the probability of producing a large false positive rate was high.

Before applying statistical comparison tests such as one-way ANOVA together with post-hoc Tukey HSD, we have verified the normality and the equality of variances distributions of the samples that contained the 100 computer runs for each model. As normality test we have applied the Shapiro-Wilk W test, because it has a greater power at detecting non-normality than the Kolmogorov-Smirnov and Lilliefors test, and as equality of variance test we have applied the Levene's test.

Regarding the EC/DL model, the algorithm chose as activation functions between the sigmoid, hyperbolic tangent, and ReLU. The softmax activation function was used between the dense layer and the output layer. For the optimizer the choice was between Adam, AdamX, and NAdam.

In our approach, we have used as recombination operator the uniform crossover adapted for the integer representation. Technically, we have generated a vector of probabilities having the same length as the chromosomes in our population. The values in this vector are uniformly generated. The offspring are created using a threshold of 0.5. The first offspring is created as follows: if the value of the probability vector is below the threshold, then we copy the gene from the first parent, otherwise the gene from the second parent. For the second offspring the situation is vice-versa. The mutation was uniform having as mutation probability $p_m = 0.25$.

The initial chromosome population contained 30 candidate solutions. We have set the number of generations to 50. The chromosomes were generated from the intervals [1, 5] and [1, 256]. The stride was set to 1. The algorithm ran for 10 training epochs having a batch size of 64.

The results that we have obtained are presented in the Results section.

4 Results

In table 1 we present the results obtained when applying EC/DL on our image dataset in terms of average accuracy over 100 computer runs (AVA), SD, Precision, Recall, and F1-score. We have also added the architecture chosen, the one that gave the best results.

Table 1: Performance metrics for EC/DL

AVA (%)	SD	Precision	Recall	F1-score	Architecture
74.63	3.67	0.744	0.739	0.742	4 convolutional layers Layer 1: 34 neurons Layer 2: 68 neurons Layer 3: 134 neurons Layer 4: 240 neurons Activation function: ReLU Optimizer: Adam

We can see from table 1 that the EC/DL had obtained an average accuracy of 74.63%, having a fairly small SD, 3.67, which proves that the model is robust and stable. The values of the precision, recall and F1-score show that the model performs good.

We were interested in seeing how our novel approach behaves in comparison to other state-of-the-art DL algorithm. For this task we have selected the DenseNet201 and ResNet50. A DenseNet is more condensed than other DLs, because each layer from the network received collective knowledge from the previous layers. DenseNet reprocesses features, instead of having a deep architecture, by concatenating the feature map of one layer with the feature map of the next layer, and not by summing the outputs. ResNet on the other hand uses a skip connection, which technically means that it creates a substitute route for the gradient to pass through the network. The model learns an identity function making lower layers perform exactly like higher ones.

Table 2 presents the average accuracy over 100 computer runs obtained by each DL model.

Table 2: Average accuracy of EC/DL and other DLs

DL model	AVA (%)
EC/DL	**74.63**
DenseNet201	70.24
ResNet50	69.11

The data screening process's results, regarding the Shapiro-Wilk W test for all models are presented in table 3. We can see that the test establishes that the data samples are not governed by the normal distribution. Nevertheless, the *Central Limit Theorem* states that if the sample size is large enough ($>$ 30) then the sample is approximately normal [29]. Since our sample sizes surpass 30, we can presume that the samples are indeed approximately normal.

Table 3: Normality testing. Shapiro Wilk W test

Model	W statistic	p-level	Skewness	Skewness shape	Excess kurtosis	Kurtosis shape
EC/DL	0.8229	0.0000	-1.008	Asymmetrical left/negative	-0.31	Mesokurtic
DenseNet201	0.6104	0.0000	1.1877	Asymmetrical right/positive	-0.409	Mesokurtic
ResNet50	0.672	0.0000	0.762	Asymmetrical right/positive	-1.280	Platykurtic

Next, before applying one-way ANOVA, we had to verify assumption referring to the equality of variances. The results of the Levene's test are displayed in table 4. We can see that the test reveals that the EC/DL vs. DenseNet201 and EC/DL vs. ResNet50 have different variances, but in practice, if we have the same number of observations, we can presume that in fact the variances are approximately equal.

Table 4: Equality of variances

Variance	Degrees of freedom	Effect size	F statistic	p-value
EC/DL vs. DenseNet201	1	0.43	72.338	**0.0000**
EC/DL vs. ResNet50	1	0.51	103.314	**0.0000**
DenseNet201 vs. ResNet50	1	0.064	1.5856	0.2087

After having covered all the assumptions, we proceeded with applying the one-way ANOVA together with the post-hoc Tukey HSD. The results in terms of degrees of freedom (df), sum of squares (SS), mean sum of squares (MS), F-value, and p-level are displayed in table 5 [30] [31].

Table 5: One-way ANOVA results

SS	df	MS	F-value	p-level
768625	2	38431255	44.781	0.0000

Please notice, that even if the mean accuracies of the three models are fairly close, the one-way ANOVA shows that there are statistically significant differences between them. To see between which models there are statistical differences in terms of accuracies, we have applied Tukey's post-hoc test. The results presented in table 6 show that there indeed exist statistically significant differences between our novel proposed model and the two state-of-the art DLs (p-level < 0.05), proving beyond any doubt that the method improves the performance.

Table 6: Tukey's post-hoc test results

Pair	p-level
EC/DL vs. DenseNet201	0.0000
EC/DL vs. ResNet50	0.0000
DenseNet201 vs. ResNet50	0.2696

5 Discussion

The aim of this study was to find a way to optimize a DLs architecture through neuroevolution. We have designed an EC algorithm that computes the best architecture of DL depending on dataset used. After the EC optimized the architecture, we have applied the new EC/DL on a second trimester morphology that regards the abdomen dataset. At first, we have pre-processed the data by eliminating the text from the ultrasound images. Because the dataset was not large enough, we have augmented the data.

Our EC/DL model was compared with two state-of-the-art DL algorithms, DenseNet201 and ResNet50. The statistical analysis performed showed that there are significant differences in performances between the three models, even if the accuracies might seem close enough. The statistical analysis included the power analysis, average accuracy over 100 computer runs, SD, precision, recall, F1-score, Shapiro-Wilk W, Levene, one-way ANOVA and Tukey's post-hoc test. The results show that there are differences in performances, our optimized DL outperforming the rest.

6 Conclusions

In this study we used evolutionary computation to optimize the architecture of a deep neural network. We have proposed a way to encode the architecture of a DL in a chromosome, after which we have applied different evolutionary operators (selection, recombination, and mutation) to explore the search space efficiently. We have built a model that had as architecture the best selected chromosome and applied it on a fetal morphology dataset. Before applying it, we pre-processed the dataset by eliminating the text from the ultrasounds and by augmenting it. The results were statistically analysed and benchmarked in comparison with the results obtained by DenseNet201 and ResNet50. Our findings show that indeed neuroevolution can optimize a DLs architecture, because the resulting EC/DL outperformed classical DL algorithms.

Conflict of interest

The authors declare that there is no conflict of interest.

Acknowledgements

This work was supported by a grant of the Ministry of Research Innovation and Digitization, CNCS-UEFISCDI, project number PN-III-P4-PCE-2021-0057, within PNCDI III.

References

[1] Boyle, B., et al., (2018). Estimating Global Burden of Disears due to congenital anomaly: an analysis of European Data, *Archives of Disease in Childhood – Fetal and neonatal edition*, 103, F22-F28, 2018

[2] Kinsner-Ovaskainen, A., et al., (2018). European Monitoring of Congenital Anomalies: JRC EUROCAT Report on Statistical Monitoring of Congenital Anomalies (2008-2017), EUR 30158 EN, *Publications Office of the European Union, Luxembourg*, doi: 10.2760/65886, 2018.

[3] Lobo, I., Zhaurova, K., (2008). Birth defects: causes and statistics, *Nature Education,* 1 (1), 18, 2008.

[4] AlQaheri, H., et al., (2021). Toward an autonomous incubation system for monitoring premature infants. *Studies in Informatics and Control,* 30 (4), 121-131, https://doi.org/10.23846/v30i4y202111, 2021.

[5] Tegnander, E., Eik-Nes, S.H., (2006). The examiner's ultrasound experience has a significant impact on the detection rate of congenital heart defect at the second trimester fetal examination. *Ultrasound Obstet Gyncol,* 28, 8-14, 2006.

[6] Salomon, L., et al., (2008). A score-based method for quality control of fetal images at routine second trimester ultrasound examination. *Prenat Diagn,* 28 (9), 822-827, 2008.

[7] Matsuoka, R., Komatsu, M., et al., (2019). A novel deep learning based system for fetal cardiac screening. *Ultrasound Obstet Gyn,* https://doi.org/10.1002/uog.20945, 2019

[8] Komatsu, R., Matsuoka, R., et al., (2019). Novel AI-guided ultrasound screening system for fetal heart can demonstrate findings in timeline diagram. *Ultrasound Obstet Gyn,* http://doi.org/10.1002/uog.20796,2019.

[9] Namburete, A. et al., (2018). Fully automated alignment of 3D fetal brain ultrasound to a canonical reference space using multi-task learning. *Med Image Anal.,* 46, 1-14, 2018.

[10] Torrents-Barrena, J. et al, (2019). Assessment of radiomics and deep learning for the segmenation of fetal and maternal anatomy in magnetic resonance imaging and ultrasound. *Acad. Radiol.,* S1076-6332(19)30575-6, 2019.

[11] Al-Bander, B. et al., (2020). Improving fetal head countour detection by object localization with deep learning. *Annual Conference on MedicalImage Understanding and Analysis, Springer,* 142-150, 2020.

[12] Belciug, S., (2022). Learning deep neural networks' architectures using differential evolution. Case study: medical imaging processing, *Computers in Biology and Medicine,* 146, 105623, 2022.

[13] Lindauer, M., Hutter, F., (2019). Best Practices for Scientific Research on Neural Architecture Search, arxiv.org/abs/1909.02453, 2019.

[14] Stanley, K.O. (2017). Neuroevolution: a Different Kind of Deep Learning, 2017.

[15] Whitelam, S., Selin V., Park, S-W., Tamblyn, I., (2021). Correspondence between neuroevolution and gradient descent. *Nat Commun,* 12, 6317 https://doi.org/10.1038/s41467-021-26568-2, 2021.

[16] Khadka, S. et al., (2019). Evolutionary reinforcement learning for sample-efficient multiagent coordination. *ECLR 2020,* 2019.

[17] Hajewski, J., Oliviera, S., Xing, X. (2020). Distributed evolution of deep autoencoders. arxiv.org/abs/2004.07607, 2020.

[18] Sun, Y., Wang, B., Xue, B, Jin, Y., Yen, G.G., Zhang, M., (2020). Surrogate-assisted evolutionary deep learning using an end-to-end random forest based performance predictor. *IEEE Trans Evol Comput,* 24 (2), 350-364, 2020.

[19] Xie, L., Yuille, A., (2017). Genetic CNN, *Computer vision and pattern recognition,* arxiv:1703.01513, 2017.

[20] Al-Oudat, M., et al., (2021). An Interactive automation for human biliary tree diagnosis using computer vision. *International Journal of Computers Communications & Control,* 16, 5, 4275, https://doi.org/10.15837/ijccc.2021.5.4275, 2021.

[21] Dumitrach, I et al., (2021). Neuro-inspired framework for cognitive manufacturing control, *International Journal of Computers Communications & Control,* 16, 6, 4519, https://doi.org/10.15837/ijccc.2021.6.4519, 2021.

[22] Miikkulainen, R., Liang, J.Z. et al., (2017). Evolving Deep Neural Networks, *CoRR,* abs/1703.00548, 2017.

[23] Liu, X. et al. (2021). A method based on multiple population genetic algorithm to select hyperparameters of industrial intrusion detection classifier. *Studies in Informatics and Control,* 30 (3), 39-49, https://doi.org/1024846/v30i3y202104, 2021.

[24] Serban, C., Carp, D., (2021). Using genetic algorithm to solve discounted generalized transportation problem, *Studies in Informatics and Control,* 30 (3), 29-38, https://doi.org/10.24846/v30i3202103, 2021.

[25] Gorunescu, F. et al., (2005). An evolutionary computation approach to probabilistic neural networks with application to hepatic cancer diagnosis, *18th IEEE Symposium on computer based medical systems,* 461-466, 2005.

[26] Eiben, A.E., Smith, J.E., (2003). Introduction to Evolutionary computing, *Berlin, Springer-Verlag,* 2003.

[27] Haupt, R.L., Haupt, S.E. (2004). Practical genetic algorithms, 2nd ed, UK. John Wiley & Sons, 2004.

[28] Puiu, A., et al., (2021). Privacy-preserving and explainable AI for cardiovascular imaging, *Studies in Informatics and Control,* 30 (2), 21-32, 2021, https://doi.org/10.24846/v30i2y202102, 2021.

[29] Gorunescu, F. et al. (2010). A statistical framework for evaluating neural networks to predict recurrent events in breast cancer, *J Gen Sys,* 39 (5), 471-488, 2010.

[30] Demsar, J. (2006). Statistical comparisons of classifiers over multiple datasets, *Mach Learn Res,* 7, 1-30, (2006).

[31] Seltman, H., (2018), Experimental design and analysis.
https://stat.cmu/edu/hseltman/309/Book/Book.pdf, 2018.

COPE

Member since 2012

JM08090

This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

Cite this paper as:

Ivanescu, R.C.; Belciug, S.; Nascu, A.; Serbanescu, M.S.; Iliescu, D.G. (2022). Evolutionary Computation Paradigm to Determine Deep Neural Networks Architectures, *International Journal of Computers Communications* & *Control*, 17(5), 4886, 2022.
https://doi.org/10.15837/ijccc.2022.5.4886