Computers in Biology and Medicine 146 (2022) 105623

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Learning deep neural networks' architectures using differential evolution. Case study: Medical imaging processing

Smaranda Belciug

Department of Computer Science, Faculty of Sciences, University of Craiova, Craiova, 200585, Romania

ARTICLE INFO

Keywords:
Cancer MRI scan
Cancer histopathological image
Maternal-fetal ultrasound
Deep learning
Differential evolution
Statistical assessment

ABSTRACT

The COVID-19 pandemic has changed the way we practice medicine. Cancer patient and obstetric care landscapes have been distorted. Delaying cancer diagnosis or maternal-fetal monitoring increased the number of preventable deaths or pregnancy complications. One solution is using Artificial Intelligence to help the medical personnel establish the diagnosis in a faster and more accurate manner. Deep learning is the state-of-the-art solution for image classification. Researchers manually design the structure of fix deep learning neural networks structures and afterwards verify their performance. The goal of this paper is to propose a potential method for learning deep network architectures automatically. As the number of networks architectures increases exponentially with the number of convolutional layers in the network, we propose a differential evolution algorithm to traverse the search space. At first, we propose a way to encode the network structure as a candidate solution of fixed-length integer array, followed by the initialization of differential evolution method. A set of random individuals is generated, followed by mutation, recombination, and selection. At each generation the individuals with the poorest loss values are eliminated and replaced with more competitive individuals. The model has been tested on three cancer datasets containing MRI scans and histopathological images and two maternal-fetal screening ultrasound images. The novel proposed method has been compared and statistically benchmarked to four state-of-the-art deep learning networks: VGG16, ResNet50, Inception V3, and DenseNet169. The experimental results showed that the model is competitive to other state-of-the-art models, obtaining accuracies between 78.73% and 99.50% depending on the dataset it had been applied on.

1. Introduction

Since the outburst of the COVID-19 pandemic in 2020, cancer patient and obstetric care landscapes have been distorted. While hospitals got more and more crowded with COVID-19 patients, disturbances appeared through all aspects of cancer care from diagnostic to tailored or classical treatment [1–5], as well as maternal-fetal care [6–8]. Europe and North America experienced a lot of pressure on the healthcare system, and changes in the routine cancer and maternal-fetal care were necessitated. Even if the cancer care remained available, cancer screening programs were interrupted. Delaying diagnosis ultimately increased the number of preventable cancer deaths [9–11]. The onset of the fear and the anxiety of being infected with COVID-19 among individuals, prevented patients with potential non-specific symptoms of cancer to avoid consulting specialists [12].

Colonoscopy rates fell by 4.1%–75% [13], lung screening rates were reduced by 57%, 74%, and 56% respectively [13,14]. Cancer biopsies also recorded reductions, i.e colon (−33 to −79%), and lung (−47 to −58%). In the case of neuro-oncology patients, things seem even worse since the urgency in their care is changing at a much faster pace. So far, the impact on brain tumor patients of the COVID-19 pandemic is yet unknown [15]. In what regards maternal-fetal care in the COVID-19 pandemic, long-lasting congenital anomalies of infants have been observed, caused either by the actual infection, or by therapeutic maneuver (Khan et al., 2020). The number of caesarean sections has also increased as a secondary cause of the pandemic, [16]. The importance of a correct interpretation of the ultrasound is given by the fact that it allows a detailed discussion regarding the prognosis with parents (i.e. procedural risks, long-term mortality, morbidity, and, ultimately, quality of life). The current approaches have limitations. A study of the pre- and postnatal diagnosis discrepancy of congenital anomalies obtained by a manually interpreted ultrasound reported a performance sensitivity that ranged from 27.5% to 96%, [17]. The lack of necessary sonography know-how, fatigue, time pressure, fetal involuntary movement, and different characteristics of the patient, such as obesity might make it difficult, or in some cases, even impossible for a sonographer to

E-mail address: sbelciug@inf.ucv.ro.

https://doi.org/10.1016/j.compbiomed.2022.105623
Received 11 December 2021; Received in revised form 3 March 2022; Accepted 31 March 2022
Available online 17 May 2022

get a clear ultrasound image. Studies reported that obesity can lead up to 50% in misreading ultrasound in women with a body mass index over 30 kg/m^2 versus women with normal weight, [18].

It is a reality the fact that the COVID-19 pandemic has changed the face of medical practice. We must find means to support medical care at a faster rhythm. Early, fast, and accurate diagnosis from medical imagining can be achieved by employing Artificial Intelligence methods, such as deep learning (DL) neural networks (NNs). The COVID-19 pandemic has opened the path for a fast integration of deep learning algorithms in the healthcare system. The Food and Drug Administration has already granted the regulatory approval for select DL diagnostic software to be used in clinical practice [19,20]. This puts even more pressure in optimizing the design of the deep NNs, so that their applicability should improve the healthcare system [21]. In this paper we are interested in brain, lung, colon cancer, and maternal-fetal ultrasound classification using medical imagining.

Several research studies have applied DL for the classification of lung nodules into benign and malignant using CT scans. For instance, Kumar et al. used an autoencoder with deep features to classify lung cancer, obtaining an accuracy of 75.01% [22], while Sun et al. obtained an 81.19% accuracy using deep belief networks, [23]. Non-small cell lung cancer histopathology images were classified using a deep convolutional neural network (DCNN) achieving a 0.97 AUC, [24]. Another CNN achieved 84.15% accuracy, 83.96% sensitivity, and 84.32% specificity in classifying lung nodules on CT images, [25]. Other obtained results are a merger between Unet and Resnet obtained 84% accuracy [26], while a CNN applied on PET/CT lung images reached a 90% sensitivity [27].

Deep learning architectures achieved competitive results when applied on histopathological images of colon tumors. Two of such examples are: a shallow neural network that obtained an accuracy of 84% in classifying colon cancer [28], and a spatially constrained CNN merged with a neighboring ensemble predictor that obtained and AUC of 0.917, and F1 of 0.784 [29].

Regarding the classification of brain tumors, an input cascade CNN applied on MRI images obtained 0.84 dice, 0.88 specificity, and 0.84 sensitivity [30], while a multi-layer stacked denoising auto-encoder network obtained an average accuracy of 98.04% [31]. Other reported results include: a U-net which obtained around 0.88 sensitivity for high grade glioma, and 0.84 sensitivity for low grade glioma, [32]; a conditional generative adversarial network which obtained 0.68 dice, 0.99 sensitivity, and 0.98 specificity, [33]; a fully convolutional neural network which obtained 0.86 dice, [34]; a multi-view deep learning framework which obtained a 0.55 accuracy, [35]; and a deep wavelet autoencoder which obtained an average accuracy of 0.93, [36].

In [37], the authors applied different pretrained CNNs and two non-deep learning methods on two datasets regarding maternal-fetal ultrasounds and obtained accuracies ranging from 54% to 93.6%. Fujitsu started a research project with the Cancer Translational Research team and the Department of Obstetrics and Gynecology Showa University School of Medicine, in which they study fetal heart ultrasounds using deep learning, [38,39]. Namburete et al. proposed a fully CNN for the segmentation of the 3D fetal brain, [40]. A convolutional neural network was used for automated fetal cardiac assessment using 4D B-mode ultrasound, [41]. A segmentation of the fetal lungs and brains was obtained by using deep learning with sequential forward feature selection techniques and Support Vector Machines on magnetic resonance images (MRI) and ultrasounds, [42].

Finding the best architecture of the CNN for the problem at hand can be quite tricky. There is no perfect NN model that can be applied on every problem. This hypothesis was first introduced by Wolpert and Macready under the name of the ‘no-free-lunch-theorem’ [43]. All the NNs play the role of a certain ‘restaurant’ that provide us a ‘dish’, in our case a measure of performance, at a certain ‘price’ – the computational cost. Determining the ‘smart-deal’ takes a lot of time and effort. In recent years, the interest in automatically learning NN architectures has increased substantially. Three directions can be distinguished: reinforcement learning [44–48], progressive neural architecture search [49], and evolutionary computation [50,51]; Xie & Yuille, 2017). In reinforcement learning, the structure of the model is encoded as the sequence of actions the agent makes. The built model is afterwards trained and tested. The reward of the agent is computed as the obtained validation performance. In the progressive neural architecture search a sequential model-based optimization strategy is used. A surrogate model learns simultaneously to guide the search through the structure space. In evolutionary computation, the NN’s structure is represented as an array, which is subjected to random mutations and recombinations during the search process. Each model is trained and evaluated on the validation set. The top performing model is returned. All automated methods outperform manually tuning of the architectures. Ingenious architecture representations together with interesting methodologies have delivered astonishing results when compared to human designed networks, [51–53]. The downside is represented by the necessity of significant computational resources. Nevertheless, neuroevolution necessitates less computational time than reinforcement learning models, [54].

We propose the use of differential evolution in determining the best NN architecture. We have applied and tested this approach on three different cancer datasets. For comparison purposes we have compared our best performing models with state-of-the-art DL algorithms, such as VGG-16, ResNet50, Inception V3, and DenseNet169. A thorough statistical analysis is performed, to determine is the obtained results are robust and trustworthy.

The remaining part of the paper is organized in the following manner. Section 2 describes briefly the related work in the field, Section 3 presents the design and implementation of the novel model, while Section 4 summarizes the benchmarking datasets, the design of experiments and parameter settings. Section 5 details the experimental results obtained by the proposed model and other state-of-the-art DLs, followed by thorough statistical assessment them. Section 6 comprises the discussion. The paper ends with Section 7 that contains the conclusions.

2. Related work

The need for a fast and reliable diagnosis, led to the quest of finding the best architecture of CNNs for the problem at hand. Recent studies have proven that by automatically determining the network’s architecture we obtain far better results rather than by performing it manually. As we have state above, three directions are established: reinforcement learning, progressive neural architecture search, and evolutionary computation. By 2019, there were over 300 works published papers regarding NN architecture search, [55].

In [47], the authors proposed a neural architecture search method together with the algorithm named REINFORCE, first published in 1992, [56]. REINFORCE estimated the parameters of a recurrent neural network, parameters that represented the sequence of actions that the agent took. The authors used as reward for the agent the classification accuracy obtained by the new designed model on the validation data. The study has been extended in Ref. [48] through a more controlled search space by using stacked cells, and through the replacement of the REINFORCE algorithm with the proximal policy optimization algorithm, developed by Ref. [57]. In Ref. [46] the same neural architecture search method has been used, only the authors have replaced the policy gradient with the Q-learning method. The Q-learning algorithm was also deployed by Ref. [44] the difference between the studies being the lack of exploitation of the cell structure in the latter. In Ref. [45] the authors added an extra layer to the recurrent neural network trained through the policy gradient. In Ref. [58], the authors developed an evolutionary reinforcement learning scheme, which involved alternating physical and evolutionary dynamics, that ultimately led to building networks that were able to promote self-assembly of a certain structure at a faster and a better manner than other methods, such as intuitive cooling protocols. The newly developed networks were able to select between two polymorphs that were equal in energy and had been formed in unpredictable quantities under slow cooling protocols. No human input was needed, beyond the specification of which target parameter to promote.

[49] proposed a progressive neural architecture. The method implements a progressive scan of the neural architecture search space, choosing at each step the best performing ones. The networks' validation errors are collected and used to train a surrogate function which will predict the validation error of the succeeding architectures [59]. proposed a Pareto-optimal progressive neural architecture search that merges the architecture proposed by Ref. [49] with a time-accuracy Pareto optimization problem. Technically, a new time predictor is added in order to do a joint prediction of time and accuracy to each candidate architecture, searching over the Pareto front.

The area of neuroevolution uses evolutionary computation strategies to define the NNs' architecture [60]. Different types of evolutionary algorithms and stochastic gradient descent are used to learn the structure and/or the hyperparameters of the network [61]. combined a hierarchical genetic representation that models the design pattern used by human experts, and expressive search space for complex topologies. In Ref. [52] AmoebaNet-A image classifier has been evolved through the modification of the tournament selection of an evolutionary algorithm. The selection was modified by introducing an age property to favor the young genotypes [50]. proposed a new automated method, CoDeepNeet, that optimizes the DNN's architecture using the neuroevolution technique of NEAT, [62]. NEAT is used to evolve topologies, weights, and hyperparameters. (Xie & Yuille, 2017) deployed a genetic algorithm to optimize the CNN's architecture.

In another study, the authors developed a scalable evolutionary algorithm for NN architecture search [63]. They have applied their novel method to the evolution of deep encoders. In Ref. [64], meta-models with ensemble members can be used to estimate the accuracy of different CNNs. Their advantage consists in reducing the training time from 33 GPU days to 10 GPU days, gaining the same competitive results as other state-of-the-art techniques. A drawback of their approach is that they do not report the required number of training runs needed to reach that performance.

In [65] the authors show that neuroevolution performs the same as gradient descent on the loss function in the presence of Gaussian white noise. In this study numerical simulations were performed in order to illustrate the correspondence between the two methods which can be detected when applied to shallow and deep neural network. This connection between machine learning and statistical mechanics was also pointed out in Ref. [66]. The authors provide a review of recent works which show the associations between deep learning and different mathematical and physical methods such as random landscapes, jamming, dynamical phase transitions, chaos, spin glasses, Riemannian geometry, random matrix theory, nonequilibrium statistical mechanics, free probability. Contrary to the above-mentioned studies, authors such as Khadka et al. (Khadka et al., 2019), suggest that we should be careful when comparing neuroevolution methods to gradient descent, on generation of neuroevolution being not sufficient enough for such a comparison.

Thorough reviews of modern-day neuroevolution which present various significant features of the process, including large-scale computing, advantages of novelty and diversity, the power of indirect encoding, meta-learning and architecture search, together with future challenges can be studied in Refs. [67,68].

Different from the above methods, we propose the use of differential evolution for determining the architecture of CNNs. The obtained results of this method prove that it is competitive in terms of performance to other state-of-the-art CNNs.

3. The model

3.1. Convolutional neural Network's architecture

Convolutional Neural Networks (CNNs) are a specific type of NNs. They architecture usually consists of three types of layers: convolutional layer (CONV), pooling layer (POOL), and fully connected layer (FC). The CONV layer uses filters that perform convolution operations by scanning the input and producing a feature map. The CONV's parameters include the filter size and the stride. The POOL layer is applied after a convolutional layer and downsamples the feature map producing spatial invariance. The FC layer works on a flattened input, where each input is connected to all neurons. The FC is the last layer of the CNN's architecture.

In terms of hyperparameters in a CNN we encounter the size of the filter, the stride (i.e. number of pixel by which the window moves after each operation, and zero-padding (i.e. the process of boarding with zeros the input).

In a CNN we have as activation functions the rectified linear unit layer (ReLU), with its variants the Leaky ReLU, and Exponential linear unit, ELU, and the softmax classifier. ReLU, Leaky ReLU, or ELU are used on all elements of the volume. They induce non-linearities in the network, whereas softmax is the generalized logistic function that takes as input a score vector $y \in \mathbb{R}^n$ and outputs a probability vector $p \in \mathbb{R}^n$. The functions are defined as follows:

- ReLU:

$$f(x) = \begin{cases} 0, \ for \ x < 0 \\ x, \ for \ x \ \geq 0 \end{cases}.$$

- Leaky ReLU:

$$f(x) = \begin{cases} 0.01x, \ for \ x < 0 \\ x, \ for \ x \ \geq 0 \end{cases},$$

- ELU:

$$f(\alpha, x) = \begin{cases} \alpha(e^x - 1), \ for \ x < 0 \\ x, \ for \ x \geq 0 \end{cases},$$

- Softmax:

$$p = \begin{pmatrix} p_1 \\ . \\ . \\ . \\ p_n \end{pmatrix}, \text{where} p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

A CNN can be considered as a complex function that is trained using the back-propagating error signals computed as the difference between the ground truth and the prediction at the top layer. Designing a CNN's architecture is captivating. Some researchers argue that deeper CNNs obtain a higher accuracy in classification problems. Many networks have their structures set deterministic, even if stochastic processes are used to avoid over-fitting, [69,70]. Having deterministic structures limits the flexibility of the CNNs, hence we need to automatically learn the networks architecture.

3.2. Differential evolution

The inception of Differential Evolution (DE) appeared in 1997. This heuristic optimization algorithm is flexible, versatile, easy to implement and understand, [71,72]. DE mimics the natural biological evolution process. Technically, DE generates a temporary individual having as starting point the differences within populations, followed by an evolutionary restructuration of the population. Several studies proved its suitability for solving numerical optimization problems, having a good global convergence and robustness. DE has been applied fruitfully in constrained image classification [73], image segmentation [74], neural networks [75], linear array [76], global optimization problems [77], and other areas [78–81].

Different from other evolutionary computation algorithms, DE uses a population-based global search strategy. The complexity of the mutation operation of the differential is reduced by using one-on-one competition. By adapting the candidate solutions, DE explore in parallel different solutions. It enables dynamic track of the current search through its memory capacity, making possible the adjustments of the search strategy. Through this a global convergence and robustness is achieved.

Mathematically speaking, the population of each generation G contains N candidates. Each candidate can be written as $X_{iG} = (x_{iG}^{1}, x_{iG}^{2}, \ldots, x_{iG}^{M}),\ i = 1, 2, \ldots, N$, where M is the number of features.

The initial population of the candidate solutions is randomly generated between the upper and lower bound of the search interval for each feature.

$$X_i^n = X_i^{n,L} + rand()\cdot\left(X_i^{n,U} - X_i^{n,L}\right),\ = 1,\ 2, \ldots, M,\ n = 1,\ 2,\ \ldots N,$$

where $X_i^{n,L}$ is the lower bound of the variable X_i^n, and $X_i^{n,U}$ is the upper bound of the variable X_i^n.

For the mutation process to take place, we need to select three vectors $X_{r_1,G}$, $X_{r_2,G}$, $X_{r_3,G}$. The following formula is applied:

$$V_G^n = X_{r1,G}^n + F\cdot\left(X_{r2,G}^n - X_{r3,G}^n\right),$$

where V_{G+1}^n is the donor vector, $F \in [0, 1]$ is the variation factor that regulates the amplification degree of the differential variable. $X_{r2,G}^n - X_{r3,G}^n$.

Regarding the recombination process, the operator develops a trial vector $U_{i,G+1}^n$ from the target vector $X_{i,G}^n$ and the donor vector V_{G+1}^n, using the following formula:

$$U_{i,G+1}^n = \begin{cases} V_{i,G+1}^n,\ if\ rand() \gg C_p\ or\ i = I_{rand} \\ X_{i,G}^n,\ if\ rand() > C_p\ and\ i \neq I_{rand} \end{cases},$$

where $i = 1, 2, \ldots M$, $n = 1, 2, \ldots, N$, $I_{rand} \in [1, M]$ is an integer random number, and C_p is the recombination probability. The recombination strategy allows the old and the new candidate solution to exchange part of the code in order to form a new individual.

After the mutation and recombination processes are over, the selection process begins. The target vector $X_{i,G}^n$ is compared with the trial vector $U_{i,\ G+1}^n$. The vector that minimizes the fitness function values gets selected to be part of the next generation:

$$X_{i,G+1}^n = \begin{cases} U_{i,G+1}^n,\ if\ f\left(U_{i,G+1}^n\right) < f\left(X_{i,G}^n\right) \\ X_{i,G}^n,\quad otherwise \end{cases},$$

where $i = 1, 2, \ldots, M$, and $n = 1, 2, \ldots, N$.

The DE method's steps are the following:

1. Initialize candidate population.
2. Repeat:
   - 2.1. Mutation operation
   - 2.2. Recombination operation
   - 2.3. Selection operation

Until the stopping criterion is met.

3.3. Our approach

In this subsection we will present a DE/CNN algorithm for determining competitive CNN's architectures. At first, we define how to represent the network's architecture, the size of the filters in each convolutional layer, and the hyperparameters' values as a candidate solution using a fixed-length array, followed by several DE processes defined in subsection 3.2. The DE processes help us navigate through the search space in a more professional manner, which lead us into finding high-quality solutions for our problems.

3.3.1. Network's architecture representation

We define a population of candidate architectures which can be encoded in a fixed-length integer array. A CNN is composed of an input layer, λ convolutional hidden layers, π pooling layers, and an output layer. Each hidden layer has a certain number of hidden neurons, nH. The number of pooling layers is smaller than the number of hidden layers. Each filter has a width, fw, and a height, fh. The depth of the filter is not variable since it matches the number of color channels the image has (e.g. 2 for grayscale images, and 3 for RGB images). The hyperparameters are the recombination probability, Cp, and the mutation variation factor, F. Therefore, a candidate solution is an integer array $\mathbf{x}_i = (\lambda,\ nH, fw, fh, Cp, F),\ i = 1, \ldots, q$, where q is the number of candidate solutions in the population. Because all the candidate solution must be of a fixed length to apply mutation, recombination, and selection, we decided that each hidden layer in a candidate solution contains the same number of hidden units. After each convolutional layer, we added in the network a max pooling layer, except for the last one which is a dense layer.

Our study has a limitation: we have applied DE to determine only the number of convolutional layers, their units, the filter's height and depth, and the recombination probability and mutation variation factor. In future studies we shall find a way to encode the candidate solution using different number of hidden units in each convolution. However, our experiments and statistical analysis prove that the proposed model can achieve competitive performance, using DE to tune only these features. Our method can be scaled up if results are unsatisfying.

3.3.2. DE/CNN algorithm

The ReLU function was chosen as the non-linear activation function for each convolutional layer. The softmax function was chosen as activation function between the last dense layer and the output layer. The pool size was (2, 2).

1. **Input:** the image dataset D, the number of generations G, the number of candidate solutions in each generation N, $\mathbf{X}_i = (\lambda_i,\ nH_i, fw_i, fh_i, Cp_i, F_i)\ i = 1,\ 2, \ldots, N$ the candidate solutions.
2. **Initialization**: Randomly generate a set of candidate solutions $X_{i,1}$, $i = 1, 2, \ldots, N$, and built N CNNs having λ_i number of convolutional layers and pooling layer, nH_i number of hidden units per convolutional layer, the filter size (fw_i, fh_i), and the recombination probability Cp_i, and mutation variation factor F_i. Train the CNNs and record their accuracies and losses over the validation dataset. Each CNN's loss will represent the candidate solution's fitness value.

   - 3.1. **Mutation**: for each individual perform mutation using the variation factor F_i;
   - 3.2. **Recombination**: for each pair of individuals perform recombination with Cp_i;
   - 3.3. **Select**: the individuals that will for the next generation based on their validation loss.

4. Repeat

until stopping criterion is met (number G of generations is reached)

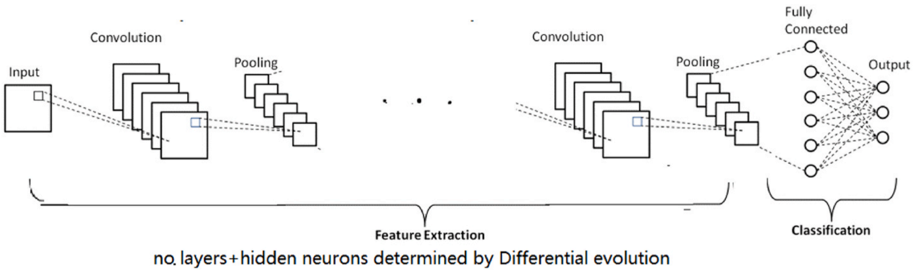


Fig. 1. DE/CNN architecture.

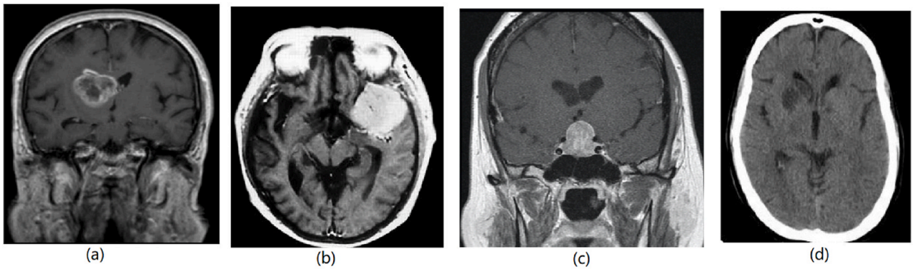


Fig. 2. (a) glioma tumor, (b) meningioma tumor, (c) pituitary tumor, (d) no tumor. (https://doi.org/10.34740/kaggle/dsv/1183165), [82].

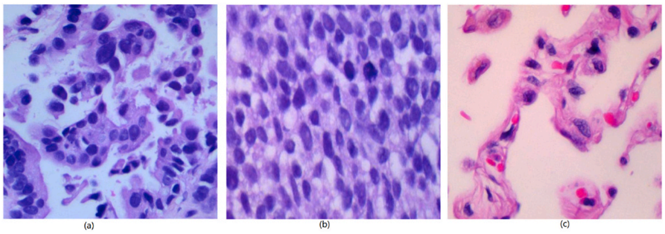


Fig. 3. (a) adenocarcinoma, (b) squamos cell carcinoma, (c) benign tissue (https://arxiv.org/abs/1912.12142v1, https://github.com/tampapath/lung_colon_image_set) [83].

4. **Output**: the best candidate solution that will represent the networks' architecture

The DE/CNN architecture is presented in Fig. 1.

5. Application case studies: lung, colon, brain tumor images, and maternal-fetal ultrasound planes

The novel proposed method has been applied on two publicly available cancer datasets that regard lung and colon cancer histopathological images, brain cancer MRI images, and two maternal-fetal ultrasound images. In what follows we shall briefly describe the datasets.

5.1. Datasets

Brain Cancer Dataset (Bc) (https://doi.org/10.34740/kaggle/dsv/1183165). The data is split into Training and Testing. The training set has four decision classes: 826 cases of glioma tumor, 822 cases of meningioma tumor, 827 cases of pituitary tumor, and 395 cases with no tumor. The testing set has four decision classes: 100 cases of glioma tumor, 115 cases of meningioma tumor, 74 cases of pituitary tumor, and 105 cases with no tumor [82]. Brain tumors are very complex, presenting abnormalities in terms of location and size. The type of tumor (glioma, meningioma, pituitary) determines the course of treatment and patient prognosis. We have preprocessed and resized them at 250 × 250. Fig. 2 presents a sample image of each class.

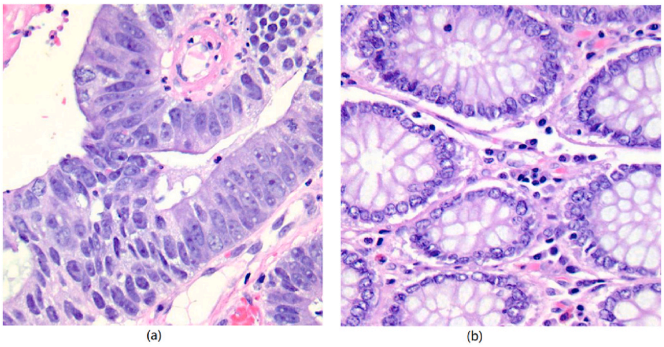


Fig. 4. (a) adenocarcinoma, (b) benign tissue (https://arxiv.org/abs/1912.12142v1, https://github.com/tampapath/lung_colon_image_set) [83].

Lung and Colon Cancer Histopathological Images (LCc) (https://arxiv.org/abs/1912.12142v1, https://github.com/tampapath/lung_colon_image_set) dataset contains 25000 images, with 5 decision classes, each class having 5000 samples. The histopathological images are 768 × 768, and we have resized them at 250 × 250. The dataset is built using the Augmentor package from 750 images of lung tissues (250 cases of benign tissue, 250 cases of lung adenocarcinomas, and 250 cases of squamous cell carcinoma), and 500 images of colon tissue (250 cases of benign tissue, and 250 cases of colon adenocarcinomas). We have split the dataset into two, one concerning lung cancer *Lc* (3 decision classes), and the other one concerning colon cancer *Cc* (2 decision classes). Fig. 3 presents three sample images from Lc, while Fig. 4 presents two sample images from Cc [83].

The maternal-fetal ultrasound dataset (https://zenodo.org/record/3904280#.YfjeTPVBzL9) was collected from two different hospitals. The dataset is split into two different sets. The first set (*FP*) contains 6 classes, 4 of which regard the fetal anatomical planes: abdomen (711 cases), brain (3092 cases), femur (1040 cases), and thorax (1718 cases), the fifth regarding the mother's cervix (1626 cases), and the last one includes the less common image plane (4213 cases). The second set (*FB*) contains images of the brain planes that are split in 3 classes: *trans*-thalamic (1638 cases), *trans*-cerebellum (714 cases), *trans*-ventricular (597 cases). The first set has 12 400 images,

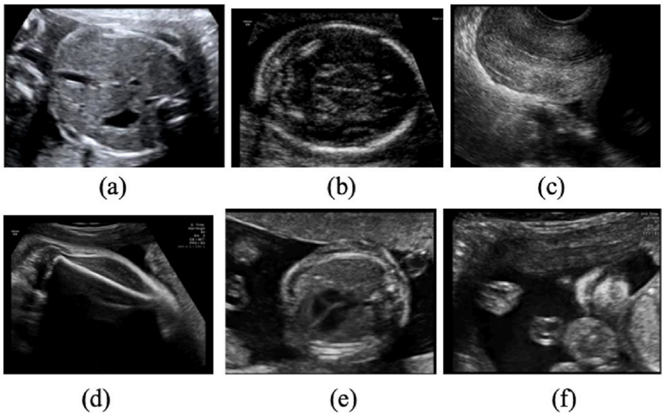


Fig. 5. (a) fetal abdomen, (b) fetal brain, (c) maternal cervix, (d) fetal femur, (e) fetal thorax, (f) other (https://zenodo.org/record/3904280#.YfjeTPVBzL9 [37],.

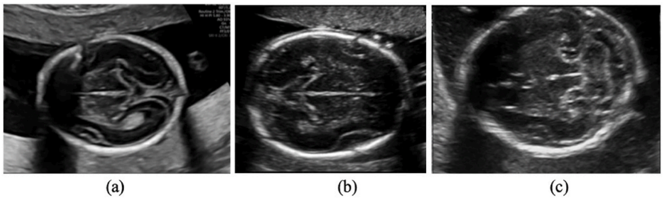


Fig. 6. (a). trans-ventricular, (b) *trans*-thalamic, (c) *trans*-cerebellar (https://zenodo.org/record/3904280#.YfjeTPVBzL9 [37].

while the second contains 2949, (Burgos-Artizzu, 2020). Fig. 5 presents three sample images from FP, while Fig. 6 presents two sample images from FB [37].

5.2. Design of experiments and parameter settings

In this study, we have compared the performance of the CNN, which had its architecture established through DE, with the performance of state-of-the-art DLs: VGG-16, ResNet50, Inception V3, and DenseNet169. All models were run on the five datasets.

To assess the models' performances, we have used the *10-fold cross-validation* as validation method. For an effective and objective evaluation of the CNN algorithms, we have evaluated their results through a throughout statistical analysis. The following rule has been applied to all the methods that have been compared in this study: each method was executed in 100 independent runs (i.e. 100 times in a complete cross-validation cycle). The purpose of this procedure is to estimate the sample size needed for a high statistical power. A model lacks in performance if the sample size is too low, or on the contrary, using too many computational and time resources might not lead to a significant increase in performance. Hence, using a sample size of 100 computer runs, we have obtained a statistical power greater than 95%, with type I error $\alpha = 0.05$, for all the statistical tests that have been performed. The average accuracy over 100 complete cross-validation cycles (ACA) is recorded for each model. Besides the ACA, we have computed the standard deviation (SD), the 95% confidence interval (CI 95%), precision, recall, and F1-score. The standard deviation gives us an insight on the model's stability. To demonstrate the *omnibus* robustness, the methods must be applied on multiple datasets. If the SD varies from dataset to dataset, from smaller to larger values, then the method has failed in providing the *omnibus* robustness. We have considered the Precision-Recall curves (PR AUC) and not the Area under the ROC curve (AUC), because the datasets are imbalanced. Imbalanced data can lead to a probable change in false positives, which are used in computing the false positive rate used by AUC. Using PR AUC we obtain more precise results, due to the fact that we compare false positives with true positives, and not true negatives, as in the AUC case.

The statistical evaluation involved the following tests, which have been applied on the sample which contained 100 performances obtained after running the method 100 independent computer runs on the test set:

- Normality tests: the *Kolmogorov-Smirnov & Lilliefors* test and *Shapiro-Wilk W* test. We have applied both tests to check whether the performance samples are governed or not by the Gaussian distribution. We must keep in mind that if the tests' results show that the data is not Normal, then we can make use of the *Central Limit Theorem*, that states that if the sample size is large enough (surpasses 30), then the distribution of the sample is approximately Normal [84].
- Equality of variances: *Levene's* and *Brown-Forsythe* tests. If the samples have unequal variances, then the Type I error might be affected, resulting false positives. In practice, the issue is less problematic, since we are using the samples with the same size (in our case 100) [84,85].

There are different methods for testing whether the data sample is governed by the Normal distribution or not. We have used the Shapiro-Wilk test because it has more power to detect the non-normality, but it is used in general for smaller sample sizes. Kolmogorov-Smirnov & Lilliefors test is recommended for larger sizes, but has a lesser power [86].

If the normality assumption and the equality of variances assumption are met, then we can proceed and apply *t*-test, *One-Way ANOVA* together with Tukey's post-hoc test to differentiate between the algorithms' performances. The One-Way ANOVA is used to establish whether there are any statistically significant differences between the means of three or more samples. It is important to understand that One-Way ANOVA is an omnibus test statistic that cannot reveal which groups are statistically significantly different from each other. Thus, to determine which samples differ from the others, we need to use a post-hoc test and the *t*-test for independent variables. Tukey's Honest Significant Difference compares all possible pair of means and gives us the answer.

The initial population of the DE/CNN contained $N = 20$ candidate solutions, and $G = 50$ generations. The recombination probability was generated from the interval [0.5, 0.7], so that new architectures to be generated at a faster pace, while the mutation variation factor was generated from the interval [0.3, 0.6]. The initial population candidate solutions were created from the intervals $[1, 6]$ and $[20, 300]$, while the kernel sizes from [2, 5]. We have set the spatial stride 1. We have applied 10 training epochs with a batch size of 64. The training phase of each candidate solution took around 4.7 min on a GeForce RTX 3070 GPU.

Table 1
DE/CNN Average accuracy over 100 computer runs (ACA%), standard deviation (SD), CI 95%, precision, recall, F1-score, and network structure on the Bc, Lc, Cc, FP, and FP datasets.

Database	ACA	SD	CI 95%	Precision	Recall	F1-score	Structure (no. convolutions/no. hidden units
Bc	90.04	2.322	(89.57, 90.50)	0.90073	0.90	0.90027	(4, 273)
Lc	(99.05)	0.808	(98.88, 99.21)	0.99072	0.99	0.99028	(5, 235)
Cc	99.50	0.502	(99.40, 99.59)	0.99919	1	0.99911	(4, 214)
FP	96.29	2.701	(95.75, 96.82)	0.99305	0.98	0.98646	(4, 233)
FB	78.73	1.994	(78.33, 79.12)	0.78634	0.77	0.78148	(5, 220)

Table 2
Testing the normality of the DE/CNN's ACA.

Database	Kolmogorov-Smirnov		Shapiro-Wilk W	
	K–S max *D*	Lilliefors *p*	*S–W W*	*p*-level
Bc	0.268	0.01	0.690	0.000
Lc	0.355	0.01	0.635	0.000
Cc	0.306	0.01	0.670	0.000
FP	0.196	0.01	0.878	0.000
FB	0.287	0.01	0.798	0.000

Table 3
Average ACAs of other DLs vs. DE/CNN.

Algorithm	Datasets – validation accuracies (%)				
	Bc	Lc	Cc	FP	FB
DE/CNN	**90.04**	**99.05**	**99.50**	**96.20**	**78.73**
VGG16	86.97	97.41	99.03	83.18	68.75
ResNet50	87.14	95.84	99.11	89.44	71.45
Inception V3	86.24	99.18	99.49	91.13	72.25
DenseNet 169	91.04	99.47	99.52	92.98	74.02

Table 4
Testing the equality of variances.

Dataset	Variable	Levene F(1, df)/*p*-level	Brown-Forsythe (1,df)/*p*-level
Bc	DE/CNN vs. VGG	5.247/0.023	4.927/0.027
	DE/CNN vs. ResNet50	0.804/0.370	0.928/0.336
	DE/CNN vs. Inception v3	8.997/0.003	8.654/0.003
	DE/CNN vs. DesNet169	8.503/0.003	7.880/0.0054
Lc	DE/CNN vs. VGG	77.006/0.000	59.172/0.000
	DE/CNN vs. ResNet50	143.537/0.000	138.422/0.000
	DE/CNN vs. Inception v3	0.156/0.692	0.021/0.8824
	DE/CNN vs. DesNet169	13.359/0.000	6.730/0.010
Cc	DE/CNN vs. VGG	19.355/0.000	16.708/0.000
	DE/CNN vs. ResNet50	54.949/0.000	40.751/0.000
	DE/CNN vs. Inception v3	0.039/0.842	0.039/0.842
	DE/CNN vs. DesNet169	0.158/0.690	0.158/0.690
FP	DE/CNN vs. VGG	9.587/0.002	3.273/0.071
	DE/CNN vs. ResNet50	0.505/0.477	0.300/0.584
	DE/CNN vs. Inception v3	67.540/0.000	42.472/0.000
	DE/CNN vs. DesNet169	2.204/0.139	1.551/0.214
FB	DE/CNN vs. VGG	54.763/0.000	29.474/0.000
	DE/CNN vs. ResNet50	57.660/0.000	37.259/0.000
	DE/CNN vs. Inception v3	81.663/0.000	37.723/0.000
	DE/CNN vs. DesNet169	178.368/0.000	38.364/0.000

The experiment can be redone with different settings. DE was able to find competitive CNN's architectures that achieved high performances. Even if at the begging of the process the accuracy of the best candidate solution was not high enough, the accuracies started to improve generation by generation. Being an optimizing method, DE assures that at the end of the process the candidate solution will be improved.

The results of the experiments are described in the Results section. The DE/CNN together with the above mentioned DLs were evaluated over the three benchmarked datasets.

6. Results

6.1. Experimental results

The results of the experiments regarding the classification of brain tumors, lung, colon cancer, maternal-fetal planes, and brain planes obtained after applying DE/CNN are depicting in Table 1 in terms of ACA, SD, 95% confidence interval (CI), precision, recall, F1-score, and network's structure.

From Table 1 we can see that DE/CNN performs excellent on the Lc, Cc, and FP datasets, (99.05, 99.50, and 96.29% accuracies), very good on the Bc dataset (90.04% accuracy), and good on the FB dataset (78.73% accuracy). The SDs are fairly small on Bc, Lc, Cc, and FP sets, ranging from 0.502 to 2.701, proving the robustness and stability of the model.

In what follows, we shall present the *data screening* process that involved the *Kolmogorov-Smirnov* and *Lilliefors test* and the *Shapiro Wilk W* test. Table 2 show the obtained results.

From Table 2, we can see that no matter what dataset we have applied our model on, the sample data is not governed by the Normal distribution. Recall that we have mentioned that in this sort of situation we can always make use of the *Central Limit Theorem*, therefore by having a sample size of 100 ACAs we can assume that the distributions are approximately Gaussian, and we can carry on with the other statistical tests.

6.2. Statistical assessment

We have evaluated the DE/CNN by statistically comparing its results with the performances of other DLs applied on the same datasets. The competitors of the proposed model are the following state-of-the-art DLs:

- *VGG16 i*s considered to have an excellent architecture. It has won the ILSVR (ImageNet) competition in 2014. The VGG16 does not have large number of hyper-parameters. Instead, its architecture consists of convolutional layers of 3 × 3 filter with stride 1, same padding, and a maxpooling layer of 2 × 2 filter with stride 2. After a series of convolutions followed by maxpooling, the architecture ends with two fully connected layers and a softmax for the output. The VGG16 has 16 layers that have weights [87].
- *ResNet50* stands for Residual Network 50. It has won the ILSVR (ImageNet) competition in 2015. ResNet's signature is the concept of skip connection. The skip connection allows an alternative cutoff route for the gradient to flow through the network. In this way, the model can learn an identity function so that any higher level performs as well as a lower layer in the CNN. The ResNet50 has 48 convolutional layers, 1 maxpool, and 1 averagepool layer [88].
- *Inception V3* aims to be more computational efficient. Its architecture is progressively built starting with factorized convolutions that reduce the computational efficiency by reducing the number of parameters in the network. Another characteristic of the Inception V3 architecture is that it replaces bigger convolutions with smaller convolutions, speeding up the training process. Besides smaller convolutions, the network supports asymmetric convolutions, and an auxiliary classifier (a small CNN) inserted in its architecture during training between other layers. This classifier acts as a regularizer. Inception V3 reduces the grid size through pooling operations [89].
- *DenseNet129* is short for Dense CNN. In DenseNet each layer receives additional input, known as collective knowledge from all the previous layers. In this way, the network is more compact, with fewer channels. Instead of using a deep architecture, DenseNet reuses the features. It does not sum the output feature map of the layer with the following feature map, it concatenates them [90].

Table 5
One-way ANOVA results.

Dataset	SS	df	MS	*F*-value	*p*-level
Bc	1789	4	447	75	0.0000
Lc	949	4	237	129	0.0000
Cc	23	4	6	13	0.0000
FP	9472	4	2368	167	0.0000
FB	5489	3	1372	30.20	0.0000

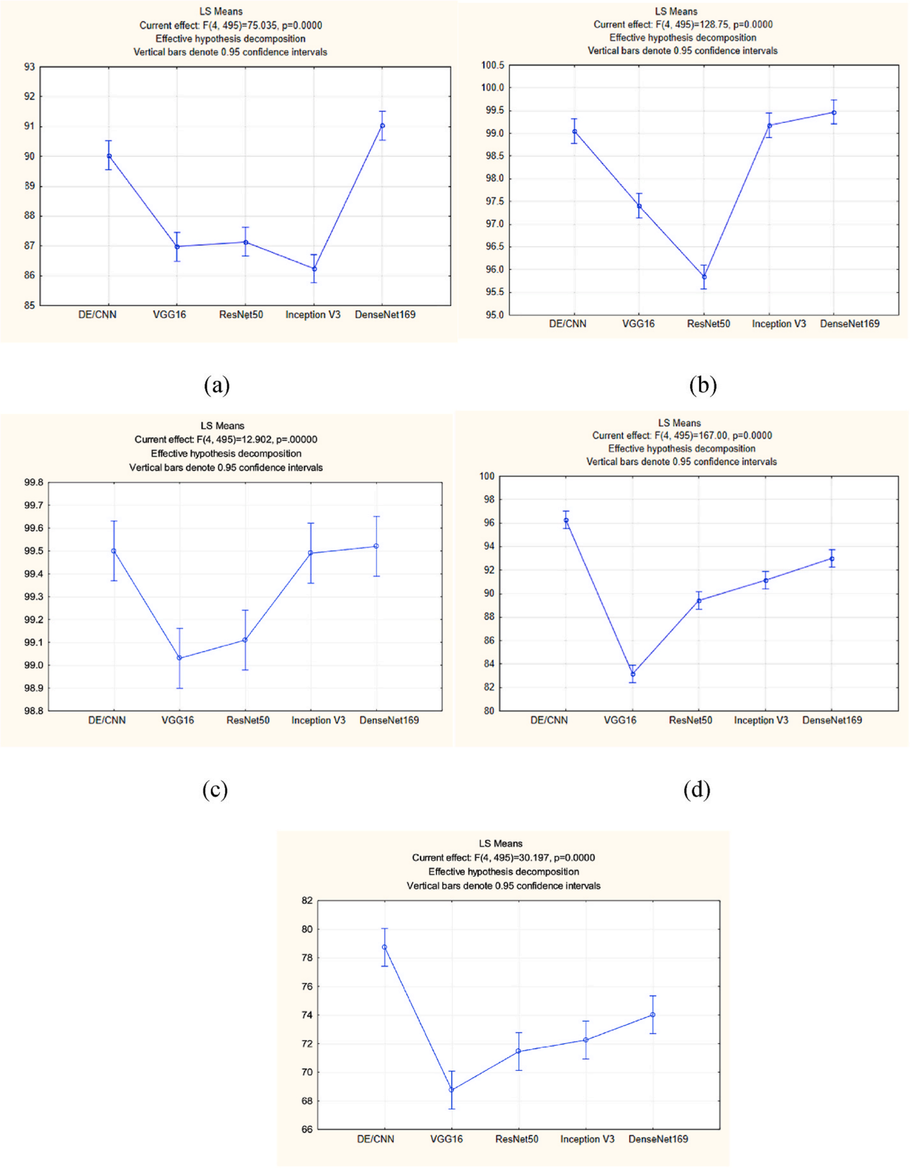
All the algorithms have been run under the same conditions for the comparison to be fair and objective. The results are displayed in Table 3.

From both Table 3 we can see that the DE/CNN performs almost the same as all the other DLs on the Cc dataset, on the Bc dataset performs better than VGG, ResNet50 and Inception V3, but worse than DenseNet 169, whereas on Lc dataset it performs better than VGG and ResNet50, and comparable as Inception V3 and DenseNet169. On the FP and FB dataset the DE/CNN surpassed the other DLs. This proves that using DE to determine a CNNs architecture can be fruitful.

We were interested in verifying the equality of variances on each dataset using *Levene's* and *Brown-Forsythe* tests. This was an important step in our statistical analysis, because we wanted to apply *One-Way ANOVA* and post-hoc Tukey to verify whether indeed, they were or weren't any statistical significant differences between our proposed model and the other competitors. Table 4 presents the results of the two tests.

From Table 4 we can draw the following conclusions: DE/CNN vs. ResNet50 on the Bc dataset, DE/CNN vs. Inception V3 for the Lc dataset, and DE/CNN vs. Inception V3, and DenseNet169 for the Cc dataset, DE/CNN vs. ResNet50, and DenseNet169 for the FP have equal variances. This implies that they behave the same. In the rest of the cases all the models behave differently having different variances (p-level < 0.05). We made use of the fact that we have the same number of observations in each sample (100 computer runs) and proceeded with applying One-Way ANOVA and post-hoc Tukey, to verify whether indeed there are statistical difference between the competitors' behavior. The One-Way ANOVA results are depicted in Table 5.

Table 5 presents the differences between the DL's validation



(a)

(b)

(c)

(d)

(e)

Fig. 7. One-way ANOVA – Least Square Means (a) Bc dataset, (b) Lc dataset, (c), Cc dataset, (d) FP dataset, (e) FB dataset.

Table 6
Tukey's post-hoc test results.

Dataset		DE/CNN	VGG16	ResNet50	Inception V3	DenseNet169
Bc	DE/CNN		**0.0000**	**0.0000**	**0.0000**	0.3098
	VGG16	**0.0000**		0.9881	0.2139	**0.0000**
	ResNet50	**0.0000**	0.9881		0.0691	**0.0000**
	Inception V3	**0.0000**	0.2139	0.0691		**0.0000**
	DenseNet169	0.3098	**0.0000**	**0.0000**	**0.0000**	
		DE/CNN	VGG16	ResNet50	Inception V3	DenseNet169
Lc	DE/CNN		**0.0000**	**0.0000**	0.9613	0.1841
	VGG16	**0.0000**		**0.0000**	**0.0000**	**0.0000**
	ResNet50	**0.0000**	**0.0000**		**0.0000**	**0.0000**
	Inception V3	0.9613	**0.0000**	**0.0000**		0.5554
	DenseNet169	0.1841	**0.0000**	**0.0000**	0.5554	
		DE/CNN	VGG16	ResNet50	Inception V3	DenseNet169
Cc	DE/CNN		**0.0000**	**0.0003**	0.9999	0.9995
	VGG16	**0.0000**		0.9150	**0.0000**	**0.0000**
	ResNet50	**0.0003**	0.9159		**0.0005**	**0.0001**
	Inception V3	0.9999	**0.0000**	**0.0005**		0.9977
	DenseNet169	0.9995	**0.0000**	**0.0001**	0.9977	
		DE/CNN	VGG16	ResNet50	Inception V3	DenseNet169
FP	DE/CNN		**0.0000**	**0.0000**	**0.0000**	**0.0000**
	VGG16	**0.0000**		**0.0000**	**0.0000**	**0.0000**
	ResNet50	**0.0000**	**0.0000**		**0.0130**	**0.0000**
	Inception V3	**0.0000**	**0.0000**	**0.0130**		**0.0046**
	DenseNet169	**0.0000**	**0.0000**	**0.0000**	**0.0046**	
		DE/CNN	VGG16	ResNet50	Inception V3	DenseNet169
FB	DE/CNN		**0.0000**	**0.0000**	**0.0000**	**0.0000**
	VGG16	**0.0000**		**0.0373**	**0.0022**	**0.0000**
	ResNet50	**0.0000**	**0.0373**		0.9184	00545
	Inception V3	**0.0000**	**0.0022**	0.9184		0.3411
	DenseNet169	**0.0000**	**0.0000**	0.0545	0.3411	

performances in terms if sums of squares (SS), degrees of freedom (df), mean squares (MS), *F*-value, and *p*-level (contrast quadratic polynomial), [91,92]. Even if the accuracies seem close enough, the test reveals that there are statistical differences between the competitors. In Fig. 7a, b, 7c, 7d, and 7e we present the visual representations of the least squares means for the five datasets.

The One-way ANOVA revealed that there are significant differences between the competitors' performances, but one should ask the following question: are there differences between all the competitors, or just between some of them? To answer the question above, we have applied Tukey's post-hoc test. Its results presented in Table 6 shows that: on the Bc dataset there are significant differences between the performances of DE/CNN vs. all the rest of the DLs, except for DenseNet169; on the Lc dataset there are significant differences between the performances of DE/CNN vs. VGG16 and ResNet50; on the Cc dataset there are significant differences between the performances of DE/CNN vs. VGG16 and ResNet50; while on the FP and FB datasets there were significant difference between DE/CNN and all the DLs, proving that the method improves the performance.

For a better visualization, we present in Fig. 8a, b, 8c, 8d, and 8e, the distribution of the samples that contain the accuracies of each CNN recorded after 100 computer runs in the shape of boxplots together with the obtained *p*-values obtained after applying the *t*-test for independent variables. We demonstrate once more that there are statistical differences between the models.

The benchmarking process was completed by presenting results that have been reported in literature on the same five datasets. It should be noted that these results were obtained by networks which were pretrained on ImageNet Large Scale Visual Recognition Challenge, and then fully retrained using these datasets. In our study, the networks were not previously pre-trained. The datasets are recent, therefore there are not many papers in recent literature (2020–2022) that regard them. Through Tables 7–10 we enable a fair and direct comparison between DE/CNN and the most recent methodologies.

Regarding the FB dataset [97], reported 74% accuracy, the result of the best performing CNN, the DenseNet-169.

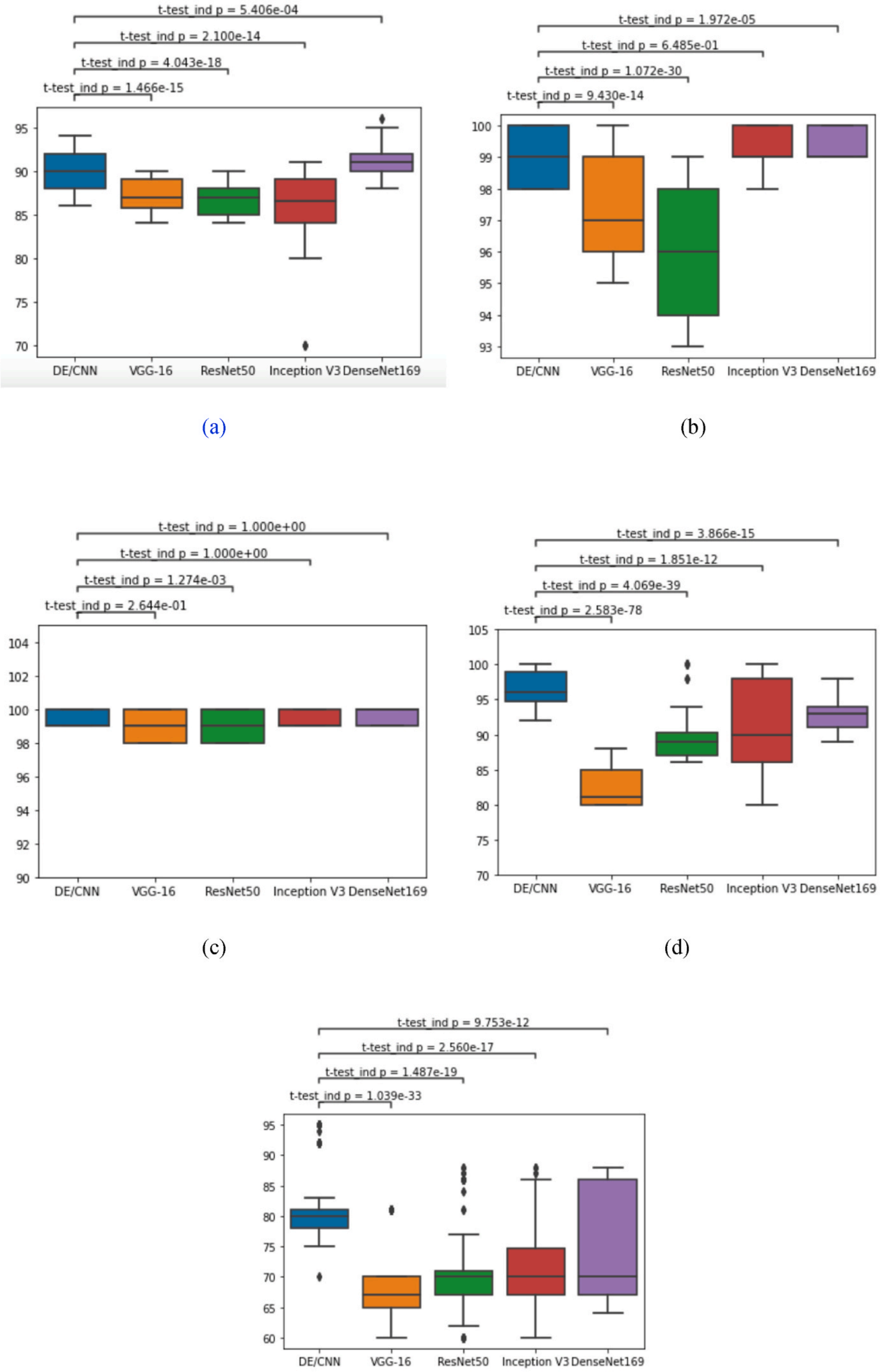
7. Discussion

In this study we have proposed a new manner of applying neuroevolution for establishing the architecture of a CNN by using DE. The method was applied on three cancer datasets that contain MRI scans and histopathological images. Correspondingly to the statistical analysis performed, even if the validation accuracies values do not vary so much, there are significantly differences in performances between the models.

Through this study we propose a new method of choosing a CNN's architecture and also a statistical framework for validating the results of different DLs. We have demonstrated that even if the performances of different methods might seem almost equal, in fact the differences between them are statistically significant. The statistical framework included benchmarking results in terms of ACA, SD, 95% CI, precision, recall, F1-score, along with Kolmogorov-Smirnov and Lilliefors, Shapiro-Wilk W, Levene, Brown-Forsythe, *t*-test for independent variables, One-way ANOVA and Tukey's post-hoc tests. The results revealed that even if on the Bc dataset there were differences between the DE/CNN and the other DLs, on the Lc and Cc datasets the DE/CNN obtained comparable results with Inception V3 and DenseNet 169, whereas on FP and FB, the DE/CNN obtained better performances. It can be seen that the performance of a classifier depends on the dataset used.

8. Conclusions

In this study we propose a new way to determine the architecture of a deep neural network through the use of differential evolution. At first, we proposed an encoding method for representing the structure of each CNN with a fixed-length integer array, after which we have used differential evolution processes (mutation, recombination, and selection) to explore in an efficient manner the search space. We have tested our method on three cancer related datasets that contain MRI scans and histopathological images concerning brain, lung, and colon tumors. The experimental results were further statistically analyzed in comparison with the results obtained by other state-of-the art DLs. The findings show that this neuroevolution method for determining a CNNs architecture is competitive with other methods.



(a)

(b)

(c)

(d)

(e)

Fig. 8. Distribution boxplot together with *p*-level: (a) Bc dataset, (b) Lc dataset, (c) Cc dataset, (d) Fp dataset, (e) Fb dataset.

Table 7
Classification performance on the Bc dataset reported in literature of other ML models.

Reference	Fully connected	Gaussian Naïve Bayes	AdaBoost	k-NN	Random Forest	SVM (linear)	SVM (sigmoid)	ELM
[93]	87.88%	68.51%	69.82%	86.48%	84.53%	87.48%	90.19%	86.10%

Despite the interesting results, our method still has some drawbacks that will be resolved in future works. The limitation of the study consists in the fact that we use the same number of hidden neurons for each convolutional layer. We aim to explore how the performance changes when we set different numbers of hidden neurons using differential evolution. Also, in future studies we wish to see whether the performance improves if we use pretrained networks.

Table 8
Classification performance on the Lc dataset reported in literature of other ML models.

Reference	Shallow CNN	3 layered CNN
[94]	97.89%	
[95]		96.33%

Table 9
Classification performance on the Cc dataset reported in literature of other ML models.

Reference	Shallow CNN	ResNet-18	ResNet-30	ResNet-50
[94]	96.61%			
[96]		93.04%	93.04%	93.91%

Table 10
Classification performance on the FP dataset reported in literature of other ML models.

Reference	VGG-M	VGG-16	MobileNet	Inception-V3	ResNet-18
[97]	84.8%	92.1%	87.5%	93.5%	92.5%
	ResNet-34	ResNet-50	ResNet-101	ResNet-152	ResNeXt-50
	92.5%	93.1%	93.4%	92.8%	92.7%
	ResNeXt-101	SENet	SE-ResNet-50	SE-ResNet-101	SE-ResNet-152
	94%	92.9%	93.3%	93.3%	92.7%
	SE-ResNeXt-50	SE-ResNeXt-101	DenseNet-121	DenseNet-169	PCA + boosting
	92.7%	92.7%	92.9%	93.6%	54.7%
	Hog + boosting				
	68.6%				

Declaration of competing 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] E. Feletto, P. Grogan, C. Nickson, M. Smith, K. Canfell, How has COVID-19 impacted cancer screening? Adaptation of services and the future outlook in Australia, Publ. Health Res. Pract. 30 (4) (2020), e3042026.

[2] K.Y.Y. Ng, S. Zhou, S.H. Tan, N.D.B. Ishak, Z.Z.S. Goh, Z.Y. Chua, et al., Understanding the psychological impact of COVID-19 pandemic on patients with cancer, their caregivers, and health care workers in Singapore, JCO Global Oncol. 6 (2020) 1494–1509.

[3] J. van de Haar, L.R. Hoes, C.E. Coles, K. Seamon, S. Frohling, D. Jager, et al., Caring for patients with cancer in the COVID-19 era, Nat. Med. 26 (5) (2020) 665–671.

[4] A. van Dorn, COVID-19 and readjusting clinical trials, Lancet (London, England) 396 (1025) (2020) 523–524.

[5] A.M. Young, F.D. Ashbury, L. Schapira, F. Scotte, C.I. Ripamonti, I.N. Olver, Uncertainty upon Uncertainty: Supportive Care for Cancer and COVID-19, Support Care Cancer, 2020.

[6] J. Deprest, M. Choolani, F. Chervenak, et al., Fetal diagnosis and therapy during the COVID-19 Pandemic: guidance on behalf of the international fetal medicine and surgery society, Fetal Diagn. Ther. 47 (2020) 689–698, https://doi.org/10.1159/000508254.

[7] A.I. Mazur-Bialy, D.K. Bogucka, S. Tim, M. Oplawski, Pregnancy and Childbirth in the COVID-19 Era – the course of disease and maternal-fetal transmission, J. Clin. Med. 9 (11) (2020) 3749, https://doi.org/10.3390/jcm9113749.

[8] B. Chmielewska, I. Barratt, R. Townsend, et al., Effects of the COVID-19 pandemic on maternal and perinatal outcomes: a systematic review and meta-analysis, Lancet Global Health (2021), https://doi.org/10.1016/S2214-109X(21)00079-6.

[9] I. Alkatout, M. Biebl, Z. Momenimovahed, E. Giovannucci, F. Hadavandsiri, H. Salehiniya, L. Allahgoli, How COVID-19 affected cancer screening programs? A systematic review, Front. Oncol. 11 (2021), 675038.

[10] K. Gong, Z. Xu, Z. Cai, Y. Chen, Z. Wang, Internet hospitals help prevent and control the epidemic of COVID-19 in China: multicenter user profiling study, J. Med. Internet Res. 22 (4) (2020), e18908.

[11] S.Y. Cheng, C.F. Chen, H.C. He, L.C. Chang, W.F. Hsu, M.S. Wu, et al., Impact of COVID-19 pandemic on fecal immunochemical test screening uptake and compliance to diagnostic colonoscopy, J. Gastroenterol. Hepatol. 20 (2020), https://doi.org/10.1111/jgh15325.

[12] A.G. Dinmohamed, O. Visser, R.H.A. Verhoeven, M.W.J. Louwman, F.H. van Nederveen, S.M. Willems, et al., Fewer Cancer diagnoses during the COVID-19 epidemic in The Netherlands, Lancet Oncol. 21 (6) (2020) 750–751.

[13] D. Patt, L. Gordan, M. Diaz, T. Okon, L. Grady, M. Harmison, et al., Impact of COVID-19 on cancer care: how the pandemic is delaying cancer diagnosis and treatment for American seniors, JCO Clin. Cancer Inf. 4 (2020) 1059–1071.

[14] M. Lang, T. Yeung, J.O. Shepard, A. Sharma, M. Petranovic, E.J. Flores, et al., Operational Challenges of a low-dose CT lung cancer screening program during the coronavirus disease 2019 pandemic, Chest 159 (3) (2020) 1288–1291.

[15] R.V. Mathew, K. Oliver, S. Farrimond, et al., Brain tumors and COVID-19: the patients and caregiver experience, Neurooncol. Adv. 2 (1) (2020) vdaa104.

[16] R. Dube, S.S. Kar, COVID-19 in pregnancy: the foetal perspective-a systematic review, Neonatology 4 (1) (2020), https://doi.org/10.1136/bmjpo-2020-000859.

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

[18] D. Paladini, Sonography in obese and overweight pregnant women: clinical, medicolegal and tehncial issues, Ultrasound Obstet. Gynecol. 33 (6) (2009) 720–729.

[19] E.J. Topol, High performances medicine: the convergence of human and artificial intelligence, Nat. Med. 25 (2019) 44–46.

[20] S. Benjamens, P. Dhunno, B. Mesko, The state of artificial intelligence-based FDA approved medical devices and algorithms: an online database, NPJ Digit. Med. 3 (2020) 118.

[21] X. Liu, et al., A comparison of deep learning performances against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis, Lancet Digit. Health 1 (2019) e271–297.

[22] D. Kumar, A. Wong, D.A. Clausi, Lung nodule classification using deep features in CT images, 12th Conf. Comput. Robot Vis. (2015) 133–138.

[23] W. Sun, B. Zheng, W. Qian, Computer aided lung cancer diagnosis with deep learning algorithms, Med. Imaging: Computer-Aided Diagnosis 9785 (2016), 97850Z.

[24] N. Coudray, P.S. Ocampo, T. Sakellaropoulos, et al., Classification and mutation prediction from non-small cell lung cancer histopathology images using deep learning, Nat. Med. 24 (10) (2018) 1559–1567.

[25] Q. Song, L. Zhao, X. Luo, X. Dou, Using deep learning for classification of lung nodules on computed tomography images, J. Healthc. Eng. (2017), 8314740.

[26] S. Bhatia, Y. Sinha, L. Goel, Lung Cancer Detection: a Deep Learning Approach, Soft Comp for Probl Sol, Singer, Singapore, 2019, pp. 699–705.

[27] A. Teramoto, H. Fujita, O. Yamamuro, T. Tamaki, Automated detection of pulmonary nodules in PET/CT images: ensemble of false-positive reduction using a convolutional neural network technique, Med. Phys. 43 (2016) 2821–2827.

[28] H. Chen, H. Zhao, J. Shen, R. Zhou, Q. Zhou, Supervised machine learning model for high dimensional gene data in colon cancer detection, IEEE Int. Congr. Big Data (2015) 134–141.

[29] K. Sirinukunwattana, S.E. Raza, Y.W. Tsang, D.R. Snead, I.A. Cree, N.M. Rajpoot, Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images, IEEE Trans. Med. Imag. 35 (5) (2016) 1196–1206.

[30] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P. M. Jodoin, H. Larochelle, Brain Tumor Segmentation with Deep Neural Networks, 2015 arxiv.org/abs/1505.03540.

[31] Z. Xiao, R. Huang, Y. Ding, T. Lan, R. Dong, Z. Qin, X. Zhang, W. Wang, A deep learning-based segmentation method for brain tumor in MR images, in: IEEE 6th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS), 2016, pp. 1–6.

[32] H. Dong, Yang, et al., in: Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks, Annul Conference on Medical Image Understanding and Analysis, Springer, 2017, pp. 506–517.

[33] M. Rezaei, et al., A Conditional Adversarial Network for Semantic Segmentation of Brain Tumor, International MICCAI Brainlesion Workshop, Springer, 2017, pp. 241–252.

[34] X. Zhao, et al., A deep learning model integrating FCNNs and CRFs for brain tumor segmentation, Med. Image Anal. 1 (43) (2018) 98–111.

[35] K. Munir, H. Elahi, A. Ayub, F. Frezz, A. Rizzi, Cancer diagnosis using deep learning: a bibliographic review, Cancers 11 (9) (2019) 1235.

[36] M.Z. Alom, et al., A state-of-the-art survey on deep learning theory and architectures, Elecronics 8 (3) (2019) 292.

[37] X.P. Burgos-Artizzu, et al., FETAL_PLANES_DB: common maternal-fetal ultrasound images, in: Nature Scientific Reports, vol. 19, Zenodo, 2020, p. 10200, 1.0, https://doi.org/10.5281/zenodo.3904280.

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

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

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

[41] M. Phillip, et al., Convolutional Neural Networks for automated fetal cardiac assessment using 4D B-Mode ultrasound, in: IEEE 16th International Symposium on Biomedical Imaging, 2019, pp. 824–828, https://doi.org/10.1109/ISBI.2019.8759377.

[42] J. Torrents-Barrena, et al., Assessment of radiomics and deep learning for the segmentation of fetal and maternal anatomy in magnetic resonance imaging and ultrasound, Acad. Radiol. S1076–6332 (19) (2019) 30575–30576.

[43] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997) 67.

[44] B. Baker, O. Gupta, N. Naik, R. Raskar, Designing neural network architectures using reinforcement learning, in: International Conference on Learning Representations,, ICLR, 2017, p. 2017.

[45] H. Cai, T. Chen, W. Zhang, Y. Yu, J. Wang, Efficient Architecture Search by Network Transformation, Association for the Advancement of Artificial Intelligence, 2018, p. 2018.

[46] Z. Zhong, J. Yan, C.L. Liu, in: Practical Network Blocks Design with Q-Learning, International Conference on Learning Representations, vol. 2017, ICLR, 2018.

[47] B. Zoph, Q.V. Le, Neural architecture search with reinforcement learning, in: International Conference on Learning Representations, vol. 2017, ICLR, 2017.

[48] B. Zoph, V. Vasudevan, J. Shlens, Q.V. Le, in: Learning Transferable Architectures for Scalable Image Recognition, Conference on Computer Vision and Pattern Recognition, vol. 2018, 2018.

[49] C. Liu, B. Zoph, et al., Progressive neural architecture search, in: European Conference on Computer Vision, 2018, p. 2018.

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

[51] E. Real, S. Moore, et al., Large-scale evolution for image classifiers, in: Proceedings of the 34th International Conference on Machine Learning, vol. 70, ICML, 2017, pp. 2902–2911, 17.

[52] E. Real, A. Aggarwal, Y. Huang, Q.V. Le, Regularized evolution for image classifier architecture search, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, the Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, the Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAAI 2019, Honolulu, Hawaii, USA, 2019, pp. 4780–4789.

[53] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Evolving deep convolutional neural networks for image classification, IEEE Trans. Evol. Comput. 24 (2) (2020) 394–407.

[54] Y. Sun, B. Xue, M. Zhang, G.G. Yen, Completely automated CNN architecture design based on blocks, IEEE Transact. Neural Networks Learn. Syst. 31 (4) (2020) 1242–1254.

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

[56] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (1992) 229–256.

[57] J. Schulman, et al., Proximal Policy Optimization Algorithm, 2017. CoRR abs/1707.06347.

[58] S. Whitelam, I. Tamblyn, Learning to grow: control of material self-assembly using evolutionary reinforcement learning, Phys. Rev. E E. 101 (2020), 052604, https://doi.org/10.1103/PhysRevE.101.052604.

[59] E. Lomurno, S. Samele, M. Matteucci, D. Ardagna, Pareto-optimal progressive neural architecture search, in: GECCO'21: Proceedings of the Genetic and Evolutionary Computations Conference Companion, 2021, pp. 1726–1734.

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

[61] H. Liu, K. Simonayan, O. Vinyals, C. Fernando, K. Kavukcuoglu, Hierarchical representations for efficient architecture search, in: ICLR, 2018.

[62] K.O. Stanley, R. Miikkulainen, Evolving neural networks through augmenting topologies, Evol. Comput. 10 (2002) 99–127.

[63] J. Hajewski, S. Oliviera, X. Xing, Distributed Evolution of Deep Autoencoders, 2020 arxiv.org/abs/2004.07607.

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

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

[66] Y. Bahri, J. Kadmon, J. Pennington, S.S. Schoenholz, J. Sohl-Dickstein, S. Ganguli, Statistical mechanics of deep learning, Annu. Rev. Condens. Matter Phys. 11 (2020) 501–528, https://doi.org/10.1146/annurev-conmatphys-031119-050745.

[67] K.O. Stanley, J. Clune, J. Lehman, R. Miikkulainen, Designing neural networks through neuroevolution, Nat. Mach. Intell. 1 (2019) 24–35, https://doi.org/10.1038/s42256-018-0006-z.

[68] E. Galvan, P. Mooney, Neuroevolution in deep neural networks: current trends and future challenges, IEEE Trans. Artif. Intell. 2 (2021) 476–493, https://doi.org/10.1109/TAI.2021.3067574.

[69] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Weinberger, Deep Networks with Stochastic Depth, European Conference on Compute Vision, 2016.

[70] S. Ioffe, C. Szegedy, Accelerating Deep Network Training by Reducing Internal Covariate Shift, International Conference on machine Learning, 2015, p. 2015.

[71] R. Storn, K. Price, Differential-evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[72] R. Storn, K. Price, Differential evolution for multi-objective optimization, Evol. Comput. 4 (2003) 8–12, 2003.

[73] M.G.H. Omran, A.P. Engelbrecht, Self-adaptive differential evolution methods for unsupervised image classification, Proc. IEEE Conf. Cybern. Intell. Syst. (2006) 1–6.

[74] V. Aslantas, M. Tunckanat, Differential evolution algorithm for segmentation of wound images, in: Proc. Of the IEEE Interantional Symposium on Intelligent Signal Processing (WISP), 2007.

[75] H. Dhahri, A.M. Alimi, The modified differential evolution and the RBF (MDE-RBF) neural network for time series prediction, Proc. Int. Joint Conf. Neural Network (2006) 2938–2943.

[76] S. Yang, Y.B. Gan, A. Qing, Sideband suppression in time-modulated linear arrays by the differential evolution algorithm, IEEE Trans. Antenn. Propagations Lett. 1 (1) (2002) 173–175.

[77] H.K. Kim, J.K. Chong, K.Y. Park, D.A. Lowther, Differential evolution strategy for constrained global optimization and application to practical engineering problems, IEEE Trans. Magn. 43 (4) (2007) 1565–1568.

[78] A. Massa, M. Pastorino, A. Randazzo, Optimization of the directivity of a monopulse antenna with a subarray weighting by a hybrid differential evolution method, IEEE Trans. Antenn. Propagations Lett. 5 (1) (2006) 155–158.

[79] C.T. Su, C.S. Lee, Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution, IEEE Trans. Power Deliv. 18 (3) (2003) 1022–1027.

[80] M.F. Tasgetiren, P.N. Suganthan, T.J. Chua, A. Al-Hajri, Differential evolution algorithms for the generalized assignment problem, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09), 2009, pp. 2606–2613.

[81] T. Sum-Im, Taylor, M.R. Irving, Y.H. Song, A differential evolution algorithm for multistage transmission planning, in: Proceedings of the 42nd International Universities Power Engineering Conference (UPEC'07), 2007, pp. 357–364.

[82] S. Bhubaji, A. Kadam, P. Bhumkar, S. Dedge, S. Kanchan, Brain tumor classification (MRI), Kaggle (2020), https://doi.org/10.34740/Kaggle/dsv/1183165.

[83] A.A. Borkowski, M.M. Bui, L.B. Thomas, C.P. Wilson, L.A. DeLand, S.M. Mastorides, Lung and Colon Cancer Histopathological Image Dataset (LC25000), 2019 arxiv:1912.12142v1 [eess.IV].

[84] D.G. Altman, Practical Statistics for Medical Research, Chapman and Hall, New York, 1991.

[85] S. Belciug, Artificial Intelligence in Cancer: Diagnostic to Tailored Treatment, Elsevier, 2020.

[86] B.W. Yap, C.H. Sim, Comparisons of various types of normality tests, J. Stat. Comput. Simulat. 81 (12) (2011) 2141–2155, https://doi.org/10.1080/00949655.2010.520163.

[87] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, 2014 arxiv.org/abs/1409.1556.

[88] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, 2015 arxiv.org/abs/1512.03385.

[89] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, 2015 arxiv.org/abs/1512.00567vol. 3.

[90] G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, 2016. Arxiv.org/abs/1608.06993.

[91] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

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

[93] J. Kang, Z. Ullah, J. Gwak, MRI-Based Brain Tumor Classification using ensemble of deep features and machine learning classifiers, Sensors 21 (6) (2021) 2222, https://doi.org/10.3390/s21062222.

[94] S. Mangal, A. Chaurasia, A. Khajanchi, Convolutional Neural Networks for Diagnosing Colon and Cancer Histopathological Images, 2020 arXiv:2009.03878.

[95] B.K. Hatuwal, H.C. Thapa, Lung cancer detection using convolutional neural network on histophatological images, Int. J. Comput. Trends Technol. 68 (2020) 21–24.

[96] S.U.K. Bukhari, S. Asmara, S.K.A. Bokhari, et al., The histological diagnosis of colonic adenocarcinoma by applying partial self-supervised learning, medRxiv (2020), https://doi.org/10.1101/2020.08.15.20175760.

[97] X.P. Burgos-Artizzu, D. Coronado-Guiterrez, B. Valenzuela-Alcaraz, et al., Evaluation of deep convolutional neural networks for automatic classification of common maternal fetal ultrasound planes, Sci. Rep. 10 (2022) 10200, https://doi.org/10.10138/s41598-020-67076-5.