IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2020.DOI

An Evolutionary DenseRes Deep Convolutional Neural Network for Medical Image Segmentation

TAHEREH HASSANZADEH (IEEE STUDENT MEMBER)[1], DARYL ESSAM[1], and RUHUL SARKER[1].

[1]School of Engineering and Information Technology (SEIT), University of New South Wales, Canberra, Australia (e-mail: t.hassanzadehkoohi@student.unsw.edu.au, d.essam@adfa.edu.au, r.sarker@adfa.edu.au)

Corresponding author: Tahereh Hassanzadeh (e-mail: t.hassanzadehkoohi@student.unsw.edu.au).

ABSTRACT The performance of a Convolutional Neural Network (CNN) highly depends on its architecture and corresponding parameters. Manually designing a CNN is a time–consuming process in regards to the various layers that it can have, and the variety of parameters that must be set up. Increasing the complexity of the network structure by employing various types of connections makes designing a network even more challenging. Evolutionary computation as an optimisation technique can be applied to arrange the CNN layers and/or initiate its parameters automatically or semi–automatically. Dense network and Residual network are two popular network structures that were introduced to facilitate the training of deep networks. In this paper, leveraging the potentials of Dense and Residual blocks, and using the capability of evolutionary computation, we propose an automatic evolutionary model to detect an optimum and accurate network structure and its parameters for medical image segmentation. The proposed evolutionary DenseRes model is employed for segmentation of six publicly available MRI and CT medical datasets. The proposed model obtained high accuracy while employing networks with minimal parameters for the segmentation of medical images and outperformed manual and automatic designed networks, including U–Net, Residual U–Net, Dense U–Net, Non–Bypass Dense, NAS U–Net, AdaresU–Net, and EvoU–Net.

INDEX TERMS Convolutional Neural Network, Dense Network, Evolutionary Computation , Residual Network, Medical Image Segmentation

I. INTRODUCTION

A Deep Convolutional Neural Network (DCNN) is a deep network constructed from several layers, such as convolution layers and pooling layers. Also, in some cases, it can include short and long connections with various connection patterns. Hand–designing a Deep Neural Network (DNN) is a complicated task that involves tremendous efforts and in–depth knowledge in the area of artificial neural networks. Using more parameters and operations to design a network makes this task more complicated. However, using an automatic or semi–automatic method can simplify the process of developing a DCNN. Evolutionary Algorithms [1] and Reinforcement Learning (RL) [2] are two popular methods to establish a neural network and/or initiate its parameters automatically or semi–automatically.

Neuroevolution is using an evolutionary algorithm to generate a network and/or set up its parameters [3]. Several papers investigated the application of Neuroevolution to establish different types of neural networks [4]–[9]. A very early paper in this area was published by Montana et al. [10], where a Genetic Algorithm (GA) [11] is utilised for network weights initialisation. Gradually, other parameters have also been included in the search space to uncover the optimum values for parameters [12]–[14]. Also, the first application of evolutionary computation in a deep network's evolution was introduced by Koutnik et al. [15] in 2014. During the last five years, several papers have been published in the area of using Neuroevolution to design a deep feed–forward network and deep CNN [7].

EvoDeep [16] is a graph–based evolutionary model that was developed to create a deep network structure along with its parameters for image classification. In addition, EvoCNN [17] is another GA–based evolutionary model equipped with a new encoding strategy and an original crossover method for image classification. Since the encoding of networks is a very critical issue that can affect a network's evolution, several techniques have been introduced for network representations. For instance, two different encoding models, including a binary representation and a grammatical representation, have been applied for the evolution of a network for handwriting recognition [18]. Also, a fixed IP–based encoding model and its corresponding variable–length version encoding model were introduced for image classification [19], [20].

The next issue in a network's evolution is a network construction's method, that can be layer by layer or block by block. The evolutionary encoding models that are mentioned above develop a network layer by layer. Neural Architecture Search (NAS) [21] is a cell (block) based model that evolves blocks using a Recurrent Neural Network (RNN) [22]. After finding the block's structures, the corresponding network will be established by stacking the achieved blocks on top of each other. Further, AmoebaNet [23] as an evolutionary version of NAS, has been developed to construct a network based on the obtained blocks. Another block–based evolutionary model was introduced by Fielding et al. [24] to develop a VGG [25] based architecture for image classification. In this model, most of the parameters are kept fixed, such as the number of output feature maps and filter size, although the number of layers in each block is specified using an enhanced version of the Particle Swarm Optimisation (PSO) algorithm [26].

Furthermore, Sun et al. [27] introduced another block–based evolutionary model for image classification using the combination of Dense blocks [28] and Residual blocks [29]. In their proposed model, three different units are established, including Dense Block Unit (DBU), Residual Block Unit (RBU), and Pooling Unit (PU) such that DB and RB units can contain several blocks. Also, some parameters, like filter size, and the number of convolution layers are kept fixed. Since the variable–length encoding strategy is employed for network representation, new crossover and mutation operations are also proposed. AdaResU–Net [30] is another evolutionary model, where a fixed network structure is utilised. However, a number of parameters, including learning rate, dropout probability, the number of filters, activation function, and the filter size of each convolution layer, are specified using a multi–objective evolutionary algorithm. In AdaResU–Net, all the blocks are equipped with residual connections and three convolution layers. Finally, EvoU–Net [31] is another block–based evolutionary method proposed for medical image segmentation. EvoU–Net utilised a GA to determine a network structure and its parameters automatically, where the block's internal structure, shortcut and long connections patterns were also specified during the evolution [31].

As discussed above, in most of the previous works, a part of the parameters or network structure is kept fixed, then an evolutionary or reinforcement technique is applied to set up some aspects of the networks. However, in this paper, we propose a new block–based evolutionary technique to build a network and set up its parameters automatically for medical image segmentation. In regard to the capability of Dense blocks [28] and Residual blocks [29] for feature extraction, our evolutionary DenseRes model is the first automatic U–Net [32] based model that can develop a deep convolutional neural network using a combination of Dense and Residual blocks. It is here introduced for image segmentation. In this paper, a Genetic Algorithm (GA) [11] is used to discover the appropriate network structure, and also the other parameters to create and train a network automatically. In the proposed model, 14 parameters are required to be specified by the GA, including, number of blocks (block's status), number of convolution layers in each block (layer's status), the size of filter for each convolution layer, number of output feature maps, dropout [33] probability, pooling layer, type of activation function, type of the block, long connection, Batch Normalisation (BN) [34], optimiser, learning rate, batch size, and the augmentation size.

In the proposed model, GA is employed to identify all necessary parameter's values to develop a U–Net–based network using a combination of Dense and Residual blocks. Therefore, without having in–depth knowledge related to deep learning, an optimum network structure along with its parameters can be developed for medical image segmentation. The proposed model using limited computation in a relatively short time, can find a small, accurate network for medical image segmentation. The obtained results for MRI prostate segmentation (two datasets), CT liver (two datasets), CT spleen, and MRI brain segmentation, show the capability of our proposed model for developing appropriate networks for various medical image segmentation, and outperformed U–Net [32], Residual U–Net [35], Dense U–Net [36], Non–Bypass Dense [37], NAS U–Net [38], AdaresU–Net [30], and EvoU–Net [31].

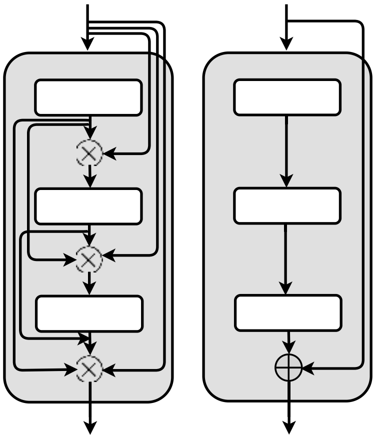
The rest of this paper is organised as follows. Section 2 provides a background concerning the network structures and genetic algorithm that we utilised in our proposed model. Section 3 demonstrates the proposed model. The dataset and experimental results are discussed in section 4. Section 5 provides the discussion and conclusion.

II. BACKGROUND

In our proposed model, a combination of three different network structures, including U–Net, Dense Network, and Residual Network, along with a Genetic Algorithm, are employed to create an evolutionary DenseRes network. In this section, a review of the listed algorithms is provided. Also, the proposed model is applied for medical image segmentation; therefore, a review of the various type of image analysis techniques is also provided in this section.

A. U–NET

A Fully Convolutional Neural Network (FCNN) is a version of CNN that is designed for image segmentation [39]. FCNN is constructed from two parts, of down–sampling (encoding, convolution) for feature extraction and up–sampling (decoding, deconvolution) for segmentation reconstruction. In some networks, there is a specific block named Bottleneck (Bridge) to connect these two parts. In the down–sampling



(a) Dense Block (b) Residual Block

FIGURE 1. An example of Dense block and Residual block.

section, the network undertakes to extract features as it goes from the higher resolution to lower resolution with applying pooling layers [40] while the up–sampling part attempts to reconstruct the coarse–to–fine segmentation with transposed convolution [41]. FCNN utilises an end–to–end method for learning. For 2D images, it uses image–to–image, and for 3D volumes, it applies volume–to–volume supervised learning.

One of the first models that utilised FCNN with 2D convolution for 2D medical semantic image segmentation is U–Net [32]. The original version of U–Net is constructed from three parts. The down–sampling part contains four blocks such that between each pair of blocks, there is a max–pooling layer to select the maximum value of the cluster and to halve the size of the feature maps [40]. In the up–sampling section, between each pair of blocks, there is a convolution layer with 2×2 kernel size to double the size of the output feature maps. Additionally, the Bottleneck block connects the two parts. All the blocks are constructed from two convolution layers followed by a non–linearity and finally, a 1×1 convolution layer is employed in the last layer. Also, to improve the results in this structure, long connections are used for cropping and copying a part of the extracted feature maps from the down–sampling part and concatenating them with the obtained feature maps from the up–sampling section.

B. DENSE NETWORK

A Dense Convolutional Neural Network [28] is a deep network composed of several Dense blocks and each Dense block consists of several layers. In this network, to reuse extracted feature maps, the output feature maps of each layer are concatenated (stacked) to the output of all previous layers in that block, then the concatenated feature maps are sent to the next layer (see Figure 1a). Stacking several feature maps with various quality that are obtained utilising shortcut connections can benefit the network to reuse features, improve the quality of propagation, and also alleviate the vanishing gradients [42] problem [28]. Although this network is introduced for natural image classification, the fully convolutional network structure of this network is also introduced in [43] for colour image segmentation.

C. RESIDUAL NETWORK

Since training deep networks is more difficult [29], [44], using the shortcut and long connections can help to ease the training process of the networks. A Residual Neural Network [29] is a kind of neural network that applies skip connections or shortcut connections to jump over layers. A Skip connection transfers the input feature maps of one block to its outputs. Assigned feature maps will be element–wise summed with the output feature maps such that the number and size of feature maps should the same size (see Figure 1b). Using skip connections can reduce the probability of vanishing gradients [42] during backpropagation [45].

D. GENETIC ALGORITHM

Genetic Algorithm (GA) [11] is a meta–heuristic algorithm that was inspired from natural selection and belongs to the category of evolutionary algorithms. GA, as an optimisation technique, relies on three critical operations, namely selection, crossover, and mutation. Generally, the evolution process in GA commences with an initialisation stage, where a number of chromosomes are created randomly as possible solutions; moreover, each chromosome must be evaluated to specify the quality of the generated solution. To create a new generation, in the selection stage, some chromosomes will be selected and then utilising crossover; the new solutions will be designed using combinations of the parents. In the end, a random alternation by mutation must be applied to a number of chromosomes to increase the exploration ability of the algorithm. Finally, a part of or all of the population in the last generation will be replaced with the new chromosomes to generate a new generation, and this process can be repeated to some specified end point.

E. IMAGE SEGMENTATION

Medical image segmentation is one of the critical medical image analysis techniques, which can be used to find organs, cancers, tumours or any other abnormalities in medical images. Established semi–automated and automated medical image segmentation models can mainly be categorised into four various groups: Atlas–based, Shape–based, Image–based, and Superpixel–based segmentation. Recently, Deep learning–based approaches have achieved state–of–the-art results in image processing and specifically in image segmentation [46]–[49].

The capability of Neural Network (NN), specifically Convolutional Neural Network (CNN), for image analysis convinced researchers to develop new CNN–based structures for medical image analysis [50], [51]. Most of the CNN–based networks use a combination of an image processing technique and CNN for image analysis, such as the combination of Atlas–based model and CNN [52]–[54]. Also, a combination of the Proposal–based technique and CNN was successful for image segmentation [55]. However, recently several CNN–based architectures have been developed [56], such that in some cases using only CNN is good enough for feature extraction and image analysis, such as image segmentation [37], [57] and classification [58].

For example, U–Net [32], is a FCNN–based network that was developed for medical image segmentation without using any pre–processing and post–processing. In some studies, the trial and error approach is applied to find a good network structure for medical image segmentation. For instance, Hassanzadeh et al. [37] tried several types of shortcut connections to find the best one for Prostate MRI segmentation. However, to deal with various and changing datasets, this would be very time consuming and computationally expensive.

F. RESEARCH CONTRIBUTION

- In this paper, we propose the first evolutionary U–Net–based model to construct a network using a combination of Dense blocks and Residual blocks for medical image segmentation.
- To develop a precise network structure, the Genetic algorithm is utilised to uncover the best combination of the Dense blocks and Residual blocks, as well as the internal arrangement of the blocks.
- All the parameters that are needed to create a U–Net–based network, along with network training parameters, are specified using the GA.
- A new block–based encoding model is developed to create blocks and consequently the network. We propose a fixed–length encoding model, to create variable–depth networks that are compatible with standard GA operations.
- Users without in–depth knowledge in the area of artificial neural networks will be capable of establishing an optimum network for medical image segmentation utilising our proposed model.
- The proposed framework is general enough to be applied to the different types of medical images.

III. EVOLUTIONARY DENSERES NETWORK

In this paper, we propose an evolutionary DenseRes model to create a U–Net–based deep convolutional neural network for medical image segmentation. In regards to the difficulty of training a deep network, the combination of Dense blocks and Residual blocks that were developed to ease the training process of the deep networks, can be utilised to alleviate the complexity of the network's training. However, the problem in regard to the numerous of possible combinations of Dense and Residual blocks in a network, while also considering possible arrangements inside of each block, to find a network structure that is more compatible for our specified application is very challenging. The manual design of such a complicated structure is very time consuming and in some cases, even impossible and has large elements of trial and error [37]. To address this problem, we propose a new block–based encoding model to represent the possible solutions to develop a network automatically. In the proposed model, the aim is uncovering the best combination of Dense and Residual blocks; moreover, the arrangement of layers inside the blocks, and also the training parameters of the network using GA.

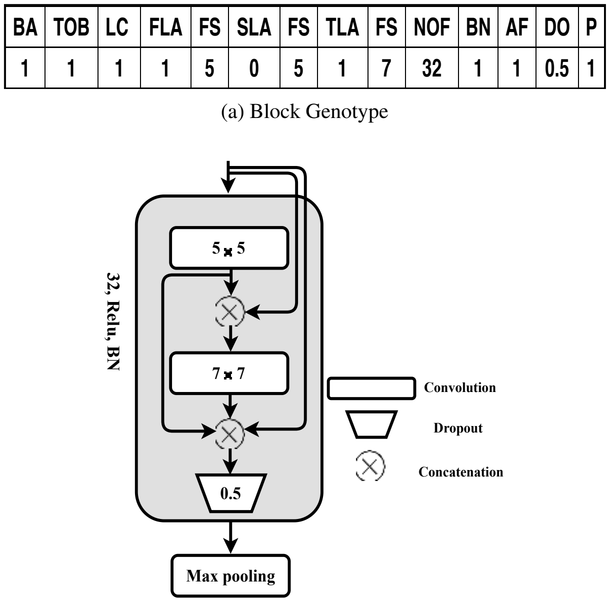
TABLE 1. The Hyper–parameters and their corresponding range to create a network.

Hyper–parameters	Range
Number of blocks	7
Number of convolution layers	$[1 - 3]$
Filter size	$[3 \times 3, 5 \times 5, 7 \times 7]$
Number of filters	$[8, 16, 32, 64]$
Dropout	$[0 - 0.6]$
Pooling	[Averagepooling (0), Maxpooling (1)]
Activation function	[Sigmoid (0), Relu (1)]
Type of block	[Residual (0), Dense(1)]
Long connection	$[0, 1]$
Batch normalisation	$[0, 1]$
Optimiser	[adam, rmsprop, adagrad, adadelta]
Learning rate	$[0.1, 0.01, 0.001]$
Batch size	$[8, 16, 32]$
Augmentation size	$[16000, 32000, 64000]$

First of all, to create a block–based network structure, the number of blocks in the down–sampling section, up–sampling section, and whether there is a bridging block are necessary to determine. As mentioned above, the number of blocks and the block's structures in the encoding and decoding sections are the same in the U–Net–based network. Therefore, having the structures of encoding blocks and bridging block is sufficient to create the whole network. Second, as is discussed above, in the U–Net–based structure after each block there is a pooling layer in the encoding section to halve the size of feature maps, and its corresponding deconvolution layer to double the size of feature maps in the decoding section. Consequently, the maximum number of blocks that can be utilised in each part of the network is related to the size of the input image. For example, six blocks are the maximum number of blocks that can be utilised in the down–sampling and up–sampling sections of a U–Net–based network, plus a bridging block, trained by 128 × 128 input images.

Table 1 provides the list of 14 parameters and their corresponding ranges to create and train a network in our proposed model. As shown in Table 1, the maximum number of blocks in the encoding section plus the bridging block is seven. Also, each block can contain one to three convolution layers. Besides, each convolution layer has its filter size. Consequently, the convolution layers in the block can employ similar or completely different filter sizes. In a DCNN, the size of the receptive field can significantly indicate the percentage of utilisation of context information [59]. Therefore, using filters with different sizes can detect various sized feature maps. The number of filters, dropout probability, pooling, activation function, type of block, long connection, and batch normalisation are set up in a block–based manner. In other words, each of these parameters is applied to the whole block.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.3039496, IEEE Access



BA	TOB	LC	FLA	FS	SLA	FS	TLA	FS	NOF	BN	AF	DO	P
1	1	1	1	5	0	5	1	7	32	1	1	0.5	1

(a) Block Genotype

(b) Block Phenotype

FIGURE 2. An example of a block Genotype and its corresponding Phenotype. The abbreviations in the block's genotype are: BA (Block Activation), TOB (Type Of Block), LC (Long Connection), FLA (First Layer Activation), FS (Filter Size), SLA (Second Layer Activation), TLA (Third Layer Activation), NOF (Number Of Filters), BN (Batch Normalisation), AF (Activation Function), DO (DropOut), and P (Pooling).

For instance, if Batch Normalisation (BN) is active for a block, it means BN will be applied after each convolution layer. Also, when we convert genotype to phenotype, the long connection parameter will be ignored for bridging block and decoding blocks because it is meaningless for them.

In our proposed encoding model, in the initialisation stage, each chromosome corresponds to seven blocks, that is the maximum number of blocks that can be utilised in the encoding and bridging sections. In the proposed encoding model to create networks with various depths, an additional parameter is employed to determine the activation status of each block. Besides, similar activation parameters are used to specify the convolution layer activation status. Since in our encoding model, each convolution layer has its activation parameter, the block is supposed to be inactivated, while all three convolution layers inside the block or the block are inactivated. Therefore, one status parameter plus 13 other parameters are required to construct a block. Since 14 parameters are involved in constructing each block, 98 genes are necessary to represent seven blocks. Then at the end of each chromosome, four more parameters are added to represent the optimiser, learning rate, batch size, and augmentation size for training the network. Overall, each chromosome has a fixed size (102 genes). Still, the output network can have various depths, because a part of the blocks and the convolution layers inside of the blocks might be inactivated in the initialisation stage or during the evolution process. Utilising variable depth networks can increase the chance of finding more precise networks, because the depth of network is a significant parameter on the achievement of a network [17], [25], [60], [61].

An example genotype of a block (see Figure 2a) and its corresponding phenotype (see Figure 2b) are provided in Figure 2. As is shown in Figure 2a, the first parameter shows that the block is active (1) or not (0). A block will be excluded from its network if either its activation parameter is inactivated, or all three layers inside of the block are inactivated. The active blocks among the first six blocks will be considered as the encoding and decoding blocks and the last block (while it is active) as the bridging block. Therefore, the proposed encoding model is capable of establishing networks with or without a bridging block. Furthermore, if only one block is activated, then the output network contains two blocks (one encoding and one decoding block). In the provided example (see Figure 2a), considering the block is active, and there are two active layers inside the block, this block will be included in its network structure. Moreover, it is a Dense block; therefore, the output of each layer must be concatenated with output feature maps of the previous layers. There is also a long connection between this block and its corresponding block in the decoding section. The size of filters for each convolution layer is stated, and other parameters are set up for the block. The phenotype of this block is also presented in Figure 2b. All active blocks will be created and stacked on top of each other, according to the provided example. The encoding blocks will be utilised to develop decoding blocks; however, the differences are the pooling layer will be replaced with deconvolution layers, and the long connections will be ignored in the decoding section.

It needs to be noted that the element–wise sum in the Residual blocks can just be applied on the same number of feature maps. Therefore, to convert the Residual block's genotype to its corresponding phenotype, an additional 1×1 convolution layer needs to be applied to size up the feature maps, while the unequal number of features comes up. For instance, a block has 32 input feature maps; however, the output is 16 feature maps; then, a 1×1 convolution will be used to increase the number of output feature maps to 32.

After converting the genotypes to their corresponding phenotypes, each network will be trained up to predefined epochs. The best network, which achieved the maximum segmentation accuracy, will be transferred to the next generation directly. It needs to be noted that, our proposed model is a single objective model; therefore, each individual is evaluated based on its accuracy. The aim is using an evolutionary process to find networks with the highest segmentation accuracy. Then, to create the rest of the population, the Roulette Wheel selection [62] method is applied to select two best parents. However, to decrease the variance of the selection process, a normalised fitness rather than absolute fitness is utilised. The fitness normalisation process is demonstrated in the Algorithm 2.

The combination of those two selected networks by a single–point crossover will generate two new networks. Lastly, a random number ranging from zero to three, determines the number of mutation of each genotype, which is a random change based on the gene's valid range. After G generations, the best–generated networks will be selected to

```
Data: Training datasets ($D$), Label datasets ($L$)
Result: The ten best evolutionary networks.
Initialise Number of Generations ($G$), Length of Block ($LB$), Number of
  Epochs ($E$), Initial Population ($I$), Population ($pop$), Next generation ($N$);
$members = Generate\_Random\_Population(I)$;
$pop = evaluate\_population(members)$;
for every Generation do
    $members = reproduce(pop)$;
    $pop = evaluate\_population(members)$;
end
$Ten\_Best \leftarrow Ten\_Best(pop)$;
Return $Ten\_Best$;
Function $Generate\_Random\_Population$ ($I$) :
    $members \leftarrow [null]$;
    for $i \leq I$ do
        $genome \leftarrow [null]$;
        for every Block do
            for $key \leq LB$ do
                $param = block\_params[key]$;
                $genome+ = random\_choice(param)$;
            end
        end
        $genome+ = random\_choice(optimiser)$;
        $genome+ = random\_choice(learning\_rate)$;
        $genome+ = random\_choice(batch\_size)$;
        $genome+ = random\_choice(augmentation\_size)$;
    end
    $members[i] = genome$;
    return members;
Function $Evaluate\_Population$ (members) :
    $fit \leftarrow [null]$;
    for $i \leq I$ do
        $Network \leftarrow Decode(members[i])$ ;
        // Convert genotype to phenotype
        $loss, accuracy \leftarrow Train(Network(E, D, L))$;
        $fit[i] = accuracy$;
    end
    $pop \leftarrow (members, fit)$;
    return $pop$;
Function $Reproduce$ ($pop$) :
    $members \leftarrow [null]$;
    if $G \leq 1$ then
        $N \leftarrow I$;
    else
        $N \leftarrow pop$;
    end
    for $i \leq len(N) * 0.95$ do
        $members[i] = Crossover(pop.select(), pop.select())$;
    end
    $members[len(N)] = get\_best()$;
    for $i \leq len(N)$ do
        $num\_mutation = random(0, 3)$;
        $members[i] =$
          $Mutation(members[i], num\_mutation)$;
    end
    return $members$;
```

Algorithm 1: The pseudo code of the proposed evolutionary model.

```
Data: Absolute Fitness
Result: Normalised Fitness
$scores \leftarrow fitness - fitness.\min()$;
if $scores.\max() \geq 0$ then
    $scores \leftarrow \frac{scores}{scores.\max()}$;
else
    $scores \leftarrow scores$;
end
$Sum \leftarrow \sum(scores)$;
```

Algorithm 2: Fitness normalisation before roulette wheel selection.

train for more epochs to uncover the best network for image segmentation.

The pseudocode of the proposed model is presented in Algorithm 1. The evolutionary process starts with the initialisation stage in which a random population is created using *Generate_Random_Population* function. To create a chromosome, first the block's parameters initialised randomly, and then the network's training parameters (including optimiser, learning_rate, batch_size, and augmentation_size) are also added at the end of it. This process needs to be repeated to create the whole population (I). In the evaluation stage, using *evaluate_population* function, each genotype is converted to its corresponding phenotype and network is trained up to the specified epochs. Finally, to reproduce a new population (using *Reproduce* function), the best network (network with the highest segmentation accuracy) is sent to the next generation directly, and to create the rest of the population, selection, crossover and mutation are also employed.

TABLE 2. The list of datasets and the number of image slices and volumes in train, test, and validation sets.

Dataset	Train	Test	Validation
	Volumes/Slices	Volumes/Slices	Volumes/Slices
Promise12	40/1115	5/165	5/97
CHAOS	16/2025	4/419	4/360
SLIVER07	12/2712	4/538	4/909
Prostate	22/408	5/94	5/100
Spleen	29/2203	6/619	6/491
BraTs	233/36115	51/7905	51/7905

IV. EXPERIMENTS

A. DATASET

The Promise12 dataset [48] for prostate MRI segmentation, Combined Healthy Abdominal Organ Segmentation (CHAOS) dataset [63], and segmentation of the liver competition 2007 (SLIVER07) dataset [64] for liver CT segmentation, two MRI prostate and CT spleen segmentation datasets from Decathlon challenge [65] and also, BraTs2019 dataset [66]–[68], for brain tumour (whole tumour) segmentation were selected for evaluating the proposed model. Detailed information about the six datasets is provided in Table 2. Since Promise12 was collected from various centres with different qualities, a Z–score [69] normalisation model was applied on its images. However, the rest of the datasets were utilised without any pre–processing. All the images were resized to 128×128 for training the networks and evaluation. Since the number of images is limited, for all six datasets the number of training images was augmented to a specified number using various types of augmentation techniques, including, rotation, zooming, vertical and horizontal flips, and elastic transformation [70]. In the proposed model, all image slices, regardless of whether they included Region Of Interest (ROI) or not, were used for training, validation and testing of the obtained networks.

B. IMPLEMENTATION

The proposed model was implemented using the Keras python package [71]. All experiments were carried out on one Nvidia GPU. In regards to the six datasets that we utilised

TABLE 3. The parameters and their corresponding values for training the models.

Training parameters	Range
Number of generations	9
Number of epochs	5
Number of runs	10
Early stopping	3
Initial population size	60
Population size	30

for evaluation, the proposed model trained using each dataset separately according to the provided information in Table 3. The number of generations and population size were set up based on preliminary experiments. In the initialisation stage, 60 networks were established randomly as the possible solutions. This is a relatively large population size in this application. The larger population size can increase the diversity of solutions, and consequently the chance of discovering a precise network. Then the size of the population is halved in the second generation and training is continued to nine generations. Networks were trained for up to five epochs in the training stage to save time. Also, to show the consistency of the proposed model, each model was run ten times. For evaluating the evolved networks, Dice Coefficient (DSC) [72] is employed (see equation 1). Also, DSC is used as a fitness function to evaluate the networks during evolution. Where, $Y^{'}$ shows the label image, Y represents predicted segmented image, and $\mid Y^{'} \mid$ and $\mid Y \mid$ indicates the cardinality of $Y^{'}$ and Y. Moreover, in the proposed model, DSC was exploited as the loss function for training the networks.

$$DSC = \frac{2 \mid Y^{'} \cap Y \mid}{\mid Y^{'} \mid + \mid Y \mid} \tag{1}$$

Finally to compare the proposed model to previous works, we also employ: Hausdorff distance [64], Average Boundary Distance (ABD) [64], absolute Relative Volume Difference (aRVD) [73], and Area Under ROC (Receiver Operating Characteristic) Curve (AUC) [74] (see equation 2), where FP shows False Positive, TN represents True Negative, FN demonstrates False Negative, and TP states for True Positive.

$$AUC = 1 - \frac{1}{2}(\frac{FP}{FP+TN} + \frac{FN}{FN+TP})$$

(2)

C. EXPERIMENTAL RESULTS

As mentioned above for each dataset, we repeated the proposed evolutionary process ten times. In the end, based on the obtained results after nine generations, the best run was selected. Consequently, the ten best networks from the specified sequence were selected as the best networks for final segmentation. However, the best networks were trained for just five epochs; therefore, they need to be trained to more epochs. Since the network's weight initialisation is random, and it is effective on the final results, to find the best network structure, each network was trained five times to 25 epochs. In the end, the best results were selected for each network. The obtained results after five and then 25 epochs for each dataset are in Table 4. As can be seen from Table 4, our proposed model found ten networks corresponding to each dataset with high accuracy, such that for the MRI prostate segmentation, the best network obtain 0.902 DSC, and for the CT liver segmentation 0.921 and 0.943 respectively for CHAOS and SLIVER07 datasets. Besides, for Decathlon spleen and prostate datasets, the obtained DSC's are 0.95 and 0.92 respectively. Finally, our proposed model obtained 0.767 DSC for the whole brain tumour segmentation. These results show the capability of the proposed framework to find networks with high accuracy for datasets with various features.

TABLE 4. The best DSCs of the ten best networks after training to five epochs during the evolution, and then to 25 epochs to find final segmentation.

Net	Promise12		CHAOS		SLIVER07		Spleen		Prostate		BraTs	
	5	25	5	25	5	25	5	25	5	25	5	25
N 1	0.765	0.880	0.858	0.910	0.832	0.874	0.888	0.932	0.887	0.91	0.579	**0.767**
N 2	0.741	0.873	0.778	0.782	0.814	0.920	0.875	0.841	0.885	0.886	0.567	0.713
N 3	0.738	**0.902**	0.771	0.882	0.802	0.835	0.849	0.910	0.881	0.882	0.520	0.681
N 4	0.699	0.884	0.749	0.875	0.793	0.910	0.837	0.925	0.881	0.875	0.512	0.556
N 5	0.670	0.852	0.720	0.889	0.792	0.931	0.833	0.853	0.875	0.875	0.510	0.522
N 6	0.629	0.837	0.685	**0.921**	0.786	0.870	0.818	0.914	0.873	**0.920**	0.505	0.700
N 7	0.622	0.856	0.571	0.850	0.785	0.893	0.815	**0.950**	0.870	0.850	0.477	0.514
N 8	0.619	0.837	0.517	0.92	0.771	**0.943**	0.815	0.886	0.860	0.910	0.471	0.521
N 9	0.602	0.882	0.490	0.770	0.769	0.897	0.812	0.935	0.858	0.870	0.465	0.501
N 10	0.602	0.870	0.468	0.780	0.768	0.890	0.777	0.738	0.856	0.860	0.462	0.514
Std	0.062	0.084	0.143	0.059	0.020	0.031	0.032	0.063	0.011	0.022	0.033	0.089

D. BEST NETWORK'S STRUCTURES

In this section, the genotype of the best six network structures corresponding to each dataset is presented in Table 5. As can be seen from Table 5, the evolution process finds a unique network structure along with its training parameters for each dataset. As can be seen from Table 5, each genotype shows the block structures in the down–sampling and bridging sections. For example, the genotype of the first network structure indicates that there are six active blocks such that five of the active blocks are in the down–sampling, five blocks in the up–sampling and a bridging block. However, the third network (SLIVER07) has no bridging block and contains three blocks on each side of the network. Besides, there is, an exception in the Spleen best network, the third block of the Spleen dataset is active, but all three convolution layers inside of the block are deactivated; therefore, this block should be excluded from the network's phenotype. Also, each network has its training parameters. The type of the optimiser, learning rate, augmentation and batch size are specified during the evolution of each network.

E. CROSS–VALIDATION

In this section, to show the capability of our proposed model, we have applied four–fold cross–validation for evaluation of the SLIVER07 dataset. Its data was partitioned to four–fold according to Table 6. As can be seen from Table 6, for each fold the image slices corresponding to four volumes were considered for the test, four volumes for validation and the rest for the training of each fold. Then we trained each fold using the proposed evolutionary model and obtained the ten

TABLE 5. The genotypes of the best network structures corresponding to each dataset.

Dataset	Block	BA	TOB	LC	FLA	FS	SLA	FS	TLA	FS	NOF	BN	AF	DO	P	Training Parameters
Promise12	Block 1	1	1	1	0	5	1	7	1	3	16	1	0	0.1	0	Optimiser = Adadelta Learning Rate = 0.1 Batch Size = 8 Augmentation= 64000
	Block 2	0	0	1	0	7	0	3	0	5	8	1	0	0.4	0	
	Block 3	1	1	0	1	7	0	5	1	3	16	0	1	0.0	0	
	Block 4	1	1	1	0	7	1	7	1	7	64	1	0	0.0	1	
	Block 5	1	1	1	0	5	1	7	0	5	64	1	0	0.5	0	
	Block 6	1	1	0	1	7	1	7	1	5	32	0	1	0.0	1	
	Bridge	1	0	1	0	3	1	3	1	7	16	1	1	0.1	0	
CHOAS	Block 1	1	0	1	1	5	1	7	0	3	16	1	0	0.6	0	Optimiser = Adadelta Learning Rate = 0.01 Batch Size = 16 Augmentation= 16000
	Block 2	1	1	1	1	3	0	3	0	5	16	0	1	0.1	0	
	Block 3	1	1	0	0	5	1	7	0	3	16	1	1	0.0	1	
	Block 4	1	0	1	1	7	0	5	0	7	16	1	0	0.3	1	
	Block 5	0	1	1	1	3	0	7	0	5	16	1	0	0.3	1	
	Block 6	1	0	0	0	3	0	3	1	7	32	1	0	0.0	0	
	Bridge	1	0	1	1	7	1	3	0	7	8	1	0	0.0	1	
SLIVER07	Block 1	1	0	1	0	7	1	7	1	3	8	1	1	0.2	0	Optimiser = Adam Learning Rate = 0.001 Batch Size = 16 Augmentation= 32000
	Block 2	1	1	1	1	5	1	7	1	3	64	0	1	0.5	0	
	Block 3	0	0	0	1	3	0	3	0	5	16	1	0	0.6	0	
	Block 4	0	0	0	0	3	1	3	1	7	8	1	0	0.4	1	
	Block 5	1	0	0	0	5	1	5	1	5	32	1	1	0.1	0	
	Block 6	0	1	0	0	5	1	3	0	7	32	1	0	0.1	1	
	Bridge	0	0	0	1	5	1	7	1	3	32	1	0	0.2	1	
Spleen	Block 1	1	0	1	1	3	1	3	1	5	16	1	1	0.6	0	Optimiser = Rmsprop Learning Rate = 0.001 Batch Size = 16 Augmentation= 32000
	Block 2	1	0	0	1	3	1	3	1	3	32	1	1	0.6	0	
	Block 3	1	1	0	0	5	0	5	0	5	8	1	0	0.1	0	
	Block 4	1	0	0	1	3	0	7	1	7	8	1	1	0.1	0	
	Block 5	1	1	1	1	3	1	3	1	7	32	1	1	0.2	0	
	Block 6	0	0	1	1	5	1	5	1	5	64	1	0	0	0	
	Bridge	0	1	0	0	3	0	3	1	5	64	0	0	0.1	1	
Prostate	Block 1	1	1	0	0	7	1	3	0	3	64	1	0	0.1	1	Optimiser = Rmsprop Learning Rate = 0.001 Batch Size = 32 Augmentation= 64000
	Block 2	1	0	0	0	7	0	3	1	3	32	1	1	0.1	1	
	Block 3	0	0	1	1	3	1	3	0	3	32	0	1	0.1	1	
	Block 4	1	1	1	1	5	1	7	1	7	32	1	1	0.2	1	
	Block 5	0	0	1	1	3	1	5	0	5	32	0	0	0.6	0	
	Block 6	0	0	1	0	7	1	7	1	5	8	1	0	0.6	0	
	Bridge	1	1	1	1	7	0	5	0	3	8	1	0	0.6	0	
BraTs	Block 1	1	1	0	1	7	1	3	0	5	32	0	1	0.4	0	Optimiser = Adadelta Learning Rate = 0.1 Batch Size = 32 Augmentation= 64000
	Block 2	1	1	1	0	3	0	7	1	7	8	0	1	0.1	0	
	Block 3	0	0	0	1	7	1	7	0	7	64	0	0	0.4	1	
	Block 4	1	0	0	0	7	1	7	1	7	32	1	1	0.2	0	
	Block 5	0	0	1	0	5	0	5	0	3	32	1	0	0.1	0	
	Block 6	1	0	1	1	5	1	7	0	3	64	1	0	0.1	0	
	Bridge	1	0	1	0	3	1	5	0	3	32	0	0	0.6	1	

TABLE 6. Data partitioning for four–fold cross–validation using SLIVER07 dataset.

Folds	Train		Test		Validation	
	Slices	Volumes	Slices	Volumes	Slices	Volumes
Fold1	2712	12	538	4	909	4
Fold2	2241	12	909	4	1009	4
Fold3	2331	12	1009	4	819	4
Fold4	2456	12	819	4	884	4

best networks regarding each fold. Train and validation sets were used for training and evolving the networks. Such that networks with the highest validation accuracy have higher chance to be selected for the next generation. In the end, the test set is used for the evaluation of the best networks.

The obtained DSCs of the test set for the ten best networks regarding each fold is provided in Table 7. As can be seen from Table 7, our proposed model found ten networks for each fold with high accuracy. This shows, despite the limited amount of data for training of each fold, our proposed model can still find networks with high segmentation accuracy.

F. COMPARISON WITH PRIOR WORK

The comparison of the obtained results with previous works is also reported in Table 8. In this section, we compare the obtained results versus U–Net [32], Dense U–Net [36], Res U–Net [35], and Non–Bypass Dense [37] which are all manually designed networks; also, AdaResU–Net [30] and EvoU–Net [31] that are evolutionary networks and finally with NAS U–Net [38] which was developed using reinforcement learning. As shown in Table 8, in all six datasets, our proposed DenseRes evolutionary models obtained the best accuracy for image segmentation. Since each network was developed for a specific dataset, the segmentation performance considerably improved when compared to prior networks. We also compared the networks in terms of their size (the number of trainable parameters). The results show that for each model, the obtained networks are usually far smaller than the others while achieving high accuracy. Although the DenseRes model for segmentation of the prostate MRI dataset used more trainable parameters in comparison to AdaresU–Net and EvoU–Net; however it obtains better accuracy for the segmentation.

TABLE 7. The obtained DSCs for Four–fold cross–validation using SLIVER07 dataset after five and then 25 epochs.

Networks	Forld-1		Fold-2		Fold-3		Fold-4	
	5	25	5	25	5	25	5	25
Net 1	0.832	0.874	0.843	**0.927**	0.844	0.897	0.890	0.932
Net 2	0.814	0.920	0.839	0.901	0.829	**0.921**	0.883	0.915
Net 3	0.802	0.835	0.800	0.888	0.820	0.887	0.867	**0.944**
Net 4	0.793	0.910	0.787	0.903	0.801	0.901	0.866	0.897
Net 5	0.792	0.931	0.776	0.857	0.798	0.874	0.853	0.899
Net 6	0.786	0.870	0.774	0.890	0.780	0.895	0.843	0.912
Net 7	0.785	0.893	0.764	0.850	0.778	0.837	0.837	0.873
Net 8	0.771	**0.943**	0.763	0.876	0.763	0.865	0.836	0.924
Net 9	0.769	0.897	0.754	0.849	0.740	0.874	0.830	0.899
Net 10	0.768	0.890	0.751	0.879	0.736	0.863	0.819	0.901
Std	0.020	0.031	0.032	0.025	0.036	0.036	0.023	0.020

Also, we compared the obtained results versus previous work using AUC [74] (see Table 9). AUC calculated for image slices that contained ROI based on the following rules.

- **FP** : The number of background pixels predicted as ROI pixel.
- **FN** : The number of ROI pixels predicted as a background pixel.
- **TN** : The number of background pixels recognised as background.
- **TP** : The number of ROI pixels recognised as ROI pixels.

As can be seen from Table 9, minimum, maximum, average, and standard deviation of the obtained AUC regarding the test images for each dataset are reported. As shown, again, the proposed model outperformed previous works. This shows the capability of the proposed framework for medical image segmentation.

Besides, as a subjective comparison, we provide one sample image from each dataset and their corresponding predicted segmentation using various models separately in Figure 3. As can be seen from Figure 3, the accuracy of segmentation is different, and some of them suffer from over–segmentation or under segmentation. However, in some cases, the predicted segmentation is similar because some of the methods obtained competitive results. Also, we provide five more sample images with various shapes and sizes from each dataset in Figure 4. Each column shows five segmented sample images from each dataset. In all six datasets, the proposed DenseRes evolutionary model segments the Region Of Interest (ROI) precisely. Especially for liver and spleen segmentation, where the colour and texture of the liver and spleen are similar to surrounding organs, previous work suffered from over or under segmentation; however our proposed models find the ROI with high accuracy and are much more precise than the others.

G. TIME ANALYSIS

In this section, we have compared the training times of our proposed model against existing evolutionary and non–evolutionary CNN based models. Table 10 shows a number of

TABLE 8. Comparison of the proposed DenseRes network with previous work.

Models	Promise				CHAOS	SLIVER07	Spleen	Prostate	BraTs	Trainable Parameters
	DSC±std	HD±std	ABD±std	RVD±std	DSC±std	DSC±std	DSC±std	DSC±std	DSC±std	
U–Net [32]	0.785 ± 0.012	20.567 ± 0.021	2.982 ± 0.022	20.98 ± 0.08	0.630 ± 0.055	0.700 ± 0.095	0.740 ± 0.088	0.730 ± 0.019	0.532 ± 0.099	31.0×10^6
Dense U–Net [36]	0.860 ± 0.088	17.85 ± 0.077	2.60 ± 0.032	13.20 ± 0.054	0.872 ± 0.032	0.850 ± 0.091	0.820 ± 0.052	0.837 ± 0.082	0.514 ± 0.031	15.48×10^6
Res U–Net [35]	0.840 ± 0.041	19.27 ± 0.099	4.25 ± 0.011	9.97 ± 0.076	0.775 ± 0.009	0.840 ± 0.043	0.780 ± 0.009	0.820 ± 0.052	0.532 ± 0.077	4.04×10^6
Non–Bypass Dense [37]	0.873 ± 0.042	16.13 ± 0.004	2.72 ± 0.007	14.13 ± 0.007	0.861 ± 0.074	0.924 ± 0.041	0.897 ± 0.013	0.877 ± 0.081	0.616 ± 0.012	200.0×10^6
NAS U–Net [38]	0.863 ± 0.064	20.12 ± 0.087	3.20 ± 0.008	15.12 ± 0.004	0.832 ± 0.097	0.930 ± 0.021	0.890 ± 0.006	0.821 ± 0.043	0.653 ± 0.005	30.0×10^6
AdaResU–Net [30]	0.842 ± 0.099	22.227 ± 0.005	2.500 ± 0.043	18.33 ± 0.006	0.843 ± 0.075	0.893 ± 0.043	0.900 ± 0.032	0.846 ± 0.009	0.687 ± 0.043	$4.6 \times 10^6, 5.1 \times 10^6, 4.9 \times 10^6$ $3.0 \times 10^6, 1.59 \times 10^6, 3.2 \times 10^6,$
EvoU–Net [31]	0.893 ± 0.065	17.918 ± 0.043	**1.667** ± 0.004	8.98 ± 0.008	0.881 ± 0.076	0.90 ± 0.021	0.90 ± 0.008	0.854 ± 0.081	0.712 ± 0.098	1.76×10^6 , 2.86×10^6 , 1.9×10^6 $3.5 \times 10^5, 2.19 \times 10^6, 2.5 \times 10^6$
DenseRes U–Net	**0.902** ± 0.006	**16.06** ± 0.043	2.13 ± 0.004	**7.35** ± 0.005	**0.921** ± 0.011	**0.943** ± 0.003	**0.950** ± 0.032	**0.920** ± 0.012	**0.767** ± 0.008	6×10^6 , $3.6 \times 10^4, 2.35 \times 10^6$ $7.5 \times 10^5, 1.59 \times 10^6, 1.3 \times 10^6$

TABLE 9. Comparison of the proposed DenseRes network with previous work based on AUC.

Models	Promise				CHAOS				SLIVER07				Prostate				Spleen				BraTs			
	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD	Min	Max	Mean	STD
U-Net [32]	0.5	0.915	0.812	0.15	0.5	0.95	0.725	0.19	0.471	0.955	0.727	0.163	0.487	0.997	0.848	0.171	0.495	0.991	0.798	0.187	0.458	0.995	0.71	0.177
Dense U-Net [36]	0.497	0.995	0.859	0.13	0.482	0.987	0.843	0.18	0.487	0.994	0.799	0.191	0.497	0.997	0.911	**0.134**	0.494	0.997	0.784	0.195	0.49	0.995	0.668	0.181
Res U-Net [35]	0.5	0.996	0.891	0.135	0.5	0.977	0.706	0.185	0.484	0.997	0.759	0.203	0.495	0.998	0.855	0.162	0.498	0.997	0.829	0.185	0.486	0.989	0.642	0.178
Non–Bypass Dense [37]	0.5	0.998	0.884	0.14	0.5	0.964	0.871	0.18	0.487	0.998	0.954	0.147	0.487	0.997	0.887	0.171	0.497	0.934	0.944	0.134	0.491	0.993	0.762	0.19
NAS U-Net [38]	0.5	0.998	0.86	0.147	0.5	0.951	0.735	0.16	0.5	0.999	0.923	**0.133**	0.499	0.995	0.803	0.173	0.499	0.999	0.914	0.142	0.45	0.994	0.721	**0.176**
AdaRes U-Net [30]	0.5	0.995	0.836	0.132	0.5	0.975	0.715	0.171	0.495	0.995	0.80	0.19	0.497	0.997	0.914	0.118	0.5	0.999	0.909	0.147	0.453	0.994	0.657	0.19
EvoU-Net [31]	0.5	0.990	0.886	0.131	0.492	0.992	0.877	0.153	0.499	0.998	0.917	0.14	0.493	0.996	0.880	0.15	0.5	0.999	0.927	**0.128**	0.461	0.996	0.766	0.199
DenseRes U-Net	0.499	0.989	**0.897**	**0.118**	0.5	0.982	**0.887**	**0.138**	0.5	0.995	**0.962**	0.166	0.499	0.997	**0.924**	0.165	0.499	0.999	**0.993**	0.129	0.453	0.994	**0.777**	0.19

existing automatic models that use Reinforcement Learning (RL) and the evolutionary concept for image classification or segmentation. As can be seen from Table 10, we compared our proposed model with eight different automatic search algorithms for training time. NAS [21], NASNet [75], AmoebaNet [23], and Automatically Designing CNN [76] need tens of GPU days to find a network for image classification. NAS [21] utilised 800 GPUs which took 28 days for training. While NASNet [75] using 500 GPUs and needed four days for training. Besides, AmoebaNet [23] is an evolutionary model ran on 450 GPUs for seven days. Furthermore, Automatically Designing CNN [76] used 15 GPUs took 10 days for training. The reported GPU days have been taken from their source papers. Since these papers were published at different years in the past, the time comparisons may not be accurate due to changes in technology over time.

Also, Adversarial NAS [77] is another RL–based technique that applied RL just to find the structure of the discriminator of the network which takes just six GPU days. Besides, NAS U–Net [38], need two days to find a network, however, in this method, natural images are used to find the network, and then medical images are used for network evaluation. AdaResU–Net [30] and EvoU–Ne [31] are two evolutionary–based model to find a network structure, such that AdaResU–Net uses a fixed network structure (the number of blocks and layers inside of the blocks are kept fixed) and an evolutionary technique is just applied to find some hyper–parameters. Finally, our proposed model needs two and a half to four days for training using one GPU for various datasets. Despite developing the whole network along with its training parameters using an evolutionary model, our proposed model is still among the models using minimum time for training.

H. PROPOSED DENSERES MODEL FOR 3D SEGMENTATION

The proposed DenseRes framework is a 2D evolutionary segmentation model. Developing a 3D evolutionary model is computationally expensive because we need to evolve a group of 3D networks to the specified generation, which requires a massive amount of computation and time. The advantage of using 3D convolution for 3D image analysis is using the contextual information for segmentation, which can increase the accuracy of segmentation. In this section, to show the capability of the proposed model, we also analysed the obtained 2D models for 3D image segmentation. To do this, we stack all the segmented 2D image slices on top of each other to create a 3D volume. Then, we calculate the volumetric DSC of the obtained volumes and compare the results with three previous 3D manually designed CNN networks: 3D U–Net [78], Convnet [79], and 3D Dense U–Net [36] (see Table 11). We have used the same data for training both 2D and 3D models, however for training the 2D models, we used image slices, while for 3D models the corresponding 3D volumes were used.

TABLE 10. Comparison of the proposed model versus previous work in terms of the cost.

Architecture	Search Cost (GPU days)	Search Method	Application
NAS [21]	22400	RL	Classification
NASNet [75]	2000	RL	Classification
AmoebaNet [23]	3150	Evolution	Classification
Adversarial NAS [77]	6	RL	Segmentation
NAS U–Net [38]	2	RL	Segmentation
Automatically Designing CNN [76]	150	RL	Segmentation
AdaResU–Net [30]	2.5–4.5	Evolution	Segmentation
EvoU–Ne [31]	12–38	Evolution	Segmentation
Proposed model	2.5–4	Evolution	Segmentation

TABLE 11. Comparison of the proposed model versus prior work based on obtained DSC for 3D image segmentation.

Models	Promise	CHAOS	SLIVER07	Spleen	Prostate	BraTs
	DSC±std	DSC±std	DSC±std	DSC±std	DSC±std	DSC±std
3D U–Net [78]	0.833 ± 0.043	0.883 ± 0.021	0.885 ± 0.021	0.886 ± 0.071	**0.858** ± 0.009	0.691 ± 0.08
Convnet [79]	0.779 ± 0.019	0.381 ± 0.072	0.898 ± 0.006	0.90 ± 0.0004	0.855 ± 0.016	0.572 ± 0.045
3D Dense U–Net [36]	0.721 ± 0.065	0.328 ± 0.016	0.826 ± 0.007	0.778 ± 0.009	0.817 ± 0.054	0.564 ± 0.031
DenseRes	**0.875** ± 0.004	**0.917** ± 0.011	**0.941** ± 0.004	**0.9387** ± 0.005	0.833 ± 0.052	**0.701** ± 0.043

As can be seen from Table 11, we used three previous 3D models for the segmentation of the six various datasets. To train previous works, the number of training images was augmented up to 32000 and with a batch size of 16. Their training parameters were set up based on their source papers. Since all the previous models are manually designed networks for a specific segmentation application, in some cases they obtained high accuracy. In contrast, in other instances such as CHAOS dataset segmentation, we can see low efficiency using ConvNet and 3D Dense U–Net. Inter-

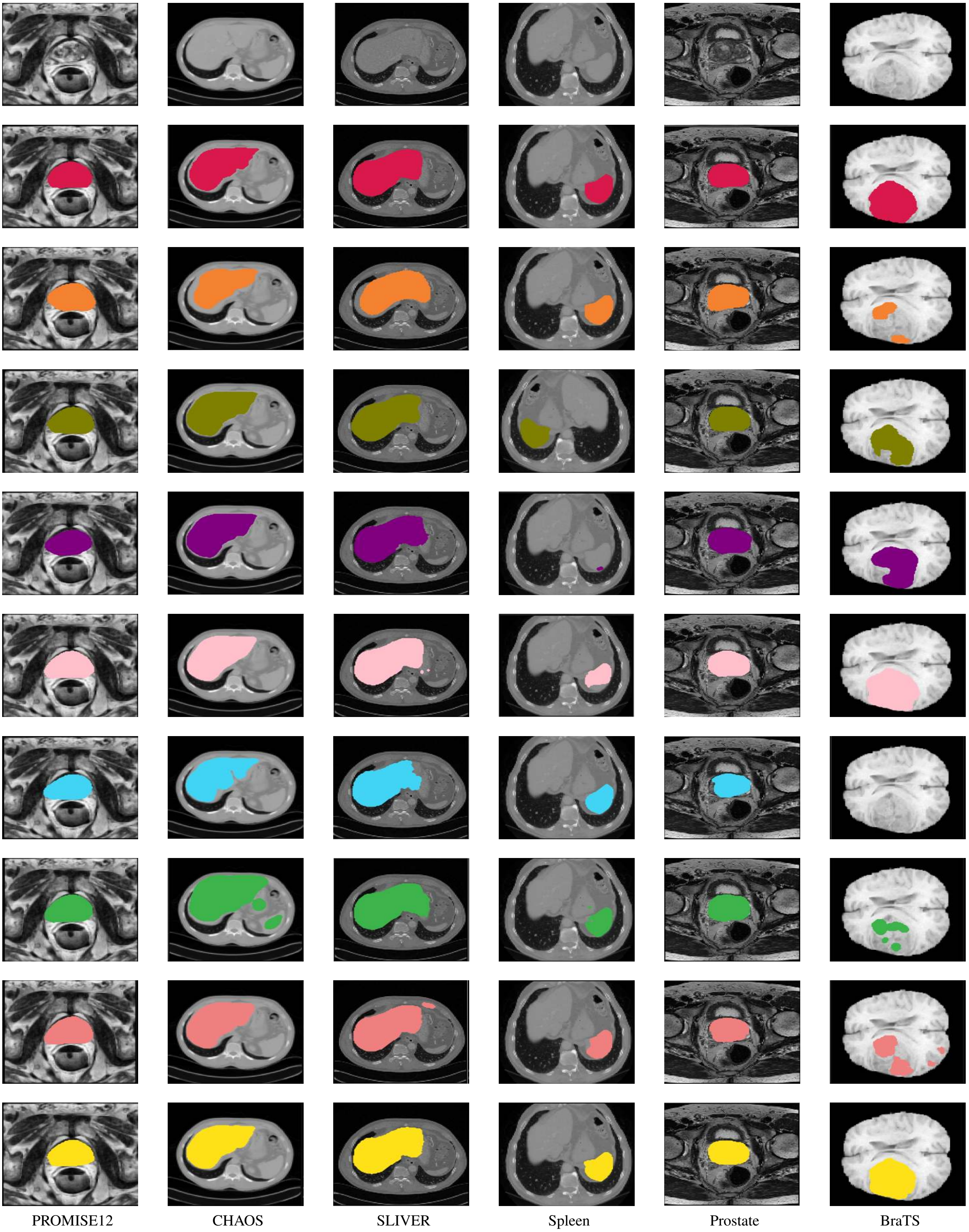


FIGURE 3. One sample segmented image from each dataset using various methods. The red contour is the ground truth, orange is U–Net, olive is Non–Bypass Dense, purple is AdaRes U–Net, pink is EvoU–Net, cyan is Res U–Net, green is Dense U–Net, light coral is NAS U–Net, and yellow is proposed DenseRes model.

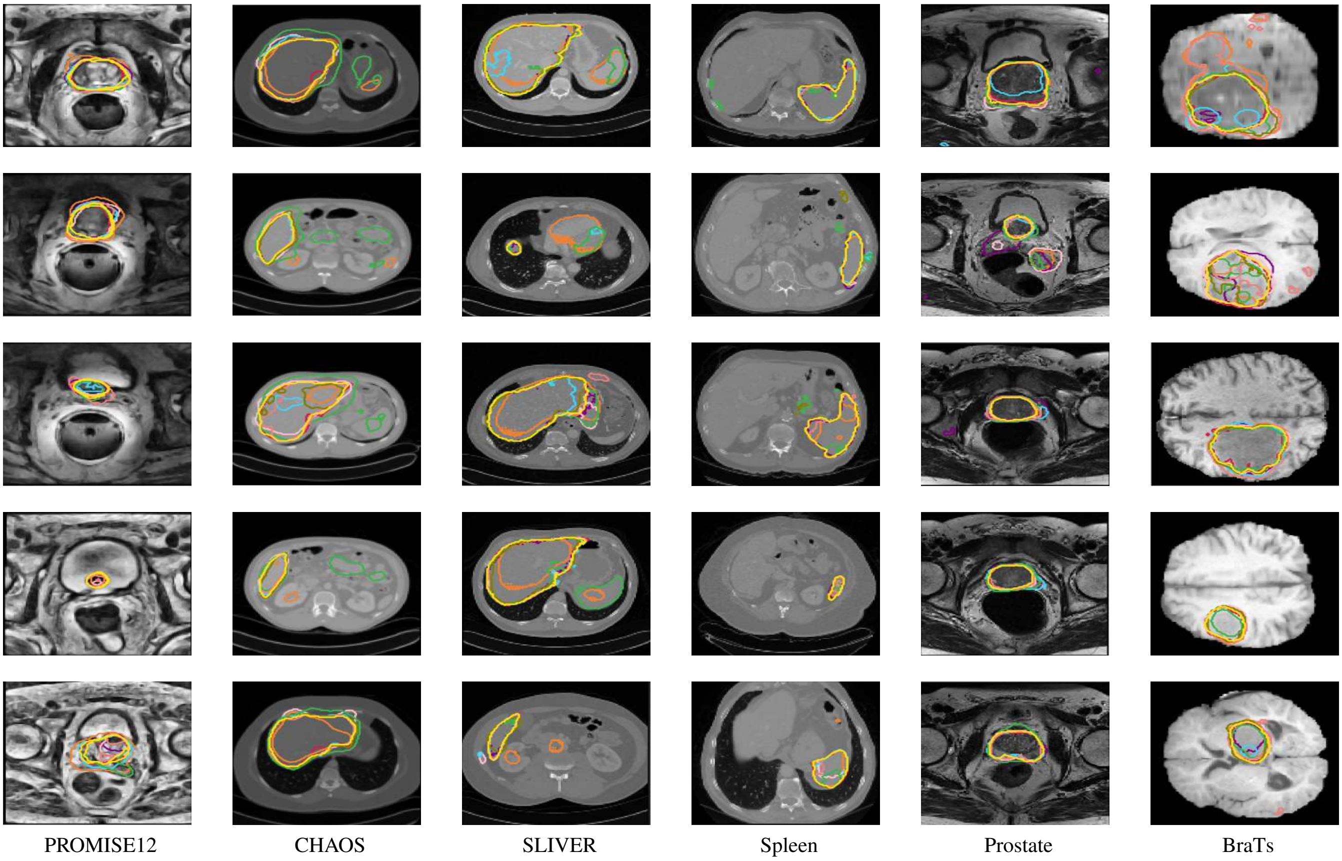


FIGURE 4. Five sample segmented images from each dataset using various models. The red contour is the ground truth, orange is U–Net, olive is Non–Bypass Dense, purple is AdaRes U–Net, pink is EvoU–Net, cyan is Res U–Net, green is Dense U–Net, light coral is NAS U–Net, and yellow is proposed DenseRes model.

estingly, except for the segmentation of the Prostate dataset, our proposed DenseRes model achieved high accuracy for the 3D image segmentation. The reason is we developed networks for each of the datasets independently using an evolutionary technique. Therefore, we can see that even when we stack the image slices and evaluate the results in 3D, again the results are comparable and competitive with original 3D models for medical image segmentation, and even in most cases outperformed them. It needs to be outlined that the 2D networks are using a considerably smaller number of parameters compared to 3D networks and require less time for training.

Also, this section is provided with an example of the 3D segmented image in regards to each dataset as subjective evaluation (see Figure 5). As can be seen from Figure 5, the proposed model obtained high performance in terms of 3D image segmentation evaluation (the red volume is ground truth, and the cyan one is the obtained segmentation). The output segmentation shows the capability of a well-designed 2D network.

I. PROPOSED EVOLUTIONARY MODEL ANALYSIS

In this section, we analysed the proposed evolutionary DenseRes network evolution, structures and the training parameters of the obtained networks. All provided information in this section are based on the ten best networks of the best run for each dataset (collectively 60 networks).

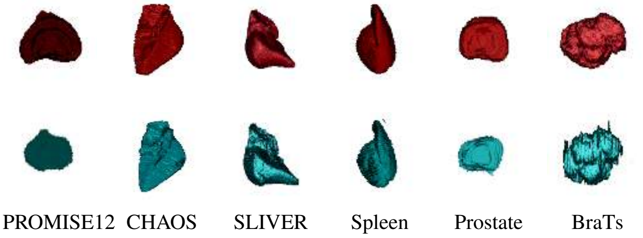


FIGURE 5. One sample 3D segmented image from each dataset. The red contour is the ground truth, cyan is output segmentation.

1) Network's training parameters

In the proposed DenseRes evolutionary technique, the type of the optimiser and its corresponding learning rate was selected automatically during the evolution. Based on the obtained results, for the six datasets, Figure 6 shows that among four various optimisers, the best networks mainly utilised rmsprop [80] and adagrad [81]. Also, Figure 6 shows the distribution of using learning rate. Most of the best networks applied the smallest learning rate (0.001) for training.

The next training parameters that were found using the

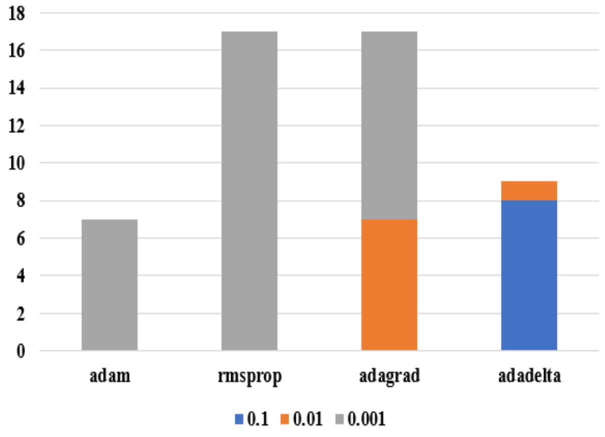


FIGURE 6. The type and number of optimisers and the learning rates that the best networks were used.

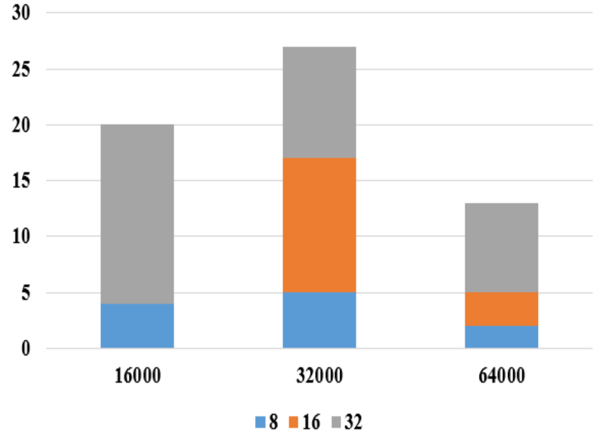


FIGURE 7. The number of augmentations and batch size of the best networks.

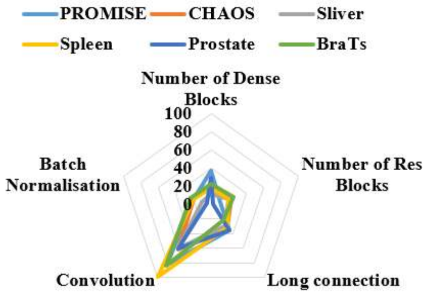


FIGURE 8. Contribution of the various parameters to create best networks.

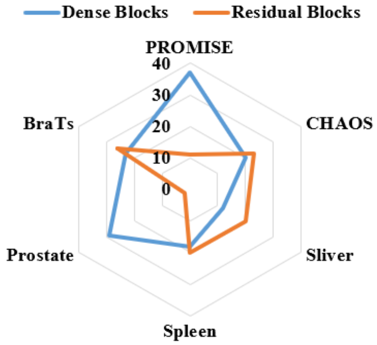


FIGURE 9. Contribution of the Dense and Residual blocks to create best networks.

evolutionary model are the number of augmentations and the batch size. The number of training data depends on the size of the network and its parameters. Using an inappropriate number of training images can be the cause of over or underfitting. Notably, in our proposed model that is constructed using Dense and Residual blocks, the size of the constructed networks can be variable, therefore finding the number of training images automatically can help to increase the accuracy of the proposed model. As shown in Figure 7, most of the best networks were trained using 32000 and then 16000 images. Also, Figure 7, shows the distribution of batch size.

2) Network's structure parameters

The components that were utilised to create the best network's structures are analysed in this section. All numbers in this section are based on the down–sampling and bridging sections of the found networks. First, the contribution of the Dense blocks, Res Blocks, convolution layers inside of the blocks, batch normalisation, and long connections to create networks are shown in Figure 8. As can be seen, convolution has the most prominent role in building networks in all six datasets. Moreover, 108 residual blocks and 136 dense blocks were used to create the best networks (see Figure 9). As shown, the ten best networks that were created using the PROMISE dataset mostly utilised Dense blocks, however, the ten best networks created using the prostate dataset used the minimum number of Residual blocks. This diversity shows the complexity of finding the best combination of blocks to create a network.

Besides the number of convolution layers, the filter size also plays a significant role in feature extraction. In the proposed model, the filter size was found for each convolution layer independently. Based on the best–obtained networks, 190 convolution layers applied 3×3, 164 convolution layers used 7×7, and just 91 convolution layers utilised 5×5 filters—the distribution of the mentioned parameters to create networks for each dataset are provided in Figure 10.

Also, pooling layers have an essential role in excluding unimportant features from extracted feature maps. In U–Net–based networks, after each block there is a pooling layer in the down–sampling section to reduce the size of the feature maps and consequently to remove less informative features. Two popular pooling layers are Maxpooling and Averagepooling layers were chosen by the proposed evolutionary model. As shown in Figure 11, to create the ten best networks for six datasets, 167 average and 78 max–pooling layers were utilised. As can be seen from Figure 11, for the Prostate and CHAOS datasets, the best networks used an almost equal number of Max and Average pooling, however, to create a network for spleen dataset, just two Max–poolings were used. Again the obtained results show the variety of

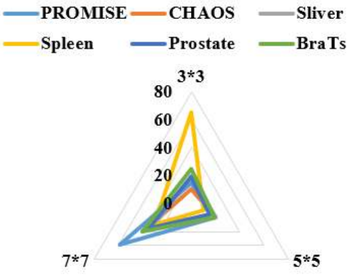


FIGURE 10. Contribution of various filter's sizes for feature extraction in the best networks.

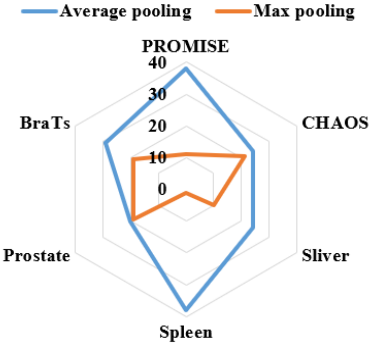


FIGURE 11. Contribution of pooling layers to create the best networks.

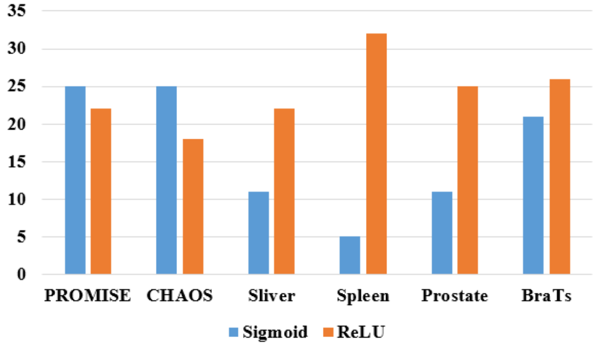


FIGURE 12. Contribution of activation functions to create the best networks.

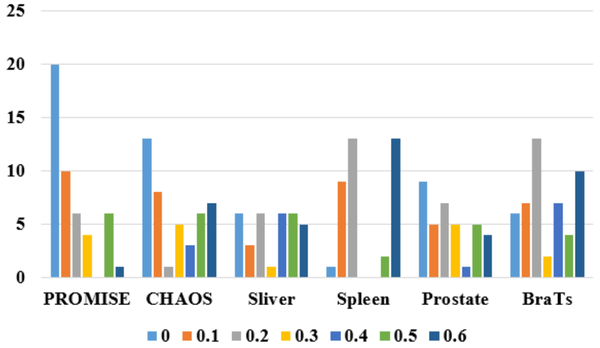


FIGURE 13. Contribution of dropout operation to create the best networks.

the network structures and the importance of the automatic creation of the networks.

Activation functions need to be used to specify the output of each node in regards to its input, which has an essential role in the neural network convergence and convergence speed. However, in some cases, it can prevent the network from converging. Therefore, it is necessary to specify an appropriate activation function for each convolution layer, which is specified for each block in the proposed model. As shown in Figure 12, networks corresponding to each dataset, applied various numbers and types of activation function. Collectively, 145 ReLu and 99 Sigmoid activation functions were assigned to the blocks.

Finally, in Figure 13, the distribution of using dropout with various probability is shown. In the proposed model, it is possible to have a dropout at the end of each block to control overfitting. When the probability of dropout is zero, it means there is no dropout, which happens several times in PROMISE's best networks. However, Figure 13 shows the patterns of using dropout in different datasets.

3) Time related parameters

One of the critical parameters to build a network is the number of trainable parameters that play significant roles in the training of networks and their accuracy. Moreover, the training time of the network is related to its size (also the number of training data). Figure 14, shows the size of the ten best networks in regards to each dataset. Notably, networks with significantly different sizes obtained usually high accuracy in most cases. It shows that increasing the depth and number of parameters of a network is not always a solution to create a more precise network.

Finally, in this section, the required wall–time to find a set of networks corresponding to each dataset, using one Nvidia GPU is provided in Figure 15. Because the size of the networks and the number of augmentations are variable, the training time is different. For example, training of the proposed model for the CHAOS dataset takes less than 60 hours; however, the case of the Spleen dataset takes about 100 hours. Overall, Figure 15 shows that the proposed model is not computationally expensive.

4) Network's evolution

As mentioned above, the proposed evolutionary model was repeated ten times for each dataset. In Table 16, the evolutionary process is shown during the nine generations for each dataset. The orange line is training, blue is the test, and green shows the validation accuracy during the evolution. The solid line shows the accuracy of the best run during the evolution, and the dashed line shows the average accuracy of the ten runs. It needs to be noted that the way the test, validation and the training sets have been selected randomly can affect the evolution process. For example, if randomly, unique and challenging samples are located in the test set while the easy samples are in the validation set, training will cause high accuracy in the validation set and low accuracy in the test set. This problem can be addressed to some extent by using an appropriate augmentation technique.

Before analysing each model separately, it needs to be out-

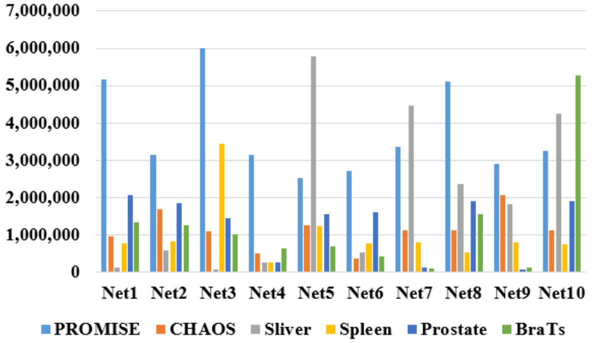


FIGURE 14. The number of trainable parameters of the best networks for each dataset.

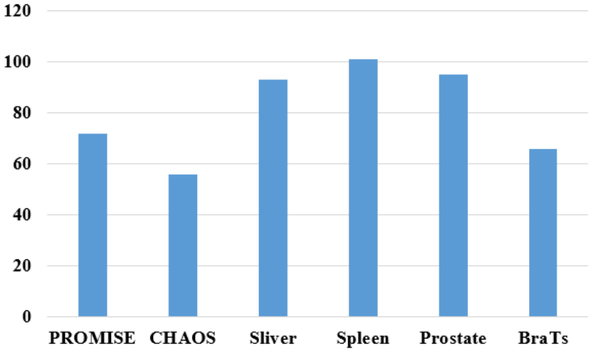


FIGURE 15. The training time of the proposed model for each dataset.

lined that the evolutionary process starts with 60 population and then training is continued with 30 networks, which is a large number of population size compared to previous evolutionary models [30]. As shown in Table 16, in the PROMISE dataset, the training accuracy starts at about 0.65 and then increases to 0.73, and then accuracy decreased. Therefore the best results were obtained in the seventh generation. Also, in the CHAOS model, there is an inconsistency during the training as training accuracy increased and decreased during the training. In the case of the Sliver dataset, training starts with a set of networks with high accuracy, in average the best networks obtained 0.94 accuracy, which increased about one per cent during the training. Besides, evolution in the Spleen dataset started with 0.77 accuracy in the best run and ended with ten per cent improvement in segmentation accuracy in the training set at the end of the training. Also, for the Prostate dataset, the training accuracy in the best run starts with 0.77, while increased to 0.90 in the ninth generation. Lastly, for the brain cancer segmentation, the training accuracy in the best run starts at 0.47 and then increased to 0.56 in the last generation.

Overall, despite the limited number of generation, we can see improvement in segmentation accuracy during evolution.

V. DISCUSSION AND CONCLUSION

In this paper, we proposed the first evolutionary DenseRes model for medical image segmentation, leveraging from U–Net [32], Res U–Net [35], Dense U–Net [36], and GA [11], such that our proposed automatic model can uncover an accurate network structure along with its training parameters for each dataset precisely and quickly. Experimental results utilising six different datasets show that the proposed DenseRes evolutionary technique can be used as a general model for developing a network automatically for different segmentation purposes. That is one of the most significant achievements of the proposed model. Because developing a precise network structure along with its parameters is sometimes completely different from a dataset or application to another one and there is no general rule to address this. However, using a general technique like our proposed model can address this problem, especially as the obtained networks are relatively small and do not need extensive computation.

The proposed model is not computationally expensive, such that to find the best network using one GPU, the maximum training time was about 100 hours. That is again reachable and comparable to other automatic techniques [21], [23], which used hundreds of GPUs in some cases, thus it is effective and computationally reasonable and cheap. Another essential aspect of our proposed model is that, unlike most of the previous automatic techniques that included only a part of their parameters in their search space, we include all necessary parameters in the search space and by utilising the relatively large population size, our proposed model found good parameters. Such that the end–user even does not need to set the batch size and augmentation numbers manually, which have a strong effective on whether a network is over or under fitted.

We evaluated the proposed model for medical image segmentation; and the proposed model is flexible enough to be applied to other medical segmentation datasets, in addition to those we used. Besides, with modifying the proposed encoding model and changing the number of channels from one to three, it is possible to develop networks for natural image segmentation. Also, depending on the size of the input image; the number of blocks can be increased to find deeper networks. In addition, based on our proposed encoding model, other evolutionary techniques can be applied in this application instead of GA. Lastly, other block's structures (like attention models) can be included in the search space to find more complicated networks' structure. Finally, in this paper, we show that using the combination of the existing network structures and an evolutionary technique, is an effective technique to find a 2D network automatically. However, the results show that even our proposed model works well for 3D segmentation.

ACKNOWLEDGMENT

This research was undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

REFERENCES

[1] T. Back, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press, 1996.

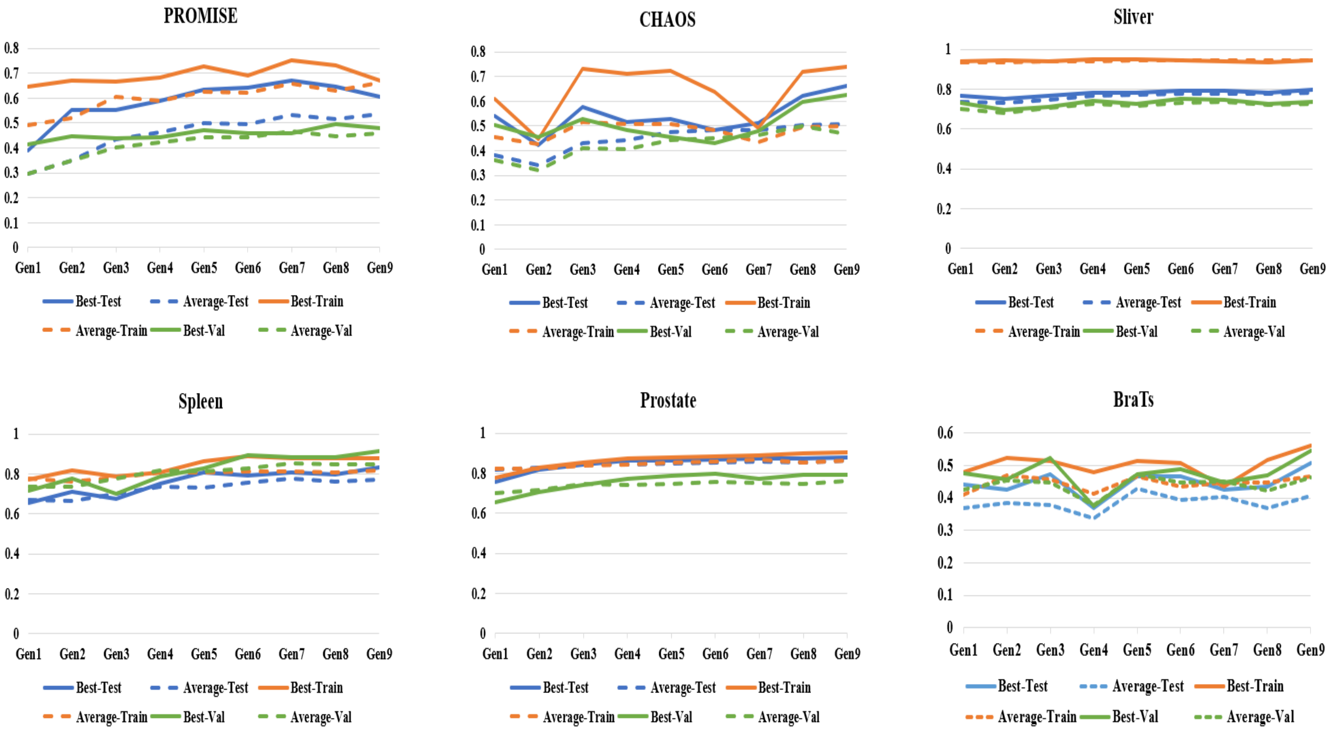


FIGURE 16. The evolution of the ten best networks for each dataset.

[2] R. S. Sutton, A. G. Barto et al., Introduction to reinforcement learning. MIT press Cambridge, 1998, vol. 2, no. 4.

[3] K. O. Stanley, "Neuroevolution: A different kind of deep learning," Obtenido de https://www. oreilly. com/ideas/neuroevolution-a-different-kindof-deep-learning el, vol. 27, no. 04, p. 2019, 2017.

[4] J. D. Schaffer, D. Whitley, and L. J. Eshelman, "Combinations of genetic algorithms and neural networks: A survey of the state of the art," in [Proceedings] COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks. IEEE, 1992, pp. 1–37.

[5] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," Nature Machine Intelligence, vol. 1, no. 1, pp. 24–35, 2019.

[6] D. Floreano, P. Dürr, and C. Mattiussi, "Neuroevolution: from architectures to learning," Evolutionary intelligence, vol. 1, no. 1, pp. 47–62, 2008.

[7] A. Baldominos, Y. Saez, and P. Isasi, "On the automated, evolutionary design of neural networks: past, present, and future," Neural Computing and Applications, pp. 1–27, 2019.

[8] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," Engineering Applications of Artificial Intelligence, vol. 60, pp. 97–116, 2017.

[9] X. Yao and M. M. Islam, "Evolving artificial neural network ensembles," IEEE Computational Intelligence Magazine, vol. 3, no. 1, pp. 31–42, 2008.

[10] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms." in IJCAI, vol. 89, 1989, pp. 762–767.

[11] J. H. Holland et al., Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press, 1992.

[12] S. Harp, "Towards the genetic synthesis of neural networks," Proc. of ICGA-89, pp. 360–369, 1989.

[13] Q.-H. Ling, Y.-Q. Song, F. Han, and H. Lu, "An improved evolutionary random neural networks based on particle swarm optimization and input-to-output sensitivity," in International Conference on Intelligent Computing. Springer, 2017, pp. 121–127.

[14] T. Hassanzadeh, K. Faez, and G. Seyfi, "A speech recognition system based on structure equivalent fuzzy neural network trained by firefly algorithm," in 2012 International Conference on Biomedical Engineering (ICoBE). IEEE, 2012, pp. 63–67.

[15] J. Koutník, J. Schmidhuber, and F. Gomez, "Evolving deep unsupervised convolutional networks for vision-based reinforcement learning," in Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM, 2014, pp. 541–548.

[16] A. Martín, R. Lara-Cabrera, F. Fuentes-Hurtado, V. Naranjo, and D. Camacho, "Evodeep: a new evolutionary approach for automatic deep neural networks parametrisation," Journal of Parallel and Distributed Computing, vol. 117, pp. 180–191, 2018.

[17] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," IEEE Transactions on Evolutionary Computation, vol. 24, no. 2, pp. 394–407, 2019.

[18] A. Baldominos, Y. Saez, and P. Isasi, "Evolutionary convolutional neural networks: An application to handwriting recognition," Neurocomputing, vol. 283, pp. 38–52, 2018.

[19] B. Wang, Y. Sun, B. Xue, and M. Zhang, "Evolving deep convolutional neural networks by variable-length particle swarm optimization for image classification," in 2018 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2018, pp. 1–8.

[20] B. Wang, Y. Sun, B. Xue, and M. Zhang, "A hybrid differential evolution approach to designing deep convolutional neural networks for image classification," in Australasian Joint Conference on Artificial Intelligence. Springer, 2018, pp. 237–250.

[21] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," arXiv preprint arXiv:1611.01578, 2016.

[22] D. E. Rumelhart, G. E. Hinton, R. J. Williams et al., "Learning representations by back-propagating errors," Cognitive modeling, vol. 5, no. 3, p. 1, 1988.

[23] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 4780–4789.

[24] B. Fielding and L. Zhang, "Evolving image classification architectures with enhanced particle swarm optimisation," IEEE Access, vol. 6, pp. 68 560–68 575, 2018.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[26] J. Kennedy, "Particle swarm optimization," Encyclopedia of machine learning, pp. 760–766, 2010.

[27] Y. Sun, B. Xue, and M. Zhang, "Automatically evolving cnn architectures based on blocks," arXiv preprint arXiv:1810.11875, 2018.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely

connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[30] M. Baldeon-Calisto and S. K. Lai-Yuen, "Adaresu-net: Multiobjective adaptive convolutional neural network for medical image segmentation," Neurocomputing, vol. 392, pp. 325–340, 2020.

[31] T. Hassanzadeh, D. Essam, and R. Sarker, "Evou-net: an evolutionary deep fully convolutional neural network for medical image segmentation," in Proceedings of the 35th Annual ACM Symposium on Applied Computing, 2020, pp. 181–189.

[32] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.

[33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.

[34] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," arXiv preprint arXiv:1502.03167, 2015.

[35] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual u-net," IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 5, pp. 749–753, 2018.

[36] M. Kolařík, R. Burget, V. Uher, K. Říha, and M. K. Dutta, "Optimized high resolution 3d dense-u-net network for brain and spine segmentation," Applied Sciences, vol. 9, no. 3, p. 404, 2019.

[37] T. Hassanzadeh, L. G. Hamey, and K. Ho-Shon, "Convolutional neural networks for prostate magnetic resonance image segmentation," IEEE Access, vol. 7, pp. 36 748–36 760, 2019.

[38] Y. Weng, T. Zhou, Y. Li, and X. Qiu, "Nas-unet: Neural architecture search for medical image segmentation," IEEE Access, vol. 7, pp. 44 247–44 257, 2019.

[39] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[40] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," arXiv preprint arXiv:1202.2745, 2012.

[41] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in 2010 IEEE Computer Society Conference on computer vision and pattern recognition. IEEE, 2010, pp. 2528–2535.

[42] Y. Bengio, P. Simard, P. Frasconi et al., "Learning long-term dependencies with gradient descent is difficult," IEEE transactions on neural networks, vol. 5, no. 2, pp. 157–166, 1994.

[43] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on. IEEE, 2017, pp. 1175–1183.

[44] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," arXiv preprint arXiv:1505.00387, 2015.

[45] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 315–323.

[46] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, p. 436, 2015.

[47] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang et al., "Evaluation of prostate segmentation algorithms for MRI: the PROMISE12 challenge," Medical Image Analysis, vol. 18, no. 2, pp. 359–373, 2014.

[48] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. van der Laak, B. van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60–88, 2017.

[49] D. Shen, G. Wu, and H. I. Suk, "Deep learning in medical image analysis," Annual Review of Biomedical Engineering, vol. 19, pp. 221–248, 2017.

[50] H. C. Shin, K. Roberts, L. Lu, D. Demner Fushman, J. Yao, and R. M. Summers, "Learning to read chest X-rays: Recurrent neural cascade model for automated image annotation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2497–2506.

[51] H. I. Suk, S. W. Lee, D. Shen, A. D. N. Initiative et al., "Latent feature representation with stacked auto-encoder for AD/MCI diagnosis," Brain Structure and Function, vol. 220, no. 2, pp. 841–859, 2015.

[52] R. Cheng, H. R. Roth, L. Lu, S. Wang, B. Turkbey, W. Gandler, E. S. McCreedy, H. K. Agarwal, P. Choyke, R. M. Summers et al., "Active appearance model and deep learning for more accurate prostate segmentation on MRI," in Medical Imaging 2016: Image Processing, vol. 9784. International Society for Optics and Photonics, 2016, p. 97842I.

[53] R. Cheng, H. R. Roth, N. S. Lay, L. Lu, B. Turkbey, W. Gandler, E. S. McCreedy, T. J. Pohida, P. A. Pinto, P. L. Choyke et al., "Automatic magnetic resonance prostate segmentation by deep learning with holistically nested networks," Journal of Medical Imaging, vol. 4, no. 4, p. 041302, 2017.

[54] H. Jia, Y. Xia, Y. Song, W. Cai, M. Fulham, and D. D. Feng, "Atlas registration and ensemble deep convolutional neural network-based prostate segmentation using magnetic resonance imaging," Neurocomputing, vol. 275, pp. 1358–1369, 2018.

[55] K. Yan, C. Li, X. Wang, A. Li, Y. Yuan, D. Feng, M. Khadra, and J. Kim, "Automatic prostate segmentation on MR images with deep network and graph model," in Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the. IEEE, 2016, pp. 635–638.

[56] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," arXiv preprint arXiv:1901.06032, 2019.

[57] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," arXiv preprint arXiv:2001.05566, 2020.

[58] M. Sornam, K. Muthusubash, and V. Vanitha, "A survey on image classification and activity recognition using deep convolutional neural network architecture," in 2017 Ninth International Conference on Advanced Computing (ICoAC). IEEE, 2017, pp. 121–126.

[59] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.

[60] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[61] O. Delalleau and Y. Bengio, "Shallow vs. deep sum-product networks," in Advances in Neural Information Processing Systems, 2011, pp. 666–674.

[62] T. Blickle and L. Thiele, "A comparison of selection schemes used in genetic algorithms. tik-report 11, tik institut fur technische informatik und kommunikationsnetze," Computer Engineering and Networks Laboratory, ETH, Swiss Federal Institute of Technology, Gloriastrasse, vol. 35, p. 8092, 1995.

[63] A. E. Kavur, M. A. Selver, O. Dicle, M. Baris, and N. S. Gezer. CHAOS - combined (CT-MR) healthy abdominal organ segmentation challenge data. https://doi.org/10.5281/zenodo.3431873.

[64] T. Heimann, B. Van Ginneken, M. A. Styner, Y. Arzhaeva, V. Aurich, C. Bauer, A. Beck, C. Becker, R. Beichel, G. Bekes et al., "Comparison and evaluation of methods for liver segmentation from ct datasets," IEEE transactions on medical imaging, vol. 28, no. 8, pp. 1251–1265, 2009.

[65] Medical segmentation decathlon. https://decathlon-10.grand-challenge.org/.

[66] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest et al., "The multimodal brain tumor image segmentation benchmark (brats)," IEEE transactions on medical imaging, vol. 34, no. 10, pp. 1993–2024, 2014.

[67] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. S. Kirby, J. B. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," Scientific data, vol. 4, p. 170117, 2017.

[68] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," arXiv preprint arXiv:1811.02629, 2018.

[69] D. Zill, W. S. Wright, and M. R. Cullen, Advanced engineering mathematics. Jones & Bartlett Learning, 2011.

[70] P. Y. Simard, D. Steinkraus, J. C. Platt et al., "Best practices for convolutional neural networks applied to visual document analysis." in Icdar, vol. 3, no. 2003, 2003.

[71] F. Chollet et al., "Keras," 2015.

[72] L. R. Dice, "Measures of the amount of ecologic association between species," Ecology, vol. 26, no. 3, pp. 297–302, 1945.

[73] S. S. Chandra, J. A. Dowling, K.-K. Shen, P. Raniga, J. P. Pluim, P. B. Greer, O. Salvado, and J. Fripp, "Patient specific prostate segmentation in

3-d magnetic resonance images," IEEE transactions on medical imaging, vol. 31, no. 10, pp. 1955–1964, 2012.

[74] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[75] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8697–8710.

[76] A. Mortazi and U. Bagci, "Automatically designing cnn architectures for medical image segmentation," in International Workshop on Machine Learning in Medical Imaging. Springer, 2018, pp. 98–106.

[77] N. Dong, M. Xu, X. Liang, Y. Jiang, W. Dai, and E. Xing, "Neural architecture search for adversarial medical image segmentation," in International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer, 2019, pp. 828–836.

[78] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, "3d u-net: learning dense volumetric segmentation from sparse annotation," in International conference on medical image computing and computer-assisted intervention. Springer, 2016, pp. 424–432.

[79] L. Yu, X. Yang, H. Chen, J. Qin, P.-A. Heng et al., "Volumetric convnets with mixed residual connections for automated prostate segmentation from 3d mr images." in AAAI, vol. 17, 2017, pp. 36–72.

[80] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural networks for machine learning, vol. 4, no. 2, pp. 26–31, 2012.

[81] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," Journal of Machine Learning Research, vol. 12, no. Jul, pp. 2121–2159, 2011.

TAHEREH HASSANZADEH Obtained a Master degree in Artificial Intelligence (AI) from Azad University–Qazvin branch, Iran and a Master by Research degree in Computing from Macquarie University, Australia. Currently, she is a Researcher at the University of New South Wales (UNSW), Australia. Tahereh's research interests are AI, medical image processing, deep learning, and optimisation.

DARYL ESSAM Received his B.Sc. degree in computer science from University of New England, Australia in 1990 and his Ph.D. degree from University of New South Wales, Australia, in 2000. Since 1994, he has been with the Canberra campus, UNSW, where he is currently a Senior Lecturer. His research interests include genetic algorithms, with a focus on both evolutionary optimisation and large scale problems.

RUHUL SARKER Obtained his Ph.D. from Dalhousie University (former TUNS), Canada. He is a Professor in the School of Engineering and IT (SEIT), and the Director of Faculty PG Research at UNSW Canberra (located at ADFA) since May 2015. He also served as Deputy Head of School (Research) of SEIT (2011-2014). Prof. Sarker's broad research interests are decision analytics, CI / evolutionary computation, operations research, and applied optimisation.