Evo-GUNet3++: Using evolutionary algorithms to train UNet-based architectures for efficient 3D lung cancer detection

Pasquale Ardimento, Lerina Aversano, Mario Luca Bernardi, Marta Cimitile, Martina Iammarino, Chiara Verdone

Abstract

The early detection of malignant lung nodules can strongly increase the chances of life in lung cancer patients. A computer tomography scan represents an effective way to identify and locate malignant nodules in the body and monitor their growth. However, the reading and interpretation of tomography scans are subject to errors that can be reduced with a second reader. The adoption of image processing systems can reduce the possibility of errors and can support radiologists in ensuring multiple readings of tomography scans. This study proposes a new approach for accurate 3D lung nodule detection starting from computer tomography scans. This work exploits an evolutionary algorithm to build variants of a UNet-based architecture, called GUNet3++, to detect patients affected by lung cancer, from the analysis of CT-scan images of lungs. The approach is validated on the LIDC-IDRI real dataset and results show that it improves segmentation quality metrics (IoU and Dice) over baselines, leading to better 3D models reconstruction of lesions.

Keywords: U-Net, CT scan Images, Lung Cancer, Segmentation, Genetic Algorithms

1. Introduction

Lung cancer is the leading cause of cancer mortality in the last years [1] and according to the estimations [2] it will increase its deadliness in the next years. However, the level of mortality can be significantly reduced by performing a regular screening of high-risk individuals [3, 4] with computed tomography scans (CTs). Indeed, CTs represents an effective way to

identify and locate the malignant nodules in the body and monitor their growth. However, the reading and interpretation of tomography scans are highly error-prone and require that radiologists read several scans to detect possible nodules. This can cause incorrect diagnoses with serious risks for the patient life. The adoption of computer-aided detection (CAD) systems (i.e., image processing systems) can significantly reduce the possibility of errors, can detect almost half of the lesions missed by humans [5] and can support radiologists by reducing the reading times or acting as a second reader [6] of the CTs [7]. For this reason, several CAD systems for the detection of lung nodules in CT images systems have been developed in the last years [8, 9]. Some machine learning approaches [10] are based on the analysis of common features (i.e., shape, volume, solidity) to discriminate malignant or benign nodules. In the last years, Deep Learning (DL) approaches also show good performances to detect and classify lung nodules [11]. These approaches show higher performance with respect to the traditional lung nodule detection systems. An example of a DL-based approach for lung nodule detection using CT scan images is proposed in [12]. Here a lobe-driven CT image clustering [13] is used to improve the detection performance compared to existing approaches.
In this study, a novel DL-based evolutionary approach for automatic and accurate 3D lung cancer model detection using CT scan images is proposed. The cancer lesions detector, performing semantic segmentation, is obtained by searching an extended version of a generalized UNet-based architecture, called GUNet3++, evolved with a direct coding scheme based on genetic programming, and fine-tuned using hyperparameter optimization.

Concerning existing approaches, the one proposed collects all the information related to a complete CT scan session to obtain a 3D model of the detected lung nodules. The main assumption at the base of this approach is that each CT scan produces a large number of images that represent different sections of the nodule. A more complete model of the nodule can be obtained by collecting all these images in a single 3D model of the nodule. This, for example, improves the capability to detect small nodules that are overlooked since they require the analysis of a large number of images [9].
Section 2 presents and discusses the most relevant related work, highlighting differences and common aspects. In Section 3, some basic notions on

the adopted pre-trained neural networks are provided. The proposed approach is described in Section 4, whereas the experiment description is explained in Section 5. A discussion of the experiment results is reported in Section 6. Finally, Section 7 highlights some threats to the validity of the described experiments, while Section 8 discusses some final remarks and future research directions.

2. Related Work

Lung nodule detection aims at identifying the features and the locations of different nodules. This is still a complex task because the sizes and features of the nodules vary. For example, juxta-pleural and juxta-vascular nodules, are hard to differentiate from the pleura and vessels. Recently, to learn discriminative features automatically, several researchers have applied DL methods to detect lung nodules. The obtained results are encouraging [14]. DL shows, infact, improved detection capability with respect to the traditional lung cancer detection systems [15]. An example of DL application for the early-stage diagnosis is described in [16]. Authors use a DL technique to perform nodules detection and classification and they reduce the possible mistakes and the number of false positives by using additional information (i.e., clinical factors). A good accuracy (85.4% and 90.1% at 1 and 4 false positives per scan in the best case) in lung cancer CT detection is obtained in [17] where multi-view convolutional networks (2-D ConvNets consisting of 3 consecutive convolutional layers and max-pooling layer) are used. The detection of nodule from the CT scans is also performed in [18] using the so-called DeepLung approach. It consists of a first step for detecting candidate nodules from the CT images and a following step for the extraction of the deep features and the detection of the nodules. The first step is based on a 3D Faster R-CNN while the following step is performed using a dual-path network (DPN). [19] proposed a 3D Deep Convolutional Neural Network where, to leverage 3-dimensional information from Computed Tomography (CT) scans, they applied median intensity projection and multi-Region Proposal Network (mRPN) for automatic selection of potential region of-interests. They obtained a 97.4% sensitivity value for 2.1 false positives per scan. [12] propose a DL-based ensemble approach based on an image clustering step that allows to specialize each classifier to a specific lobe zone. This approach ensures an increased detection accuracy since each classifier is specialized to capture the

patterns of a specific lung lobe. The detection is performed using different pre-trained convolutional neural networks (the best classifier for each case is selected) obtaining an accuracy of 0.96 in single image classification and 0.94 in the patient classification.

More recently, [20] propose the UNEt TRansformers (UNETR) network, based on a transformer as the encoder with the aim of learning sequence representations of the input volume to capture multi-scale information. The UNETR network is still based on the "U-shaped" network design for the encoder and decoder. However, the transformer encoder is directly connected to a decoder via skip connections at various resolutions to generate the output providing the semantic segmentation.

Another approach, based on AutoML, is described by [21]. The authors propose a new automated machine learning algorithm (called T-AutoML) which performs a jointly search of the best neural architecture along with the best combination of hyperparameters and data augmentation strategies simultaneously. The proposed method utilizes a transformer model to adapt to the specific properties of search space embeddings, improving the search performances.

Differently from the above techniques, our proposal exploits evolutionary algorithms to derive a U-Net based network [22] which is better suited to the specific segmentation task for which the search is performed, exhibiting final better performance. This is confirmed by several studies that mainly used U-Net and U-Net based approaches to perform image segmentation in the medical domain with good results [23, 24, 25]. However, the variants of the original U-Net come with a limitation. The diversity of features is lost due to the fixed receptive field of the convolution kernel. The same scale feature maps extracted from the convolution kernel with different receptive fields are semantically different. As a result, the performance of the network may vary with the size of the receptive field, and the performance depends on the size of the receptive field in the convolution kernel.

This study proposes, with respect to the discussed related work, a new architecture called GUNet3++ characterized by multi-scale skip-connections allowing the network to learn from contextual information across different regions of growing sizes at multiple scales. Moreover, the proposed approach exploits a genetic algorithm to drive an architectural search process in the frame of the GUNet3++ structure to select the best architecture variant for the given segmentation task.

3. Background

Image segmentation is a typical computer vision activity where each pixel of the image is labeled according to what's being shown. Different from other approaches, semantic segmentation does not distinguish among images of the same class but only cares about the category of each pixel (i.e., all the pixels of the same nodule are grouped in the same class). This makes semantic segmentation very useful in the medical domain where it supports radiologists in their diagnostic tasks. In the following the major variants of UNet networks, that have inspired the architecture used as building block of the proposed approach (called GUNET3++), are briefly described.

3.1. UNET

UNet is a fully convolutional network [26] developed for ensuring high-quality and quick segmentation of medical imagines and is able to work with small training datasets thanks to heavy data augmentation. It has also been successfully used in 3D imaging segmentation [27]. The U-net network has 23 convolutional layers and is structured into two main sections. The first one is the contracting path that exploits a classical CNN architecture. Each block of this contracting path is made up of a sequence of two 3×3 convolutions followed by an activation function unit (i.e., a rectified linear unit — ReLU) and by a max-pooling layer (with a double stride for downsampling — the feature channel are doubled at each downsampling). The interesting part is the second one, named the expansive path, where each stage performs up-sampling on the feature maps using up-convolutions. The network is characterized by a high number of feature channels in the upsampling data allowing the propagation of context information to the layers with higher resolution. For this reason, the expansion of this architecture can be more or less symmetric since the network tends to use only valid nodes (the layers are not fully connected).

The result is an overall network with a shape similar to a big U. Through the network, layers propagate contextual information, allowing to segment objects in a region using the context data arising from a larger overlapping one. The feature maps from the corresponding layer in the contracting path are cropped and concatenated to form the upsampled feature maps. This crop step is necessary to discard pixel features at the edges that have the lower amount of contextual information. Finally, in the last layer,

a 1x1 convolution relates every single component of the feature vector to the desired number of classes.

The energy function for the network can be written as:

$$E(x) = \sum_{x \in \Omega} w(x) \cdot Log(S_{\ell(x)}(x)) \tag{1}$$

where $w(x)$ defined as $w : \Omega \longrightarrow \mathbb{R}$ is a weights map to give at each pixel $x \in \Omega \subset \mathbb{Z}^2$ a lower or higher importance during training, $\ell : \Omega \longrightarrow \{1, \ldots, K\}$ is the true label of each pixel, and $S_k(x)$ is the pixel-wise SoftMax formalized as:

$$S_k(x) = \exp\left(a_k(x)\right) / \sum_{h=1}^{K} \exp\left(a_h(x)\right) \tag{2}$$

where $a_k(x)$ is the activation in feature channel k at pixel $x \in \Omega$, and K is the number of classes. Since $S_k(x) \approx 1$ for the k that has the maximum activation $a_k(x)$ and $S_k(x) \approx 0$ for all the other k, the cross entropy penalizes, at each position, the deviation of $S_{\ell(x)}(x)$ from one.

3.2. UNET+ and UNET++ variants

In [28] three UNet variants are proposed, namely UNet^e, UNet+ and UNet++. The first one is an ensemble architecture, called UNet^e, which combines U-Nets of varying depths into a single unified architecture. All inner U-Nets (partially) share the same encoders, but each one has its own decoder. As pointed out by the authors themselves, this network has some limitations: since the decoders are not connected the subsequent U-Nets do not provide supervision signals to the decoders of the previous U-Nets. Moreover, the skip connections used in the U-Net^e is too restrictive, combining the decoder feature maps only at the same-scale. From this ensemble version, the UNet+, depicted in Fig. 1-(a), is derived by modifying the original skip connections. In this variant, every two adjacent nodes are connected with a direct skip connection, allowing the deeper decoders to pass the supervision signals to the shallower decoders. For this reason, while UNet^e needs deep supervision in order to train the inner U-Nets, due to these direct skip connections, UNet+ can be trained in both normal and deep supervised fashion (i.e., the loss function shown in Fig. 1-(b) can be linked only to $X^{0,4}$ when performing normal training or to all $X^{0,j}$ nodes with $j \in \{0, \ldots, 4\}$ when training with deep supervision).

The final UNet++ architecture, depicted in Fig. 1-(b), is a further improvement over UNet+ for what concerns connectivity density. Specifically, UNet++ is constructed from U-Net+ by connecting the decoders, resulting in densely connected skip connections, enabling dense feature propagation along skip connections and thus more flexible feature fusion at the decoder nodes. Each node in a decoder is presented with not only the final aggregated feature maps but also with the intermediate aggregated feature maps and the original same-scale feature maps from the encoder. For this reason, each node in the UNet++ decoders combines multiscale features (horizontally, at same-scale) from its all preceding nodes(i.e., at the same resolution). Conversely, it integrates, vertically, multiscale features over different resolutions from the shallower previous node. The feature aggregation at multiple scales synthesizes the segmentation over layers more gently resulting in better accuracy and a more stable training process with improved convergence. This arrangement minimizes the loss of semantic information between the two UNet notable paths.

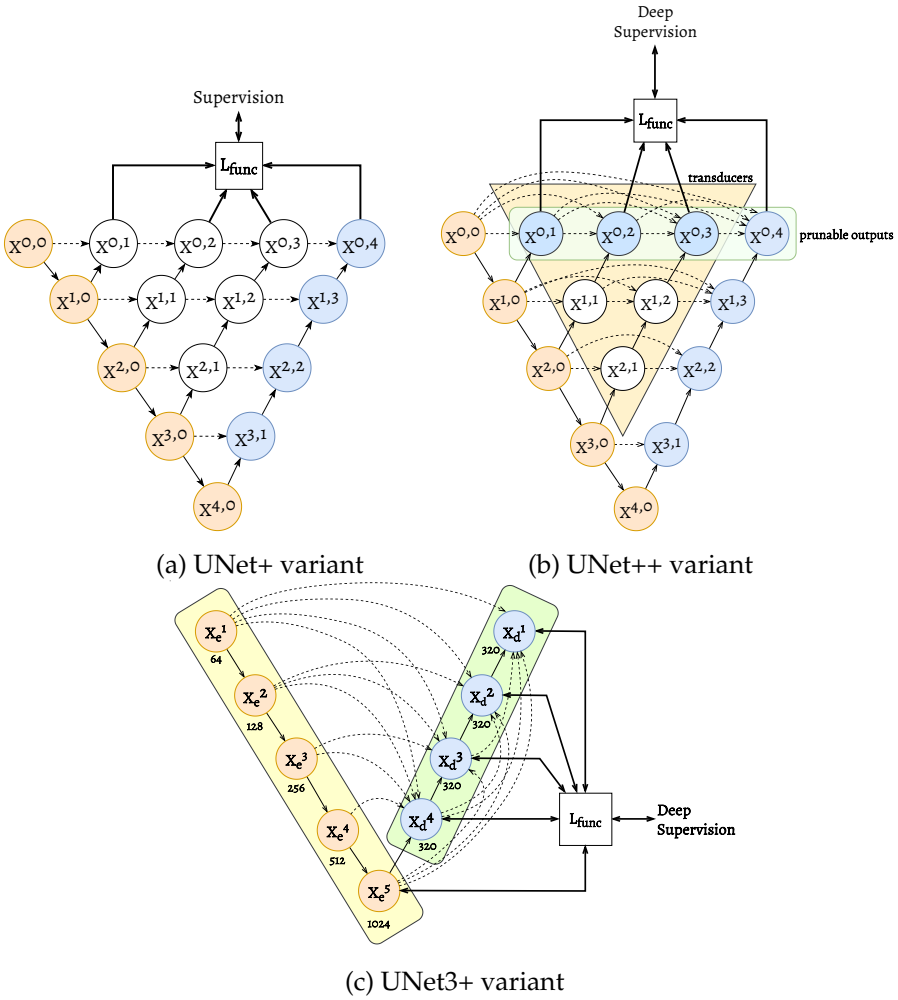
Let $x^{i,j}$ be the output of the node $X^{i,j}$ where i identifies the down-sampling layer on the encoders path and j identifies the convolution layer across the skip connections. The operation of the skip connection unit in which x is the feature map and (i, j) are the indexes down the contracting path and across the skip connections, can be defined as:

$$x^{i,j} = \begin{cases} \Theta_c(D_o(x^{i-1,j})) & j = 0 \\ \Theta_c(\left[[x^{i,k}]_{k=0}^{j-1}, U_p(x^{i+1,j-1})\right]) & j > 0 \end{cases}$$

where $\Theta_c(\cdot)$ is the convolution followed by the activation operation, $D_o(\cdot)$ is the down-sampling operation, $U_p(\cdot)$ represents the up-sampling operation, and $[\cdot]$ is the concatenation operator. The number of intermediary skip connection units depends on the layer number and decreases linearly when traversing the contracting path. Specifically, as shown in Fig. 1-(b), nodes at level $j > 1$ are feeded with $j + 1$ inputs, of which j inputs are the outputs of the previous j nodes at the same resolution and the $(j + 1)^{th}$ input is the output from the skip connection up-sampled from a lower resolution.

3.3. UNET3+ variant

Another notable variant derived from UNet is the UNet 3+ which merge the multi-scale features by modifying the skip connections and exploits



(a) UNet+ variant (b) UNet++ variant

(c) UNet3+ variant

Figure 1: Major variants of UNet architecture.

a deep supervision approach for training that spans over multiple scales. Specifically, both UNet and UNet++ do not use sufficient information from all available scales. For this reason in UNET3+, as shown in Figure 1-(c), each decoder layer acquires both smaller-or-same-scale feature maps from its corresponding encoder and larger-scale feature maps from the decoder. This results in capturing both coarse-grained contextual/semantic information and fine-grained details across all the available scales. To combine the shallow data with deep semantic data, a feature aggregation mechanism is applied (exploiting bilinear interpolation and a non-overlapping

max pooling operation) on the concatenated feature map from the available five scales and resulting, at decoding stages, in 320 filters of size 3×3, as reported in Fig. 1-(c). As suggested by the figure, the feature maps of X_d^i can be evaluated as:

$$X_d^i = \begin{cases} X_e^i, & i = N \\ \Theta_c([\mathcal{C}(D_o(X_e^k))_{k=1}^{j-1}, \mathcal{C}(X_e^i), \mathcal{C}(U_p(X_d^k))_{k=j+1}^{N}]) & i = 1, \ldots, N-1 \end{cases}$$

where $\mathcal{C}(\cdot)$ is the convolution operation, $\Theta_c(\cdot)$ is the feature aggregation operation followed by a convolution and by the activation function, $D_o(\cdot)$ is the down-sampling operation, $U_p(\cdot)$ represents the up-sampling operation, and $[\cdot]$ is the concatenation operator. The number of intermediary skip connection units depends on the layer number and decreases linearly when traversing the contracting path. Even if UNet3+ has fewer parameters than UNet++, on some tasks this arrangement yields a more accurate segmentation map improving both regions' positioning and boundaries identification.

4. The Evo-GUNet3++ approach

In this work, to find best suited UNet-based architectures for CT-scan semantic segmentation for cancer lesions detection, we propose to use a genetic algorithm to drive the architectural search process based on a more generalized UNet template network (called GUNet3++).

The overall process is highlighted in Figure 2: on the left side of the figure is depicted the overall structure of the genetic algorithm (GA) used for architectural search whereas on the right side the training process, using a partitioned dataset (training/validation), executes an hyper-parameters optimization step training each candidate network.

Specifically, as the figure shows, the algorithm executes an evolutionary process to discover the best architecture adaptation of the GUnet3++ model to perform segmentation of the CT scans provided as input. To this aim, it takes as input: (i) the set of predefined building blocks belonging to the GUnet3++ model, (ii) the population size (iii) the maximal generation number for the GA, and (iv) the image classification dataset.

The starting population is initialized using random choices with a predefined population size and exploiting an encoding strategy able to represent a set of possible desired adaptations of the original model along with their hyper-parameters. Then, during evolution, the fitness function of

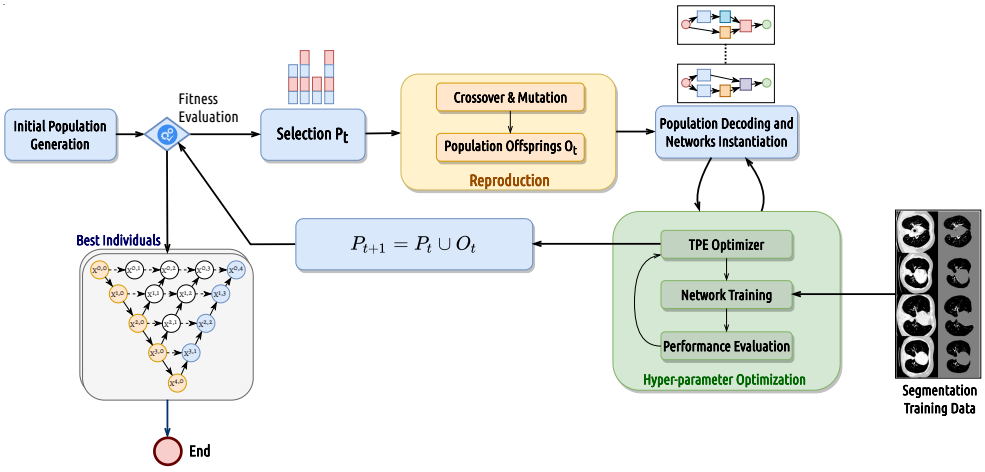


Figure 2: Overview of the proposed approach.

each individual, which encodes a particular architecture of the pre-trained model, is evaluated on the input datasets. At this point, the parent individuals are selected based on the fitness function, and then generate a new offspring by applying suitable genetic operators (e.g., mutation and crossover). Finally, the population of individuals that survives into the next generation is selected by applying environmental selection to the current population, composed of the generated offspring population and the parent one. The evolution cycle proceeds until the optimal performance is obtained or the maximum number of iterations is reached.

More specifically, the procedure of the used GA can be detailed as follows:

1. Instantiate the initial population of individuals P, each one generating a GUnet3++ model, and train the networks represented by P using the average dice as a fitness function. For each network, hyper-parameters optimization is performed by the Tree-Structured Parzen Estimator (TPE) algorithm as highlighted in figure;

2. Generate a set of λ offspring O, by applying the mutations to P. Mutation enforce the diversity of the population and avoid the search from ending in a local minimum. For this reason, the offspring is generated using crossover that has the probability M_p of for mutation, and each bit in the encoding has an independent probability

for flipping. The value is small enough (e.g., 0.035) to avoid excessive changes due to a single mutation operation.

3. Perform training on the λ modified pre-trained CNNs represented by offspring O, and assign the validation accuracy to the model as a fitness function;

4. Select elite individuals from the union of the sets of P and O, and then replace P with them;

5. Repeat from step 2 until the stopping criterion is satisfied.

The algorithm starts from individuals based on the considered pre-trained models, giving to each model equal chances to produce individuals that perform well on the specific classification task. However, if a pre-trained model is not suitable and produces individuals that are less performing, it is quickly discarded since it will be not included in the elite set at step 3 after several iterations.

4.1. GUNet3++ backbone architecture

The building block of the proposed evolutionary approach is the GUNet3++ network. The GUNet3++ architecture derives from the generalization of both UNet++ and UNet3+ networks. Looking at bottom of Figure 1, we can see that, with respect to UNet, the UNet++ network introduces dense skip-connections on each scale whereas UNET3+ does not use redesigned dense skip-connections but adopt, for each decoder, multi-scale feature aggregation.

The GUNet3++ architecture is modified to include both these aspects obtaining the structure depicted in Figure 4. As the figure shows, the architecture maintains the dense pyramidal block of transducers allowing to propagate, at each scale, information from shallow nodes to deeper ones. This is complemented with multi-scale skip-connections allowing the network to learn from contextual information across different regions of growing sizes at multiple scales.

Figure 3 shows, for instance, the generation of output signal of decoder $X^{2,2}$ using upscaling and downscaling to perform aggregation on that scale.

In this case, there are six feature maps to be unified. We used the convolution with 64 filters of size $3x3$ leading to a feature map of 384 filters. As

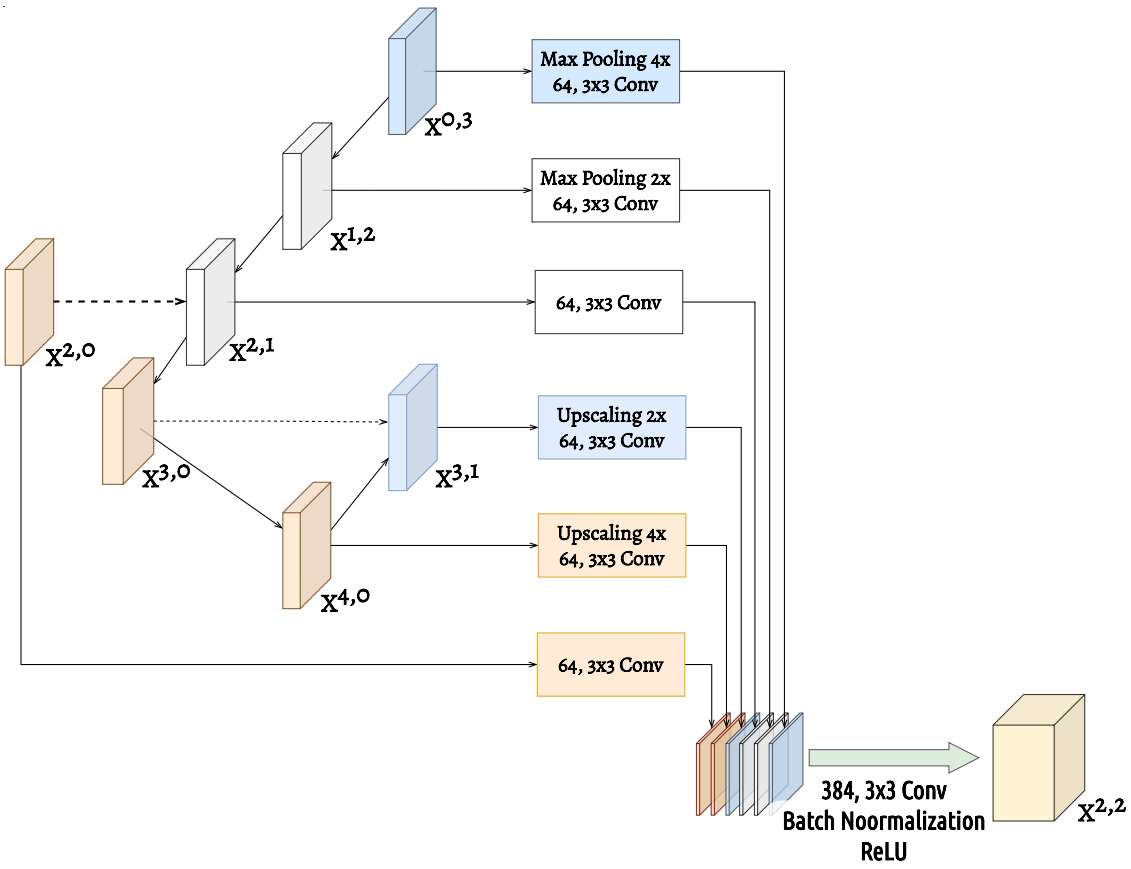


Figure 3: Aggregated feature map of decoder $X^{2,2}$.

shown in the figure, the third channels ($X^{2,0}$,$X^{2,1}$) are sent without scaling since they are at the same-resolution whereas the remaining are up-scaled or down-scaled accordingly to the source with respect to the destination.

Following the same notation of Section 3.2, the output of the node $X^{i,j}$, where i identifies the down-sampling layer on the encoders path, j identifies the convolution layer across the skip connection and $N + 1$ is the number of scales, is defined as:

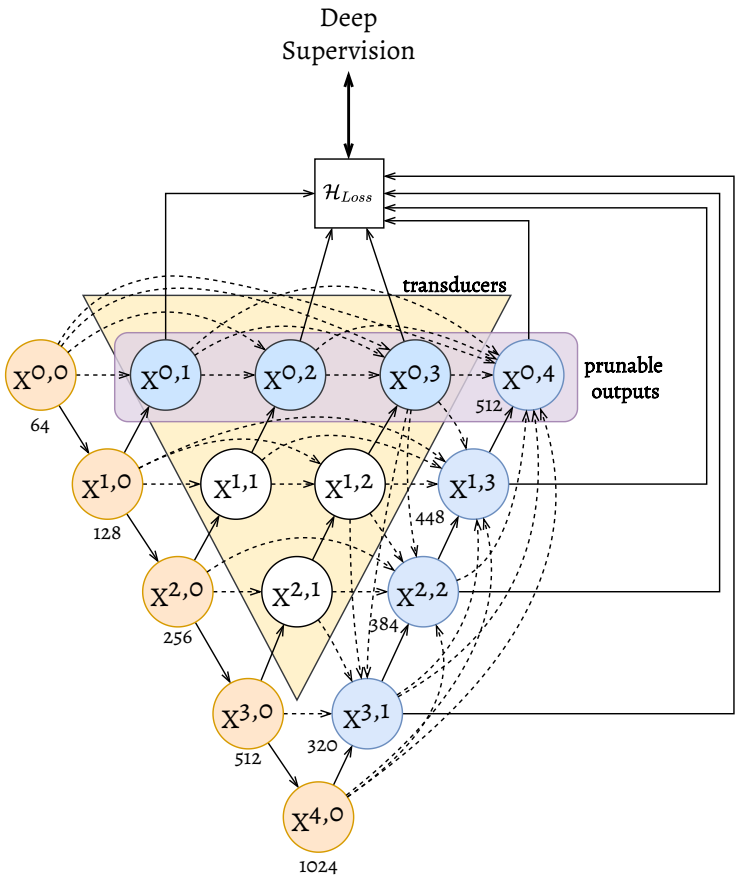


Figure 4: The GUNet3++ architecture.

$$x^{i,j} = \begin{cases} \mathcal{H}_c(\mathcal{D}\left(x^{i-1,j}\right)) & \begin{matrix} i>0 \\ j=0 \end{matrix} \\ \mathcal{H}_c\left(\left[[x^{i,k}]_{k=0}^{j-1}, \mathcal{U}(x^{i+1,j-1})\right]\right) & \begin{matrix} i\in[0,N/2] \\ j\in[1,N-i-1] \end{matrix} \\ \mathcal{H}_c\left(\left[[X^{i,k}]_{k=0}^{N-i-1}, \mathcal{C}\left(\mathcal{D}\left(X^{i-(k+1),j+k}\right)\right)_{k=0}^{N-1-j}, \right.\right. & \begin{matrix} i\in[0,N-1] \\ j=N-i \end{matrix} \\ \left.\left. \quad \mathcal{C}\left(\mathcal{U}\left(X^{i+(k+1),j-(k+1)}\right)\right)_{k=0}^{N-(i+1)}\right]\right) & \end{cases}$$

where $\mathcal{C}(\cdot)$ is the convolution operation, $\mathcal{H}_c(\cdot)$ is the convolution followed by the activation operation, $\mathcal{D}(\cdot)$ is the down-sampling operation, $\mathcal{U}(\cdot)$ represents the up-sampling operation, and $[\cdot]$ is the concatenation operator.

The network allows deep supervision at two levels. At maximum resolution, involving the decoded outputs $X^{0,j}$ with $j \in \{0, \ldots, N\}$, where outputs can also be pruned to improve inference times when needed. The deep supervision also can be performed across multiple scales, to learn hierarchical representations from feature maps of all the scales, at the deepest decoder segment (i.e., $X^{i,N-i}$ with $i \in \{1, \ldots, N-1\}$).

To train the network we then used a hybrid loss function combining pixel-wise cross-entropy loss and soft dice-coefficient loss, at each semantic scale. The overall loss function for GUNet3++ is defined as a weighted sum of the hybrid loss of each decoder:

$$\mathcal{H}_{Loss} = \sum_{i=1}^{D} \mathcal{H}_{Loss}(T, P^i)$$

where D is an index across decoders, (T, P^i) are the true labels and the labels evaluated by decoder i for every pixels in the batch, and $\mathcal{H}_{Loss}(T, P^i)$, defined as the sum of pixel-wise cross-entropy and dice-coefficient contribution, can be formulated as:

$$\mathcal{H}_{Loss}(T, P^i) = -1/S_P \sum_{c=1}^{K} \sum_{n=1}^{S_P} \left(l(x_n, c) * \log(p(x_n, c)) + \frac{2 * l(x, c) * p(x_n, c)}{l(x_n, c)^2 + p(x_n, c)^2} \right)$$

where, for class c and pixel x_n belonging to the current batch, $l(x_n, c) \in T$ are the target labels, $p(x_n, c) \in P^i$ are the predicted labels by decoder i, and S_P is the number of pixels within the batch.

4.2. GNet3++ building block and the encoding strategy

To represent a trainable model, starting from the original GUnet3++ template, an encoding scheme must be defined for both the hyper-parameters and the model structure.

To avoid having to define a distinct structure for each node, which would not have particular useful and would make the evolutionary process unnecessarily heavy, we divide the network into few distinct coding segments: therefore a node obtains an encoding which determines its structure on the basis of its position in the network.

Specifically, the GUNet3++ network has been divided in three segments: S_P to encode pyramidal transducers, S_E for encoders and S_D to represent decoders.

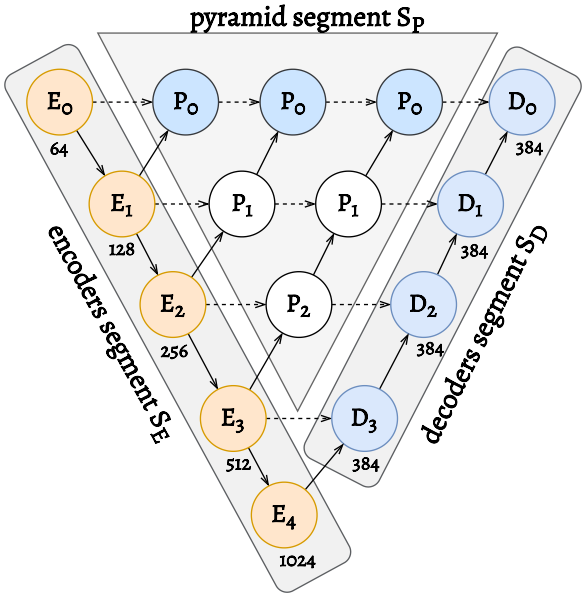


Figure 5: GUNet3++ encoding segments.

As shown in Figure 5 each segment defines the elements to be encoded in the overall architecture genotype: each element of $S_E \cup S_D$ (i.e., encoders and decoders) has an individual block gene to define its internal structure during the evolutionary search whereas for the pyramidal segment S_P each scale has several blocks sharing the same structure (i.e., one block gene for each scale, to reduce the search space).

Each gene is structured in two portions, one encoding the block structure in terms of nodes and connections and the other specifying the node structure in terms of performed operations, as depicted in the upper part of Figure 6.

The gene structure is replicated on each element over the three segments (S_E, S_D, S_P) to generate the overall genotype of the architecture (shown in the bottom part of Figure 6). This allows different blocks to have different operations and layout resulting in a very flexible architecture design. Concerning the block part of the encoding, it defines a graph (i.e., a DAG) specifying nodes and their connections. Each node represents an atomic operation sequence (defined in the node encoding), and the edges are linked among them generating a feature map from the node

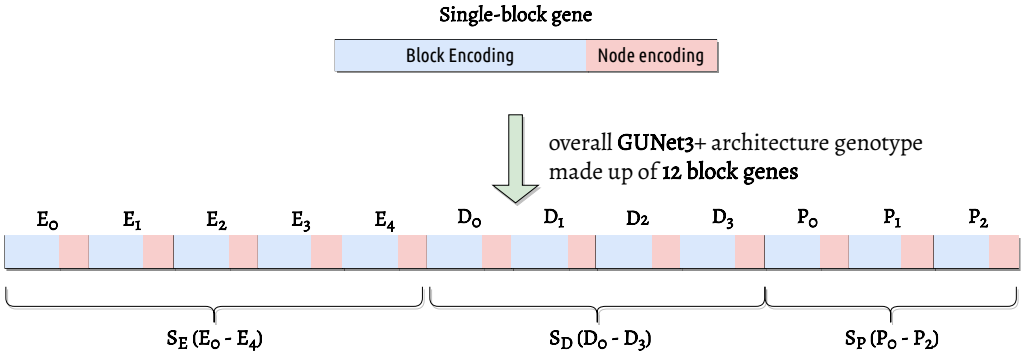


Figure 6: Encoding scheme and architecture genotype.

inputs. If the maximum number of nodes that is permitted in the DAG is N, then the binary digits used to encode the block is $N * (N-1)/2$.

Figure 7 reports two block-level encodings along with the resulting graphs for $N = 6$ (resulting in a block encoding of fifteen digits). As highlighted in the two examples 7-(a) and 7-(b), each digit specifies if an edge is present or not among the corresponding nodes. The maximum number of nodes is fixed same but, if connections generate isolated nodes, they are removed from the resulting blocks, like the node N_5 in the example (a). This makes it possible to generate blocks with a reduced number of nodes.

For what relates to the node-encoding part of each gene, it is used to specify the basic operation sequence to be used. The nodes in a standard network have a fixed sequence of operations (e.g., conv$(3 \times 3) \longrightarrow$ BNorm $\longrightarrow$ ReLU). Each individual in the evolutionary search can encode a different sequence of operations as specified in Table 1.

As the table shows, there are eight basic sequences of operations that adopted at the node-encoding level, resulting in a node encoding of three binary digits. The sequences of operations differ among themselves for kernel size of convolutional operations, for the used activation function, and for the batch normalization presence.

5. Experiment Description

In the following sub-sections, the research questions, the dataset constructed and the experiment evaluation setting are described.

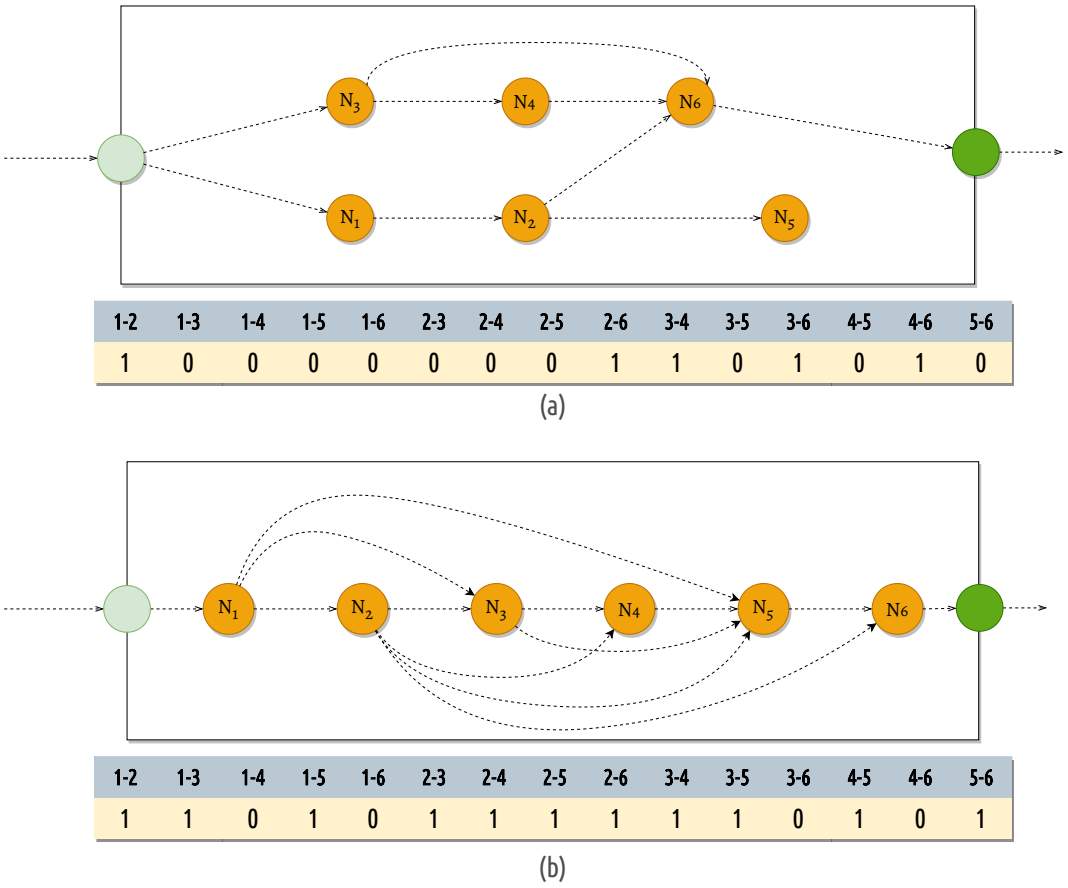


Figure 7: Two block-level encoding examples

5.1. Research questions

In this section, the high-level research goals discussed in the introduction are detailed through the following research questions:

RQ1: *Is the performance of the best fine-tuned GUNet3++ network found by the proposed genetic algorithm higher than the one obtained by baseline models?*

This research question aims to assess, evaluate and compare the performance of the proposed UNET-based variants derived using evolutionary algorithms in detecting lung cancer nodules in CT scans with respect to standard UNET variants used as baselines.

RQ2: *What is the impact of the ensemble-based correction at nodule-level on the lesion detection performance of the best fine-tuned GUNet3++?*

This research question aims at assessing and evaluating the performance

Encoding	Operations Sequence
000	conv(3×3) $\longrightarrow$ ReLU
001	conv(3×3) $\longrightarrow$ Swish
010	conv(3×3) $\longrightarrow$ BNorm $\longrightarrow$ ReLU
011	conv(3×3) $\longrightarrow$ BNorm $\longrightarrow$ Swish
100	conv(5×5) $\longrightarrow$ ReLU
101	conv(5×5) $\longrightarrow$ Swish
110	conv(5×5) $\longrightarrow$ BNorm $\longrightarrow$ ReLU
111	conv(5×5) $\longrightarrow$ BNorm $\longrightarrow$ Swish

Table 1: Node-level encodings (sequences of operations)

Cfg ID	Configurations			Hyperparameters			Performance Metrics		
	Encoder blocks	Decoder Blocks	Pyramidal Blocks	LR	BS	DO	Dice	SEN	PPV
1	5-acd-5-e5e-3-7f7-5-5bb-4-bdc	4-89f-3-1bd-1-34c-5-ff9	0-76f-7-777-3-c62	0.015	32	0.02	0.973	0.977	0.923
2	5-aec-7-e4c-3-adf-4-63f-4-2ce	0-89f-3-1bd-1-34c-5-ff9	0-76f-7-777-3-c62	0.010	32	0.02	0.965	0.980	0.921
3	5-acd-5-e5e-3-7f7-5-5bb-4-bdc	4-2db-3-4b5-0-34a-7-fe9	0-364-7-547-2-c6e	0.015	64	0.02	0.954	0.959	0.921
4	5-aee-3-65d-1-8df-0-2b7-0-a42	2-e3e-3-5b7-1-15c-5-b39	2-56c-5-5fb-7-f7e	0.015	32	0.02	0.946	0.962	0.910
5	5-ace-3-f45-3-1d7-5-1f6-3-24e	4-91f-3-e9d-4-34e-4-fd1	0-7ec-3-737-3-54b	0.015	64	0.02	0.929	0.954	0.909
6	1-2ad-2-ee4-7-217-4-63b-1-d7e	2-537-1-bb0-2-374-6-2f1	0-67f-7-79f-3-b86	0.015	32	0.02	0.925	0.935	0.892
7	2-6ec-2-064-7-2cf-0-61f-3-f7c	0-55f-1-ab0-2-878-4-b78	5-eff-2-6dd-3-5c6	0.015	64	0.02	0.916	0.933	0.873
8	1-a91-0-a8a-7-2b7-5-e53-5-d7e	6-550-1-3aa-0-7d5-6-7b1	1-778-6-c5b-3-bfa	0.010	64	0.02	0.912	0.937	0.880
9	5-aec-7-e4c-3-adf-4-63f-4-2ce	0-89f-3-1bd-1-2c8-7-0bd	0-76f-5-766-3-e52	0.150	64	0.02	0.910	0.936	0.893
10	5-e74-1-858-1-0ff-4-33d-0-73f	5-9ce-6-dbd-5-348-5-db9	4-36e-7-676-3-c51	0.200	64	0.02	0.908	0.915	0.874

Table 2: Performance of the ten best GUNET3++ configurations.

of the proposed approach when tested at the nodule-level using ensemble-based correction. To answer this question, an ablation study to investigate the impact of ensemble-based correction to reduce the false negatives is performed. The ensemble correction is performed by using the three best performing GUNet3++ networks. In this case, the ROIs are evaluated by averaging the ones produced by the single networks of the ensemble.

5.2. Dataset construction

In this experiment, we used an international publicly available dataset called Lung Image Database Consortium image collection (LIDC-IDRI) [29]. It is composed of images of digital radiography, computed radiography, and thoracic CT scans of 1010 patients. In particular, all the images are annotated after a double-check performed by four experienced radiologists to identify all the lung nodules contained in each image for a given

UNET 37.39M params								UNET++ 49.39M params								UNET3+ 26.39M params							
DO	BN	LR	Opt	BS	Dice	SEN	PPV	DO	BN	LR	Opt	DS	BS	Dice	SEN	PPV	DO	BN	LR	Opt	DS	BS	Dice	SEN	PPV
0.20	H	0.03	NADAM	32	0.728	0.807	0.708	0.30	H	0.04	SGD	Y	64.000	0.922	0.93	0.91	0.15	H	0.01	SGD	Y	64	0.934	0.936	0.918
0.20	H	0.03	SGD	64	0.724	0.800	0.677	0.25	H	0.02	SGD	Y	32.000	0.918	0.93	0.89	0.10	H	0.01	SGD	Y	64	0.932	0.937	0.897
0.20	T	0.02	NADAM	64	0.696	0.768	0.661	0.20	T	0.03	SGD	N	64.000	0.912	0.93	0.86	0.10	T	0.02	NADAM	Y	32	0.924	0.926	0.878
0.15	H	0.02	NADAM	64	0.665	0.738	0.611	0.30	H	0.02	NADAM	N	32.000	0.891	0.91	0.84	0.15	H	0.02	NADAM	N	32	0.902	0.908	0.870
0.25	N	0.01	RMSProp	32	0.661	0.733	0.605	0.25	T	0.02	SGD	Y	128.000	0.877	0.88	0.82	0.20	T	0.02	RMSProp	N	128	0.893	0.897	0.865

Table 3: Best results of baseline methods.

patient. Each nodule is identified when all four radiologists indicated the presence of a lesion having a diameter equal to or greater than 3 mm. Moreover, at each image, a set of metadata is associated indicating the characteristics of the contained nodules (if any). In particular, the description of the complete three-dimensional contour of the nodule is described by a metric called `Nodule Contour ROI`. A more extended description of this metric is reported in [29]. Starting from this data, three different datasets are obtained. The Nodules Oracle dataset is built considering all the images collected in the LIDC-IDRI database having the same format and resolution (BMP format with a resolution of 512×512 pixel). Moreover, a cleaning step is executed by removing all the low-quality scans and all the images corresponding to a patient having a reduced or incomplete number of CT images.

After the cleaning, we obtained three datasets for training, validation and test respectively relative to 715 patients. The training dataset is composed of 32,606 images of 500 patients with their corresponding metadata. The same images are also contained in the CT scan training dataset, wherein 28,2027 images are labeled as not containing nodules and the rest as containing nodules. The validation test included 71 additional patients (representing 10% of the total number of patients). Finally, the test dataset contains the images of the remaining 143 patients (representing 20% of the total number of patients) and is used for the assessment.

In the training phase, an Images Masks Generation step is performed. This allows obtaining, from the ROI provided in the DICOM files of the dataset, a set of images masks (a mask for each image of the original dataset representing all the nodules specified in the oracle or an empty mask for images with no nodules). The masks are then used to train the U-Net-based classifier that takes as input the original CT scans images and as output the corresponding masks learning to perform semantic segmentation of nodules.

5.3. Experimental setting

The goal of the proposed experimental validation spans over three levels of assessment:

- to evaluate the capability of each GUNet3++ network to detect the RoI of a lesion in a CT-scan (at image-level, investigated in RQ1);
- to evaluate the effectiveness in identifying complete nodules across different images of a CT-scan using window-based correction (at nodule-level, investigated in RQ2);

Within the first point, that involve image-level assessment, different variants of GUNet3++ and different hyperparameters configurations are evaluated. In addition to the evolutionary search that defines the concrete structure of the network, the considered hyperparameters are the dropout rate (D), learning rate (LR), and batch size (BS). The dropout rate represents the probability of training a given node in a layer and in this study belongs to the interval $[0.15, 0.35]$ with a step of 0.10. The learning rate indicates the step size at each training iteration while moving toward the minimum of the loss function. In this study, it belongs to the interval $[0.01, 0.02]$. The batch size represents the number of training samples used in one iteration of update of the neural network internal parameters. We consider in this study three batch sizes, namely 16, 32, and 64.

Considering a nodule as a sequence of images representing its different regions, the nodule-level assessment generalizes the image-level assessment to evaluate if a nodule is correctly identified as a whole. At this level, we use a window-based correction approach: the resulting masks provided by three networks are used to generate an improved RoI performing a correction pass during the detection.

Finally, to validate the model we have adopted a metric that is usually used to evaluate segmentation: the sørensen-Dice similarity coefficient, which measures the similarity between two samples and is based on presence and absence data [30], and is similar to the Intersection over Union (IoU) metric, a good metric for measuring the overlap between two bounding boxes or masks [31].

Defined G as the correct mask and P as the segmentation generated by the network, the Dice can be defined as:

$$\text{Dice} = \frac{2 * |G \cap P|}{|G| + |P|}$$

To validate pixel classification performance and correctness of the segmentation area we adopted the sensitivity (SEN) and the positive predictive value (PPV) metrics, which are defined as:

$$\text{SEN} = \frac{|G \cap P|}{|G|}$$

$$\text{PPV} = \frac{|G \cap P|}{|P|}$$

All the single classifiers have been developed using PyTorch[1], Tensorflow[2], and Keras[3], three open-source neural network libraries with Python bindings (used to implement the classifiers). The genetic algorithm is implemented in Java, using the Jenetics[4] open-source library

For this experimentation the following two workstations:

- AMD Ryzen Threadripper 3960X 24-Core, with 128GB of RAM and two GPU NVIDIA RTX 3090 (with 24Gb of RAM)
- Intel Core i9 9940X (14 cores), with 64GB of RAM and four GPU NVIDIA Tesla T4 (with 16Gb of RAM)

have been used.

6. Results and Discussion

RQ1: *Is the performance of the best fine-tuned GUNet3++ network found by the proposed genetic algorithm higher than the one obtained by baseline models?*

Table 2 shows the performance of the best GUNET3++ configurations. The first column of the table reports a configuration ID. The following columns contain respectively information about the encoder, decoder, and pyramidal blocks. Finally, the considered hyperparameters and the performance metrics are described in the last columns. The yellow row of the table shows the best configuration reaching very good performance (the

[1] https://pytorch.org/
[2] https://www.tensorflow.org/
[3] https://keras.io/
[4] https://jenetics.io

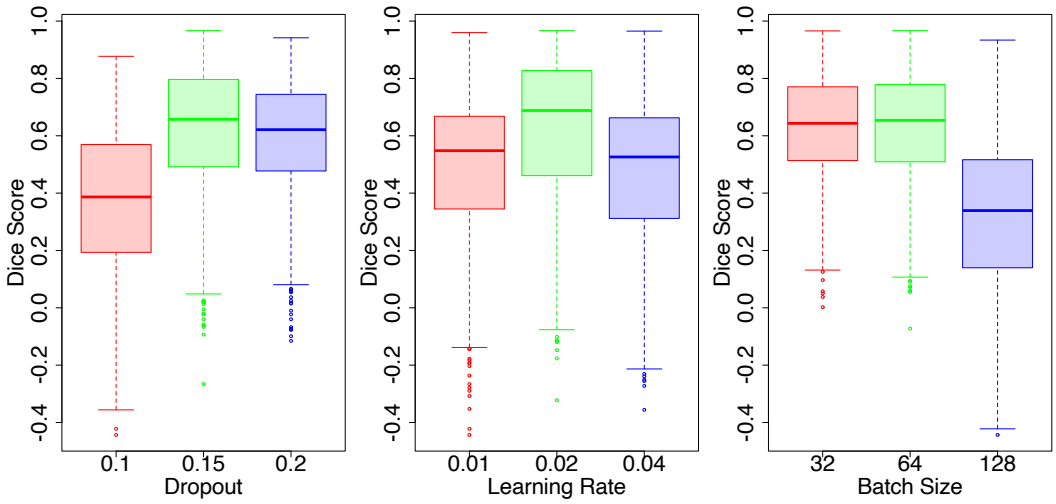


Figure 8: Hyper-parameters influence on the average dice score.

Dice is equal to 0.972, the SEN is equal to 0.977 and the PPV is equal to 0.923). It is interesting to investigate the properties of the best encoding reported in Table 2 looking at Table 1 that shows the operation sequences for each encoding. Specifically, we can observe that the top five configurations are all characterized by an encoder block having conv$(5 \times 5) \longrightarrow$ Swish at node level. Similarly, the top two configurations are also characterized by a conv$(5 \times 5) \longrightarrow$ BNorm $\longrightarrow$ Swish operations sequence for the pyramidal blocks. This suggests that initial and internal paths that extract and aggregates features at different scales (i.e., encoding and pyramidal sections) benefit most from the use of larger convolutions, Swish activation function and batch normalization with respect to decoder blocks which prefer smaller filters often with a simple ReLU. Further considerations can be made about the influence of each considered hyper-parameter (dropout, learning rate, and batch size) on the performance of the GUNet3++.

Figure 8 reports the average Dice score obtained for different values of dropout, learning rate, and batch size. In all the cases, we can observe that the Dice score is influenced by the considered hyper-parameters values. For the dropout, the best scores are obtained when the value is fixed to 0.15 while the worst dice score is obtained when the dropout is 0.1. Similarly, a greater dice score is obtained when the learning rate is 0.02 and the batch size is 64 and 32 (this is also confirmed by Table 2 that shows that in all the

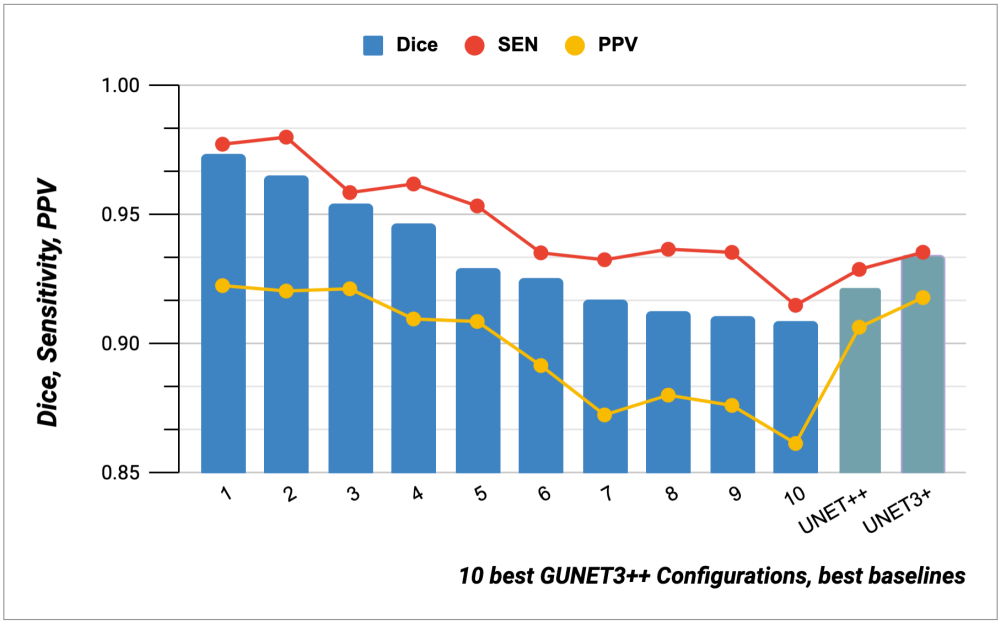


Figure 9: Performance comparison of GUNET3++ with baselines.

best configurations the DO is equal to 0.02 and the BS is 32 or 64).

Table 3 reports the best results obtained for different baseline methods and different hyperparameters configurations.

The table shows that good performance is always obtained for UNET++ and UNET3+ (the Dice, SEN, and PPV are always greater than 0.82). Lower values are then obtained by the UNET (in the worst case the PPV is equal to 0.605). The yellow row of the table shows the best configuration for each considered variant. In particular, the highest values of Dice, SEN, and PPV are obtained for UNET3+ (they are respectively 0.934, 0.936, 0.918). Finally, a comparison between Table 3 and Table 2 shows that the performance of the GUNET3++ is improved with respect to the performance of the baseline methods. The performance comparison of GUNET3++ with the baseline is highlighted in Figure 9. The figure reports the Dice, SEN, and PPV values for the top ten GUNET3++ and the best UNET++ and UNET3+. The figure shows that the Dice values of the best four GUNET3++ configurations are always better than the UNET3+ best Dice. Moreover, six of the

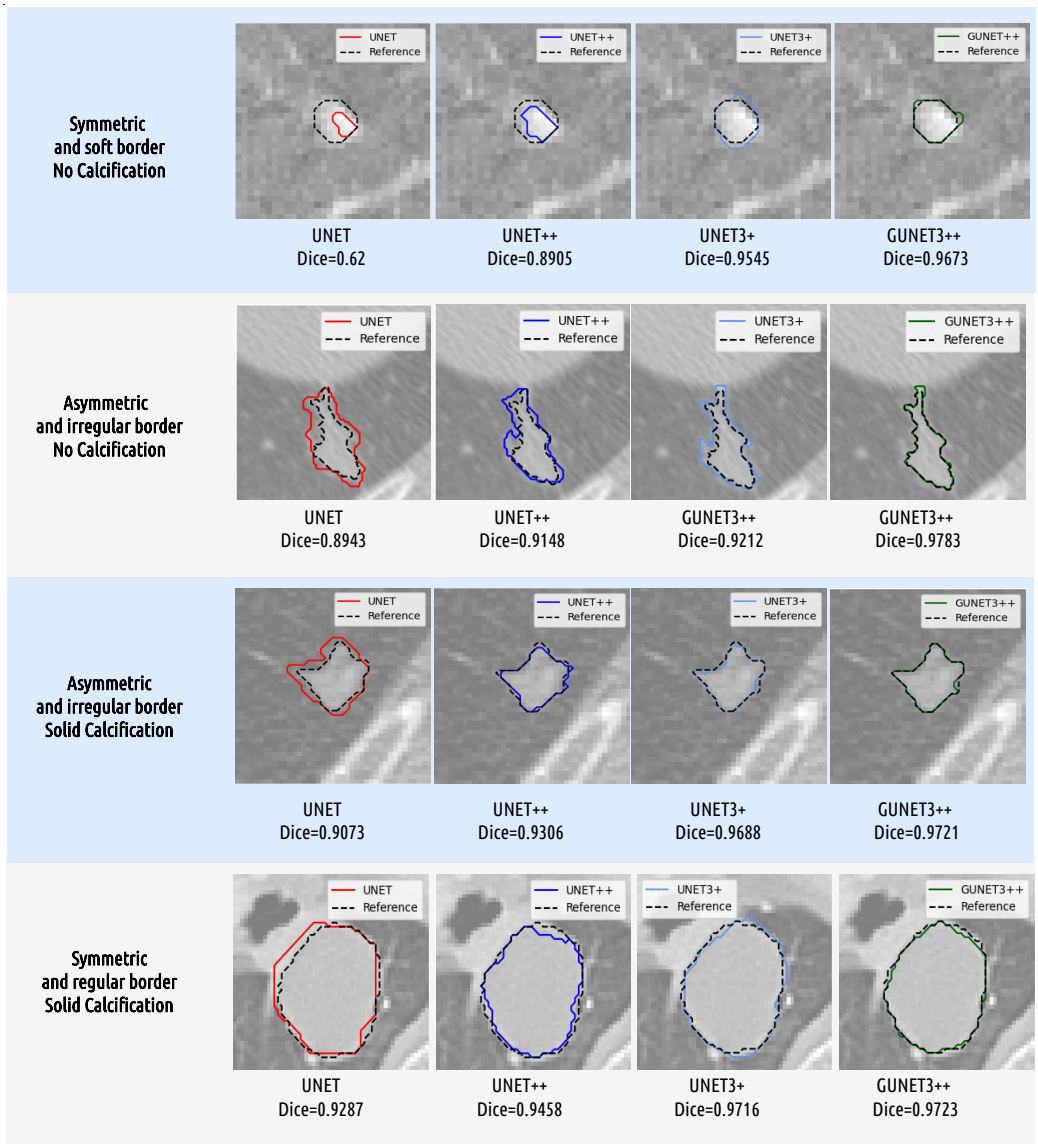


Figure 10: Qualitative analysis of predicted ROI for GUNET3++ and baselines for different kind of lesions.

ten considered GUNET3++ give better Dice of the UNET++ best configuration. Similar considerations can be made for the SEN and PPV values. The performance of the GUUNet3++ with respect to the baselines is also explored by considering a different kinds of lesions. Figure 10 reports in

each row four different kinds of lesions. For each row, the figure shows the predicted ROI (the lesion border is represented with a colored line) and the reference ROI (dotted line). Each column of the image reports a different method (UNET, UNET++, UNET3+, GUNET3++) so that a different color of the lesion border represents a different adopted method. Looking at the figure, the first row reports the kind of lesion characterized by symmetric and soft borders and no calcification. The lesion border obtained by the UNET (it is colored in red), is quite different from the reference border. The best matching between the reference ROI and the predicted ROI is then obtained by the GUNET++ (it is colored in green). Good matching is also obtained by UNET3+ and UNET++ (clear and dark blue borders). These results are also confirmed by the Dice values obtained for each method and reported in the figure. These considerations can be extended to the other rows of the figure. We can observe that the best matching between predicted ROI and reference ROI (Dice=0.978) is obtained when the lesion has an asymmetric and irregular border and there is no calcification. Generally, good results for all the considered methods are obtained when the lesion is symmetric and has a regular border with solid calcification. Finally, despite the described advantages of the proposed approach with respect to the baseline models, it is worth noting that the GUNet3++ network is characterized by multi-scale skip connections requiring more effort for both training and inference times. However, the impact of these additional connections on the inference times is particularly high. Specifically, inference times remain comparable with times exhibited by the other segmentation networks. As a consequence, even if the training process of GUNET3++ is slower and more demanding, it allows to obtain more precise segmentation networks that are characterized by inference times similar to the ones of the comparable segmentation networks.
RQ2: *What is the impact of the ensemble-based correction at nodule-level on the lesion detection performance of the best fine-tuned GUNet3++?*

The impact of ensemble-based correction to reduce the false negatives is then evaluated to answer this question. Figure 11 reports (in green) the estimation of the nodule volume of the proposed approach when tested at the nodule level using ensemble-based correction. The curve is compared with the corresponding values obtained by the reference slices (black line) and the best GUNet3++ (red line). The figure highlights that the ensemble-based correction allows reducing the curve oscillations with increased sim-

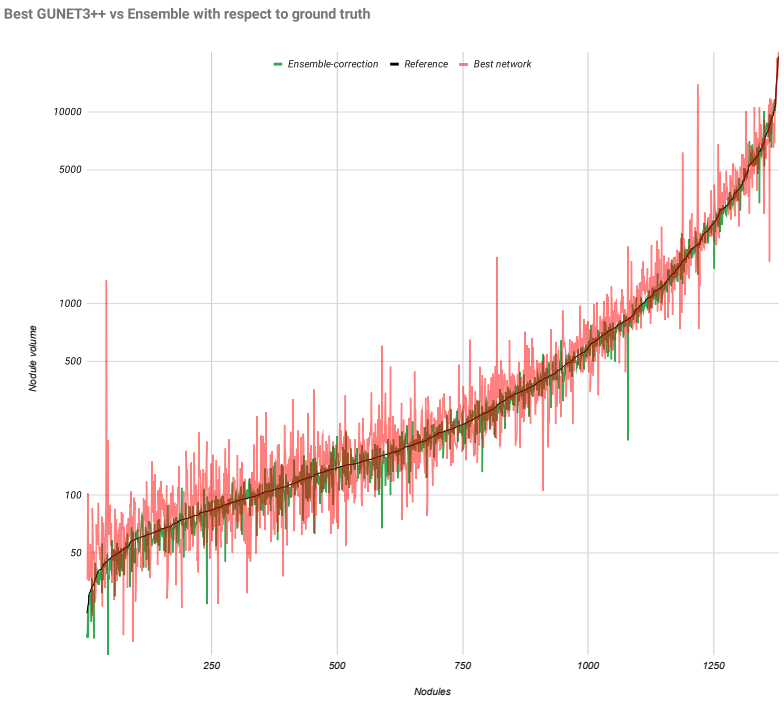


Figure 11: Estimated nodules volume (by size) using the single best GUNET3++ selected by the genetic algorithm (i.e., the 'Best-network' curve) and using the best three trained networks (i.e., the 'Ensemble-correction' curve) compared to the one evaluated by radiologists (i.e., the 'Reference' curve).

ilarities between the estimated nodules volume and the reference model. This is also confirmed by looking at the boxplots depicted in Figure 12 showing the distribution of the differences among the predicted volumes and the reference ones evaluated using the ground truth. As we can see the estimated values corrected using the ensemble of the best three networks found by the evolutionary algorithm have a smaller inter-quartile range compared to the single best GUNET3++ network. This means that the values are much more focused on the real values (meaning a much more precise ROI of lesions found on nodules slices). It is interesting also that using ensemble of different configurations allows to build a more robust segmenter. This is because different "best" configurations behave in a slightly different way, making errors on different kinds of nodules (and

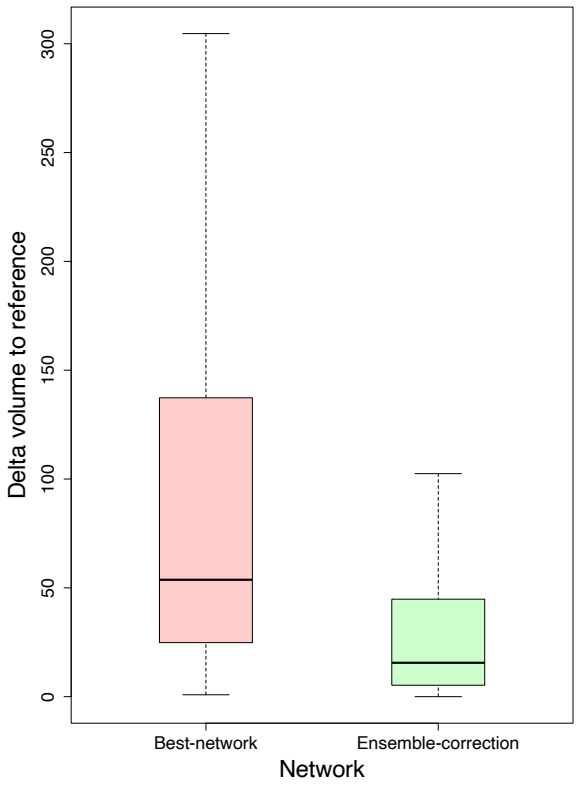


Figure 12: Boxplot of delta volumes distributions for GUNET3++ and best ensemble.

this makes ensembling beneficial).

We conclude the section by reporting, in Table 4, a comparison between the training times of the various networks on the considered dataset. This allows you to have purchasing information on the complexity of the entire workflow. From this point of view, as could be assumed from the highest number of parameters, GUNET3++ is the network that requires the greatest training effort (approximately double the time of UNET++ and 5 times the training time of UNET3+). However, it is interesting to note that it is easier for the evolutionary optimization process to find GUNET3+ networks that provide better performance with respect to other networks (since GUNET3+ structure can be altered using segments encodings). In fact, if we look at the global process training times, the GUNET3++ training times still dominate but with a smaller ratio (three times slower than

	Single Epoch Training Average	Single Model Training Average	Total Training including HPO
UNET 37.39M params	8	201	8516
UNET++ 49.39M params	15	1119	68482
UNET3+ 26.39M params	11	358	27902
GUNET3++ 61.82M params	25	1919	77268

All training times (sec) are obtained on a set of 4 NVIDIA V100 40GB on LIDC-IDRI dataset

Table 4: Training times in seconds for a single epoch, a single model and the total training time of the hyper-parameter optimization.

UNET3+ and substantially equivalent to UNET++).

7. Threats to the validity

In this study there are three main threats to validity: construct validity, internal validity, and external validity.

As regards the construct validity threats, a possible limitation could be due to the quality of the adopted dataset. However, this risk is mitigated by the used dataset that has been largely adopted and referenced in medical and engineering studies. Moreover, the dataset labeling has been performed through an established process involving four radiologists.

In addition, in the proposed study, the dataset has been pre-processed cleaning, filtering, and deleting all the images with low quality and/or a different format. Referring to the internal validity threats, a limitation could be due to possible variables that are not considered in our experiment but can influence our observations. However, in this study, we split the CT scan dataset into a test (20% of the data) and a training set (80% of the data). We can not be sure that different splits could give different results.

Similarly, detection performances are strongly influenced by the adopted network architectures (this is also confirmed by the discussed results). For this reason, we are aware that architectures not considered in this study

could dive into different performances. However, to mitigate this threat four different networks have been considered and evaluated in this study.

Finally, the threats to external validity regard the capability to generalize the obtained outcomes. Even if the considered dataset includes a high number of images and patients, it is necessary to further evaluate the proposed approach on additional datasets including images of different image resolution, color and format.

8. Conclusions

Several studies have been recently proposed to deal with the lung cancer detection problem. The relevance of the topic is due to the importance of early diagnosis for people affected by lung cancer to increase survival rates. In this paper, we have proposed an innovative approach that aims at effectively supporting the diagnosis of patients affected by lung cancer using CT scan images. According to this, the approach exploits an evolutionary algorithm to build variants of a UNet-based architecture, called GUNet3+, to detect patients affected by lung cancer, from the analysis of CT-scan images. The approach is defined considering tomography images of the lung. To validate the approach, a large and well-known dataset has been used. The obtained results are very encouraging and provide further perspectives. However, a limitation of this study can be the lack of interpretability and transparency in the decision criteria which decreases the possibility of using the approach in real-world practical cases. This limitation is quite common to all the AI prediction models applied to the clinical context [32]. According to this, as future work, we aim to integrate into our proposed approach a GradCAM++ component able to identify the lung regions where the most relevant features involved in classification have been extracted [7]. Another limitation of this study regards the necessity to include in the experimentation more larger and heterogeneous datasets to better generalize the obtained results. However, with respect to other fields, in the medical domain the collection of medical images, is generally challenging. Medical images can be generated in different ways using different tools and obtaining medical imaging is expensive as well as patients' privacy needs to be ensured [33]. In future work, the proposed approach will be further experimented with using new datasets to generalize the obtained results also thanks to the collaboration with medical institutions. Referring to the limitations discussed in Section 7, the

approach will be extended to include also heterogeneous data (image resolution, color, and format), not considered variables (different splits), and a greater number of networks will be considered as alternative ones.

References

[1] IARC, Estimated age-standardized mortality rates (world) in 2020, worldwide, both sexes, all ages, https://gco.iarc.fr/today/data/factsheets/cancers/39-All-cancers-fact-sheet.pdf, [Online; accessed 01-February-2021] (International Agency For Research on Cancer, 2020).

[2] IARC, Estimated number of new cases from 2020 to 2040, Both sexes, age [0-85+] Trachea, bronchus and lung, https://gco.iarc.fr/tomorrow/en/dataviz/bars?mode=population&cancers=15, [Online; accessed 01-February-2021] (International Agency For Research on Cancer, 2020).

[3] WHO, Knowledge into Action Cancer Control - WHO Guide for Effective Programmes, https://www.who.int/cancer/modules/Early%20Detection%20Module%203.pdf, [Online; accessed 04-February-2021] (World Health Organization, 2020).

[4] J. L. Causey, Y. Guan, W. Dong, K. Walker, J. A. Qualls, F. Prior, X. Huang, Lung cancer screening with low-dose ct scans using a deep learning approach (2019). arXiv:1906.00240.

[5] C. White, T. Flukinger, J. Jeudy, J. J. Chen, Use of a computer-aided detection system to detect missed lung cancer at chest radiography., Radiology 252 1 (2009) 273–81.

[6] R. Gruetzemacher, A. Gupta, D. Paradice, 3d deep learning for detecting pulmonary nodules in ct scans, Journal of the American Medical Informatics Association 25 (2018) 13011310.

[7] G. Rani, A. Misra, V. S. Dhaka, D. Buddhi, R. K. Sharma, E. Zumpano, E. Vocaturo, A multi-modal bone suppression, lung segmentation, and classification approach for accurate covid-19 detection using chest radiographs, Intelligent Systems with Applications 16 (2022) 200148. doi:https://doi.org/10.1016/j.iswa.

2022.200148.
URL https://www.sciencedirect.com/science/article/pii/S2667305322000850

[8] M. W. Vannier, A. El-Baz, G. M. Beache, G. Gimel'farb, K. Suzuki, K. Okada, A. Elnakib, A. Soliman, B. Abdollahi, Computer-aided diagnosis systems for lung cancer: Challenges and methodologies, International Journal of Biomedical Imaging 2013 (2013) 942353. doi:10.1155/2013/942353.
URL https://doi.org/10.1155/2013/942353

[9] M. Javaid, M. Javid, M. Z. U. Rehman, S. I. A. Shah, A novel approach to cad system for the detection of lung nodules in ct images, Computer Methods and Programs in Biomedicine 135 (2016) 125–139. doi:https://doi.org/10.1016/j.cmpb.2016.07.031.
URL https://www.sciencedirect.com/science/article/pii/S0169260715303977

[10] K.-H. Yu, T.-L. M. Lee, M.-H. Yen, S. C. Kou, B. Rosen, J.-H. Chiang, I. S. Kohane, Reproducible machine learning methods for lung cancer detection using computed tomography images: Algorithm development and validation, J Med Internet Res 22 (8) (2020) e16709. doi:10.2196/16709.
URL https://www.jmir.org/2020/8/e16709

[11] D. Riquelme, M. A. Akhloufi, Deep learning for lung cancer nodules detection and classification in ct scans, AI 1 (1) (2020) 28–67. doi:10.3390/ai1010003.
URL https://www.mdpi.com/2673-2688/1/1/3

[12] P. Ardimento, L. Aversano, M. L. Bernardi, M. Cimitile, Deep neural networks ensemble for lung nodule detection on chest ct scans, in: 2021 International Joint Conference on Neural Networks (IJCNN), 2021, pp. 1–8. doi:10.1109/IJCNN52387.2021.9534176.

[13] L. Aversano, M. L. Bernardi, M. Cimitile, R. Pecori, Deep neural networks ensemble to detect COVID-19 from CT scans, Pattern Recognit. 120 (2021) 108135. doi:10.1016/j.patcog.2021.108135.
URL https://doi.org/10.1016/j.patcog.2021.108135

[14] Y. Gu, J. Chi, J. Liu, L. Yang, B. Zhang, D. Yu, Y. Zhao, X. Lu, A survey of computer-aided diagnosis of lung nodules from ct scans using deep learning, Computers in Biology and Medicine 137 (2021) 104806. doi:https://doi.org/10.1016/j.compbiomed.2021.104806.
URL https://www.sciencedirect.com/science/article/pii/S0010482521006004

[15] S. Das, S. Majumder, Lung cancer detection using deep learning network: A comparative analysis, in: 2020 Fifth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), 2020, pp. 30–35. doi:10.1109/ICRCICN50933.2020.9296197.

[16] N. Nasrullah, J. Sang, M. S. Alam, M. Mateen, B. Cai, H. Hu, Automated lung nodule detection and classification using deep learning combined with multiple strategies, Sensors 19 (17). doi:10.3390/s19173722.
URL https://www.mdpi.com/1424-8220/19/17/3722

[17] A. A. A. Setio, F. Ciompi, G. Litjens, P. Gerke, C. Jacobs, S. J. van Riel, M. M. W. Wille, M. Naqibullah, C. I. Sánchez, B. van Ginneken, Pulmonary nodule detection in ct images: False positive reduction using multi-view convolutional networks, IEEE Transactions on Medical Imaging 35 (5) (2016) 1160–1169. doi:10.1109/TMI.2016.2536809.

[18] W. Zhu, C. Liu, W. Fan, X. Xie, Deeplung: Deep 3d dual path nets for automated pulmonary nodule detection and classification, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), 2018, pp. 673–681. doi:10.1109/WACV.2018.00079.

[19] A. Masood, P. Yang, B. Sheng, H. Li, P. Li, J. Qin, V. Lanfranchi, J. Kim, D. D. Feng, Cloud-based automated clinical decision support system for detection and diagnosis of lung cancer in chest ct, IEEE Journal of Translational Engineering in Health and Medicine 8 (2020) 1–13. doi:10.1109/JTEHM.2019.2955458.

[20] A. Hatamizadeh, Y. Tang, V. Nath, D. Yang, A. Myronenko, B. Landman, H. R. Roth, D. Xu, Unetr: Transformers for 3d medical image

segmentation, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022, pp. 574–584.

[21] D. Yang, A. Myronenko, X. Wang, Z. Xu, H. R. Roth, D. Xu, T-automl: Automated machine learning for lesion segmentation using transformers in 3d medical imaging, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021, pp. 3962–3974.

[22] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention MICCAI, 2015.

[23] R. Su, D. Zhang, J. Liu, C. Cheng, Msu-net: Multi-scale u-net for 2d medical image segmentation, Frontiers in Genetics 12 (2021) 140. doi:10.3389/fgene.2021.639930.
URL https://www.frontiersin.org/article/10.3389/fgene.2021.639930

[24] N. Siddique, S. Paheding, C. P. Elkin, V. Devabhaktuni, U-net and its variants for medical image segmentation: A review of theory and applications, IEEE Access 9 (2021) 82031–82057. doi:10.1109/ACCESS.2021.3086020.

[25] M. G. B. Calisto, S. Lai-Yuen, Adaresu-net: Multiobjective adaptive convolutional neural network for medical image segmentation, Neurocomputing 392 (2020) 325–340.

[26] E. Shelhamer, J. Long, T. Darrell, Fully convolutional networks for semantic segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (4) (2017) 640–651. doi:10.1109/TPAMI.2016.2572683.

[27] U. Baid, S. Talbar, S. Rane, S. Gupta, M. H. Thakur, A. Moiyadi, N. Sable, M. Akolkar, A. Mahajan, A novel approach for fully automatic intra-tumor segmentation with 3d u-net architecture for gliomas, Frontiers in Computational Neuroscience 14 (2020) 10. doi:10.3389/fncom.2020.00010.
URL https://www.frontiersin.org/article/10.3389/fncom.2020.00010

[28] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, J. Liang, Unet++: Redesigning skip connections to exploit multiscale features in image segmentation., IEEE transactions on medical imaging 39 (6) (2019) 1856–1867.
URL https://neuro.unboundmedicine.com/medline/citation/31841402/UNet++:_Redesigning_Skip_Connections_to_Exploit_Multiscale_Features_in_Image_Segmentation_

[29] I. Lütkebohle, LIDC-IDRI - The Cancer Image Archive (TCIA) Public Access, https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI/, [Online; accessed 30-January-2021] (2014).

[30] A. Carass, S. Roy, A. Gherman, J. C. Reinhold, A. Jesson, T. Arbel, O. Maier, H. Handels, M. Ghafoorian, B. Platel, A. Birenbaum, H. Greenspan, D. L. Pham, C. M. Crainiceanu, P. A. Calabresi, J. L. Prince, W. R. G. Roncal, R. T. Shinohara, I. Oguz, Evaluating white matter lesion segmentations with refined sørensen-dice analysis, Scientific reports 10 (1) (2020) 8242–8242, 32427874[pmid]. doi:10.1038/s41598-020-64803-w.
URL https://pubmed.ncbi.nlm.nih.gov/32427874

[31] F. van Beers, A. Lindström, E. Okafor, M. A. Wiering, Deep neural networks with intersection over union loss for binary image segmentation, in: ICPRAM, 2019.

[32] M. Nazari, A. Kluge, I. Apostolova, S. Klutmann, S. Kimiaei, M. Schroeder, R. Buchert, Explainable ai to improve acceptance of convolutional neural networks for automatic classification of dopamine transporter spect in the diagnosis of clinically uncertain parkinsonian syndromes, European Journal of Nuclear Medicine and Molecular Imaging 49 (4) (2022) 1176–1186. doi:10.1007/s00259-021-05569-9.
URL https://doi.org/10.1007/s00259-021-05569-9

[33] Y. Said, A. A. Alsheikhy, T. Shawly, H. Lahza, Medical images segmentation for lung cancer diagnosis based on deep learning architectures, Diagnostics 13 (3). doi:10.3390/diagnostics13030546.
URL https://www.mdpi.com/2075-4418/13/3/546