Biomedical Signal Processing and Control 78 (2022) 103996

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Spatial feature fusion in 3D convolutional autoencoders for lung tumor segmentation from 3D CT images

Suhail Najeeb *, Mohammed Imamul Hassan Bhuiyan

Department of Electrical and Electronic Engineering, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

ARTICLE INFO

Keywords:
Segmentation
CT scan
Lung tumor
Convolutional autoencoders
Deep learning

ABSTRACT

Accurate detection and segmentation of lung tumors from volumetric CT scans is a critical area of research for the development of computer aided diagnosis systems for lung cancer. Several existing methods of 2D biomedical image segmentation based on convolutional autoencoders show decent performance for the task. However, it is imperative to make use of volumetric data for 3D segmentation tasks. Existing 3D segmentation networks are computationally expensive and have several limitations. In this paper, we introduce a novel approach which makes use of the spatial features learned at different levels of a 2D convolutional autoencoder to create a 3D segmentation network capable of more efficiently utilizing spatial and volumetric information. Our studies show that without any major changes to the underlying architecture and minimum computational overhead, our proposed approach can improve lung tumor segmentation performance by 1.61%, 2.25%, and 2.42% respectively for the 3D-UNet, 3D-MultiResUNet, and Recurrent-3D-DenseUNet networks on the LOTUS dataset in terms of mean 2D dice coefficient. Our proposed models also respectively report 7.58%, 2.32%, and 4.28% improvement in terms of 3D dice coefficient. The proposed modified version of the 3D-MultiResUNet network outperforms existing segmentation architectures on the dataset with a mean 2D dice coefficient of 0.8669. A key feature of our proposed method is that it can be applied to different convolutional autoencoder based segmentation networks to improve segmentation performance.

1. Introduction

Lung cancer is one of the most common forms of cancer and is the most threatening in terms of mortality. In 2020, lung cancer was the leading cause of cancer-related deaths with around 1.80 million deaths worldwide [1]. Lung cancer is accompanied by weight loss, fatigue, chronic cough, and chest pain, causing unthinkable suffering to the patient. Lung cancer occurs due to the uncontrolled growth of cells in the lung. The rapidly dividing cells accumulate and form lung masses or tumors which might be visible in a radiological investigation. Lung cancer diagnoses are classified into two main groups: small cell lung cancer (SCLC), and non-small cell lung cancer (NSCLC). Approximately 85% of all cancer diagnoses are non-small cell lung cancer [2]. Physical symptoms are often absent or similar to respiratory infections during the early stages of lung cancer. Due to the late onset of symptoms and lack of screening programs, most of the patients are diagnosed with an advanced stage of lung cancer [3]. The 5-year survival rate of lung cancer is an alarmingly low 18% [4], which is largely due to the delayed diagnosis. Diagnostic approaches like X-ray, CT and PET imaging, and histological examination of tumor biopsies can be used for lung cancer [2]. Early diagnosis of lung cancer can drastically reduce the mortality rate. A study of the National Lung Screening Trial (NLST) [5] revealed that early diagnosis using low-dose CT scans resulted in a 20% reduction in mortality from lung cancer [3]. However, several challenges remain in lung cancer diagnosis and screening. There are several steps involved in the diagnosis and treatment of lung cancer. The delineation of the tumor volume is usually performed by an expert radiologist and this step is essential for the proper detection of cancerous tumors present in the lung as well as for the next steps in treatment such as a biopsy, therapy, or surgery. This is a difficult, time-consuming, and error-prone task [6]. Especially in developing and underdeveloped countries, resources and manpower are scarce. This results in numerous cases of lung cancer going undiagnosed. To this end, computer-aided tools for automatic detection and segmentation of lung cancer might be able to assist medical professionals by greatly simplifying and speeding up the diagnostic process.

1.1. Related work

Traditional computer-aided approaches to lung cancer detection involved thresholding, morphological operations, connected component analysis, and other image processing techniques [7,8]. These

* Corresponding author.
E-mail address: suhail.najeeb@ieee.org (S. Najeeb).

https://doi.org/10.1016/j.bspc.2022.103996
Received 6 May 2022; Received in revised form 19 June 2022; Accepted 9 July 2022
Available online 19 July 2022

techniques were not fully automated and involved multiple steps for different tasks, for example — segmentation of the Lung ROI (Region of Interest) through thresholding, image processing operations for lung nodule detection, followed by a rule-based technique to classify cancer [9]. Some researchers took a two-step approach, where image enhancement and segmentation were followed by extracting tumor features [10] to detect lung cancer. Later Aerts et al. [11] and Lambin et al. [12] introduced rigorous radiomics analysis for comprehensive quantification of tumors from image features. Following these findings, machine learning algorithms like EK-Means clustering [13], Support Vector Machines (SVMs) [14], etc. were used to detect lung cancer from extracted features. While these techniques can detect the existence of lung tumors to a certain degree, there is a large variance in the tumor appearance [13], and other masses like lesions, nodules, the clavicles, and the heart present in the scans make it difficult to accurately segment lung tumor volumes using image processing techniques alone. This calls for the utilization of more advanced techniques like deep learning.

Deep learning [15] makes use of deep networks, which are computational models composed of multiple layers. Deep networks can learn complex representations of data at multiple levels of abstraction. Deep learning methods have made notable breakthroughs in speech recognition, visual object recognition, object detection, and many other domains like drug discovery and genomics as well [15]. More recently, deep learning has seen success in various medical image processing applications. For example, artificial neural networks [16] and deep learning [17] have been utilized for the detection of lung cancer from CT images. Segmentation of the tumor volume in lung cancer patients is also an important task. Unlike detection, which is the task of identifying the presence of a disease, segmentation is a more challenging task, where the goal is to identify the exact location of the tumor. Early approaches to image segmentation have used pixel-wise segmentation with fully convolutional networks [18]. However, this approach was not efficient and suitable for biomedical segmentation tasks. The U-Net network by Ronneberger et al. [19] revolutionized the field of biomedical image segmentation by outperforming all previous approaches of the time. The U-Net network consists of an encoder–decoder-based convolutional neural network architecture and made strong use of data augmentation. This architecture has been the cornerstone of biomedical image segmentation and several biomedical segmentation networks like the ResUNet++ [20], MultiResUNet [21], DRINet [22], etc. improved upon UNet. Dilated fully convolutional neural networks [23] have also been utilized for semantic segmentation of biomedical scans. Most of these networks have taken a two-dimensional approach to the task of biomedical segmentation and achieved respectable performance.

However, many biomedical images, for example, CT scans and MR images are volumetric scans consisting of several continuous slices along the transverse anatomical plane, calling for a three-dimensional segmentation problem. Two-dimensional approaches to biomedical segmentation only consider one slice at a time, therefore they are not able to process the spatial context along the missing dimension. There have been several approaches to address this issue. A three-dimensional version of the UNet network [24] has been introduced for learning dense volumetric segmentation from sparsely annotated data. Milletari et al. [25] introduced a fully convolutional autoencoder architecture for 3D segmentation of prostate MRI volumes. Chen et al. [26] introduced deep voxel-wise residual networks for brain segmentation from 3D MR images. However, utilizing the full 3D volume with a convolutional neural network is very difficult due to the exponential increase in compute and memory requirements. At the same time, networks become very large and require more training data, which is often scarce. To address these issues, these networks either take small patches of voxels instead of the full scan [26] or downsample the input to the network [25] to reduce computation costs. Taking small patches of voxels leads to the same limitation of losing spatial context. When performing volumetric segmentation of larger anatomical regions like the heart, the brain, lungs, kidneys, the prostate, etc. it is possible to downsample the scans without losing too much context. However, for the task of lung-tumor segmentation where fine pixel-level features of smaller tumors are important, downsampling may lead to loss of data. 3D segmentation of lung tumors from volumetric scans is, therefore, a challenging task.

The development, training, and evaluation of a computer-assisted diagnostic method for lung cancer detection and diagnosis require carefully annotated biomedical data which is not widely available. There have been some recent initiatives for the curation and collection of biomedical data for such applications. The Lung Image Database Consortium (LIDC) and the Image Database Resource Initiative (IDRI) have compiled a database [27] of low-dose lung CT scans from 1010 patients. The 2016 LUNA16 challenge [28] included scans from this dataset to develop an automated CAD system for the automatic detection of pulmonary nodules in CT images. The 2017 Kaggle Data Science Bowl challenge [29] aimed at improving lung cancer detection from CT images. These competitions saw a multitude of deep learning approaches to address tasks like lung nodule segmentation and classification. The 2018 IEEE VIP Cup challenge [30] involved lung tumor region segmentation on CT Scans from the NSCLC-Radiomics Dataset [31]. In conjunction with the IEEE VIP Cup challenge, the Lung-Originated Tumor Segmentation (LOTUS) Benchmark [32] was created to provide a unique dataset for lung tumor segmentation. Several approaches like a 3D Dilated Convolutional Neural Network [33], Recurrent-3D-DenseUNet [34], Deeply Supervised MultiResUNet [35] etc. were developed on this benchmark and showed promising performance on the dataset.

1.2. Our contributions

For the task of lung tumor segmentation on the LOTUS Benchmark [32], two-dimensional networks like UNet [19] and its variants [35] show respectable performance for segmenting tumors. On the other hand, it is difficult to train an effective 3D Segmentation network from scratch. In this paper, we try to address this issue by making use of the learned features at different levels of the 2D networks to enhance the performance of 3D segmentation networks. There have been a few attempts in the literature [33,36] to combine segmentation features or results from both the 2D and 3D domains. However, in this paper, we take a novel approach to utilize the spatial features learned at different levels of the segmentation network while maintaining feature space balance. The 3D segmentation networks have also been slightly modified to accommodate the changes without introducing any significant computational complexity. Our proposed approach can be incorporated into existing convolutional autoencoder-based 3D segmentation networks to enhance performance. We applied our proposed methodology to different segmentation networks like the UNet [19,24], MultiResUNet [21], Recurrent-3D-DenseUNet [34], etc. without incorporating any major changes to the underlying architecture. We were able to record performance improvements of 1.61%, 2.25%, and 2.42% in terms of 2D dice-coefficient respectively for our proposed modified versions of the 3D-UNet, 3D-MultiResUNet, and Recurrent-3D-DenseUNet networks. The best performing network is our proposed SFF-3D-MultiResUNet architecture which achieved a mean dice-coefficient of 0.8669 outperforming all previous approaches on the dataset [33–35]. We have also introduced the 3D dice coefficient to evaluate the performance of volumetric segmentation, which has not been reported by earlier approaches on this dataset. In terms of the 3D dice coefficient, our proposed networks respectively achieved 7.58%, 2.32%, and 4.28% improvement in performance compared to their baselines and the best-performing SFF-3D-MultiResUNet model achieved a 3D dice score of 0.5938 which is significantly higher than the other baseline models.

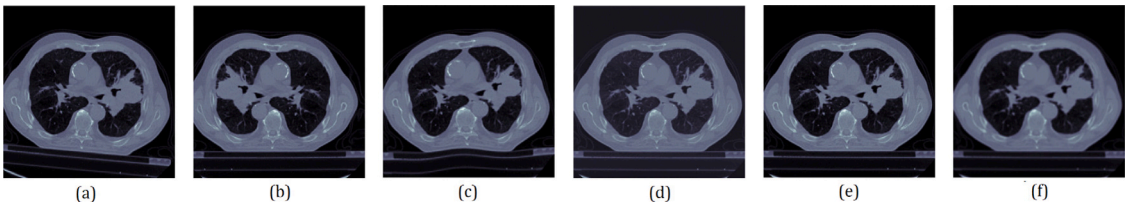

Fig. 1. Illustration of different augmentations on a training sample: (a) Random rotation, (b) Horizontal flip, (c) Random elastic deformation, (d) Random contrast normalization, (e) Random noise, (f) Blurring.

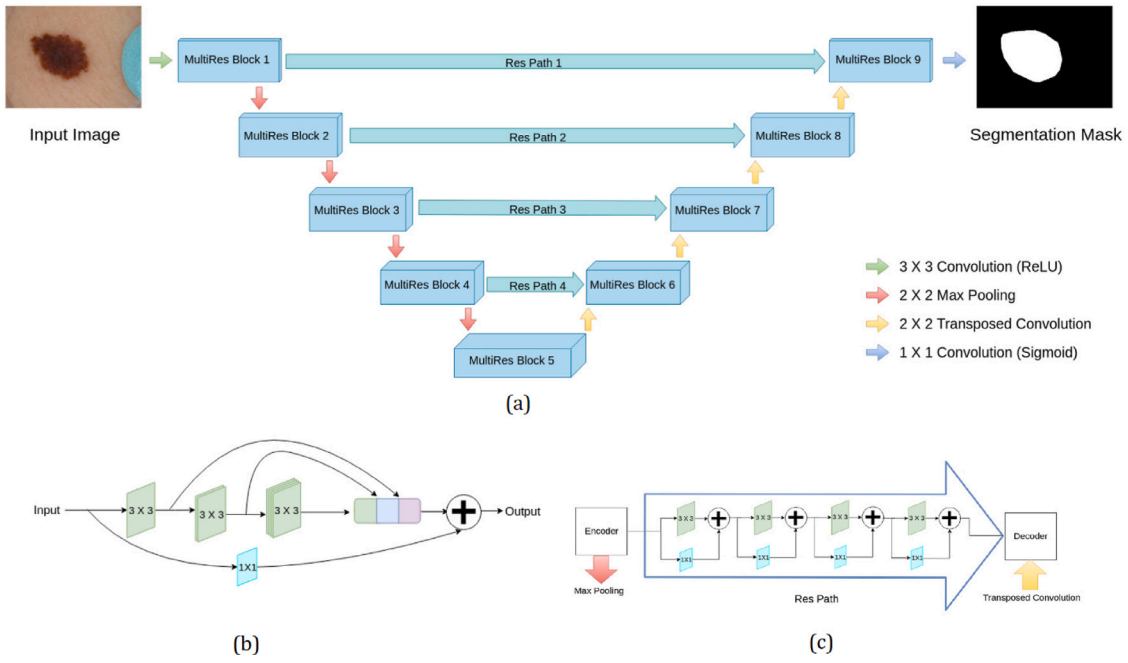

Fig. 2. MultiResUNet architecture, figure adapted with permission from [21]. (a) Network architecture, (b) MultiRes block, (c) Res path.

2. Materials and methods

2.1. Dataset

The dataset used in this paper is the LOTUS Benchmark [32] prepared as part of the IEEE VIP Cup 2018 Challenge [30]. The LOTUS dataset is a modified version of the NSCLC-Radiomics black-dataset [31]. The dataset contains Computed Tomography (CT) scans of 300 lung cancer patients which are provided in DICOM format. The CT scanners are from two different sources — Siemens and CMS Imaging Inc. 3D volumes consisting of a varying number of slices are provided for each patient. The 2D slices for the scans each have a resolution of 512×512. Annotations for the Gross Tumor Volume (GTV), Clinical Target Volume (CTV), and Planning Tumor Volume (PTV) provided by an expert radiologist are available. The LOTUS benchmark involves the segmentation of the GTV. For training and evaluation, the dataset is divided into two sets identical to the approach of [33–35]. The training and testing sets contain 260 and 40 scans respectively. 10% of the training data is kept as a validation set. A detailed summary of the dataset is presented in Table 1.

2.2. Data preprocessing

The dataset is provided in DICOM format. The PyDicom [37] library was utilized to read the DICOM scans as well as the annotations for the lung tumor volumes. However, there are discrepancies in the Hounsfield Unit (HU) values of the scans associated with different manufacturers. Scans from CMS Imaging Inc. have HU values from −1024 to 3071 whereas scans from Siemens have HU values between 0 and 4095. The HU values were adjusted to account for this discrepancy and scaled between 0 and 1. The slices are then resized using bilinear interpolation to a resolution of 256×256 to reduce the GPU memory requirements while training the deep neural networks.

Table 1
Dataset statistics for the LOTUS benchmark [32].

Dataset	Patients	CT scanner		Number of slices	
		CMS imaging Inc.	Siemens	Tumor	Non-tumor
Train	260	60	200	4296 (13.7%)	26 951 (86.3%)
Test	40	34	6	848 (18.9%)	3610 (81.1%)

As we can see from Table 1, a large portion of the slices does not contain any tumors. To enable the 2D segmentation networks to better learn the features associated with tumors, only the slices which contain tumors are taken from the 3D volume to train the 2D segmentation networks. While training the 3D segmentation networks, stacks of eight consecutive slices are taken to perform 3D volumetric segmentation. Stacks that contain at-least one tumor-containing slice are considered for training.

2.3. Data augmentation

Medical image analysis applications are heavily reliant on various data augmentation techniques to address the problem of overfitting due

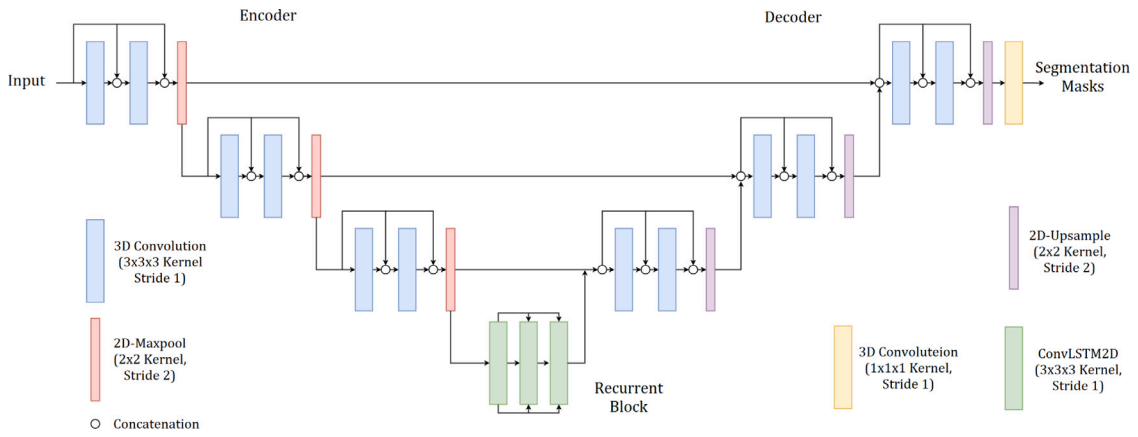

Fig. 3. Recurrent-3D-DenseUNet architecture [34].

to limited training data. There are mainly two approaches for data augmentation — transforming the data beforehand and storing it in memory, or using generators to transform data on-the-fly [38]. Wang et al. [39,40] utilized the former approach and performed a multi-way data augmentation on chest CT images. Their proposed approach yielded impressive results for the task of Covid-19 diagnosis. However, this approach might introduce difficulty when working with larger 3D datasets in our case, depending on how heavily the dataset size has been inflated. Due to memory constraints, only a finite amount of augmented data can be stored, introducing finite variability in the training data. A separate preprocessing step will also add to the total training time.

Data generators that randomly apply one or more augmentation methods from a pool of predefined transformations on-the-fly do not require a separate preprocessing step or massive memory requirements. Since on-the-fly methods apply a random number of transformations with random parameters for each sample, any sample is highly unlikely to be similar to another and thus introduces very high variability in the training data. Up until recently, these methods were considered to slow down the training process due to computational overhead. However, modern computers with parallel computing are capable of handling data augmentation on-the-fly. For example, Isensee et al. [41] utilized an on-the-fly data augmentation scheme which has shown excellent segmentation performance in a wide range of biomedical segmentation tasks. Our proposed data augmentation method introduces additional transformations and randomly performs one or more of the following transformations on the fly — horizontal flips, random rotations, random elastic deformations, random noise addition, random contrast normalization, blurring, etc. Fig. 1 illustrates different types of augmentations on a training sample.

2.4. Baseline model architectures

2.4.1. UNet

The UNet [19] network is a convolutional autoencoder architecture. The original UNet network consists of four levels of contracting paths (encoder) and four levels of expansive paths (decoder). Each step in the contracting path consists of successive convolution operations followed by max-pooling. For the expansive path, each step contains upsampling convolution operations. The results are concatenated via skip connections from the contracting path, followed by operations similar to the encoder blocks. Before the sigmoid activation at the final layer, there is a final convolution operation to generate the segmentation mask.

2.4.2. MultiResUNet

The MultiResUNet [21] architecture is a modified version of the UNet architecture. It replaces the convolutional blocks with the 'MultiRes Block' (Fig. 2b) which consists of multiple convolutional operations followed by concatenation and a shortcut connection. The skip connections of the UNet network are replaced with the 'Res Path' (Fig. 2c) which contains multiple convolutions and shortcut connections to promote residual learning. Fig. 2a shows a brief overview of the MultiResUNet architecture.

2.4.3. Recurrent-3D-DenseUNet

The Recurrent-3D-DenseUNet [34] network follows the autoencoder architecture inspired by UNet. This is a 3D segmentation network that can perform segmentation on 3D volumes of stacked 2D scans. The encoder and decoder blocks contain several 3D-convolutional layers that are densely connected. However, instead of 3D pooling, 2D max-pooling is performed. The decoder block also performs 2D upsampling in place of 3D upsampling, which is followed by the 3D Dense Convolutions. The network makes use of a recurrent block consisting of several ConvLSTM layers to make use of the inter-slice continuity. The network architecture is illustrated in Fig. 3.

2.5. Proposed methodology

2.5.1. 2D autoencoders

Fig. 4 illustrates the basic structure of a 2D autoencoder architecture with three levels of contracting and expanding paths. Both paths follow successive 2D convolutions followed by either a downsampling or upsampling operation. Let, x be the 2D input image and x_1, x_2, x_3, b be the corresponding outputs at different levels of the encoder. The encoder blocks are defined as E_1, E_2, E_3, B and the Decoder blocks are defined by D_1, D_2, D_3. P denotes the pooling operation, and U_1, U_2, U_3 define the up-sampling convolution operations. O denotes the 1×1 convolution of the final layer. The intermediate outputs at different levels of the encoder are defined by y_1, y_2, y_3, and y denotes the final output (resulting segmentation mask). The following equations express the architecture of an autoencoder network with three contracting/expanding paths:

$$\begin{aligned} x_1 &= E_1(x) \\ x_2 &= E_2(P(x_1)) \\ x_3 &= E_3(P(x_2)) \\ b &= B(P(x_3)) \end{aligned} \tag{1}$$

$$\begin{aligned} y_3 &= D_3(x_3 + U_1(b)) \\ y_2 &= D_2(x_2 + U_2(y_3)) \\ y_1 &= D_1(x_1 + U_3(y_2)) \\ y &= O(y_1) \end{aligned} \tag{2}$$

Here, the encoders (E_1, E_2, E_3, B) perform successive convolutions on a 2D input CT Scan slice. As we know, convolutional layers detect local conjunctions of features from previous layers while the pooling layers merge semantically similar features, and higher-level features are

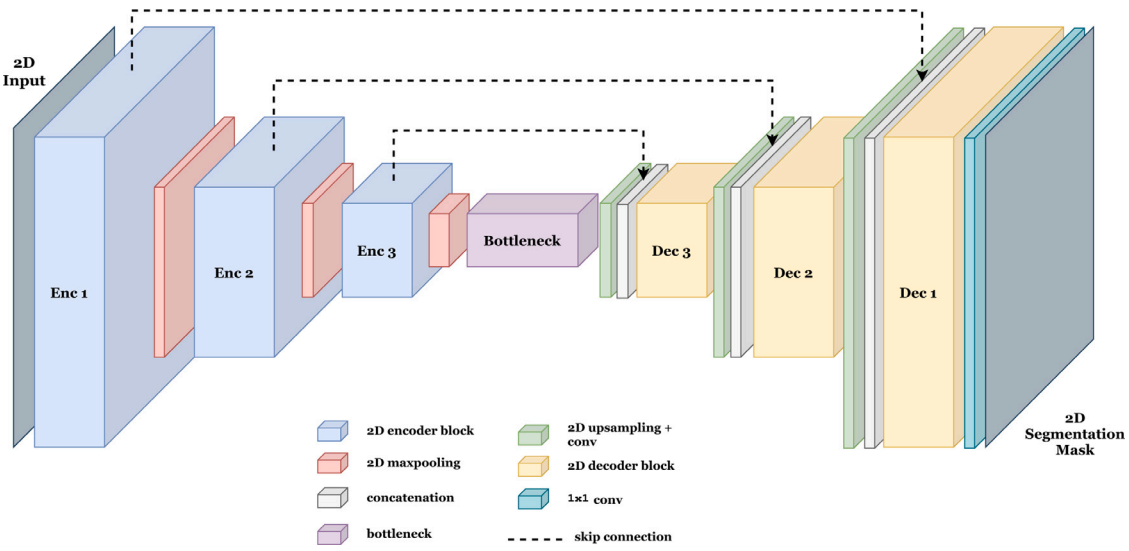

Fig. 4. Basic architecture of a 2D autoencoder.

obtained by composing lower-level ones [15]. Based on this notion, if the 2D segmentation networks can successfully segment lung tumors from 2-Dimensional CT scan slices, we can deduce that the first half of the network ($x \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow b$) has learned to extract successively higher levels of semantic features which are used by the second half of the network to gradually perform segmentation at different levels and generate the final segmentation map y at the same level of the input scan x. The primary intuition behind our proposed approach is that we could be able to utilize the learned feature representations (x_1, x_2, x_3, b) at different levels of the 2D autoencoder networks to aid the 3D segmentation process.

2.5.2. 3D autoencoders

A significant limitation of 2D segmentation networks for the task of volumetric segmentation is the inability to consider inter-slice relations and volumetric features. A simple approach to address this issue is to extend the 2D autoencoder architectures to their 3D counterparts. This can be achieved like [24] by replacing the 2D convolution, 2D pooling, and 2D upsampling operations with their 3D counterparts. 3D operations are more resource-intensive compared to their 2D counterparts and segmenting the entire 3D volume at once is computationally expensive. Patch-wise segmentation networks [26] have been introduced to solve this issue but not without limitations as discussed earlier. 3D convolution operations require more parameters compared to 2D convolutions. As a result, the 3D versions of existing 2D segmentation networks quickly become computationally complex when working with large 3D volumes. The resulting networks contain more hyper-parameters which increase the probability of overfitting [42]. Deep networks are heavily reliant on big data to avoid overfitting [38]. Therefore, more complex models will require more training data to further combat overfitting. However, medical data being scarce makes the task more difficult. Our proposed approach aims to solve this issue by utilizing the spatial feature representations learned from the 2D segmentation networks to better generalize an identical 3D segmentation network.

Certain modifications are required to allow the integration of spatial features learned from the 2D networks with the 3D architectures. If the 3D segmentation networks employ 3D max-pooling operations, it halves the spatial resolution along all axes at each encoder level. The 2D feature representations are incompatible with a network employing 3D max-pooling since the 2D networks perform 2D max-pooling along the axial plane only, which will lead to a mismatch in dimension along the longitudinal axis. In order to make the 3D networks compatible with the spatial features (i.e. have the same spatial resolution), it is essential that a similar maxpooling strategy is applied in both networks. Also, lung tumor thickness along the longitudinal axis is often small. Maxpooling along that axis may lead to the loss of valuable information relevant to the segmentation of the tumor volume. Therefore, we choose to use 2D max-pooling along the axial plane of the 3D scans instead of 3D max-pooling. This allows for the preservation of more information along the longitudinal plane and at the same time makes the network compatible with the 2D feature representations.

The decoders in the autoencoder essentially have the opposite responsibility of the encoders, which is to gradually upscale the outputs from different levels to generate the final segmentation mask. Corresponding to the 2D max-pooling operations in the encoder, 2D upsampling convolution operations are performed in the decoder instead of 3D max-pooling. Before implementing our proposed spatial feature fusion networks, we modified the baseline 2D networks (2D-UNet, 2D-MultiResUNet, etc.) to their 3D versions with the above changes to accommodate 2D feature fusion.

Say, a CT scan is a set of n consecutive slices and s_i is a single 2D slice along the axial plane, where $i \in [0, n]$. We consider stacks of d scans for segmentation, where d is the depth of the scans along the longitudinal axis. Here, the input to the 3D network is:

$$x = [s_i, s_{(i+1)}, \ldots, s_{(i+d)}], \text{where } i \in [0, n-d] \tag{3}$$

Similarly, if the 2D segmentation masks are denoted by m_i, then the target data for the 3D network can be expressed as:

$$y = [m_i, m_{(i+1)}, \ldots, m_{(i+d)}], \text{where } i \in [0, n-d] \tag{4}$$

2.5.3. Spatial feature extraction

As discussed in Section 2.5.1, the 2D segmentation networks can be used to extract relevant spatial feature maps at different stages of the network. We can define a spatial feature extraction network $\boldsymbol{\Phi}$, which takes the input 2D slice s_i and returns the relevant feature maps $s_{i1}, s_{i2}, s_{i3}, \ldots$ etc. If instead of a single slice s_i, we pass a stack of 2D scans (x) to the feature extraction network, we can get a stack of the features at different levels of the network ϕ_1, ϕ_2, ϕ_3. Here,

$$\phi_1, \phi_2, \phi_3 = \boldsymbol{\Phi}(x), \text{where } x = [s_i, s_{i+1}, \ldots, s_{i+d}] \tag{5}$$

$$\phi_j = [s_{ij}, s_{(i+1)j}, \ldots, s_{(i+d)j}], \tag{6}$$

where j is the level of the feature map

In the inset of Fig. 5, we can see the architecture of the 2D feature extraction network which is a cut-down version of the 2D segmentation network (Fig. 4) only containing the encoder blocks.

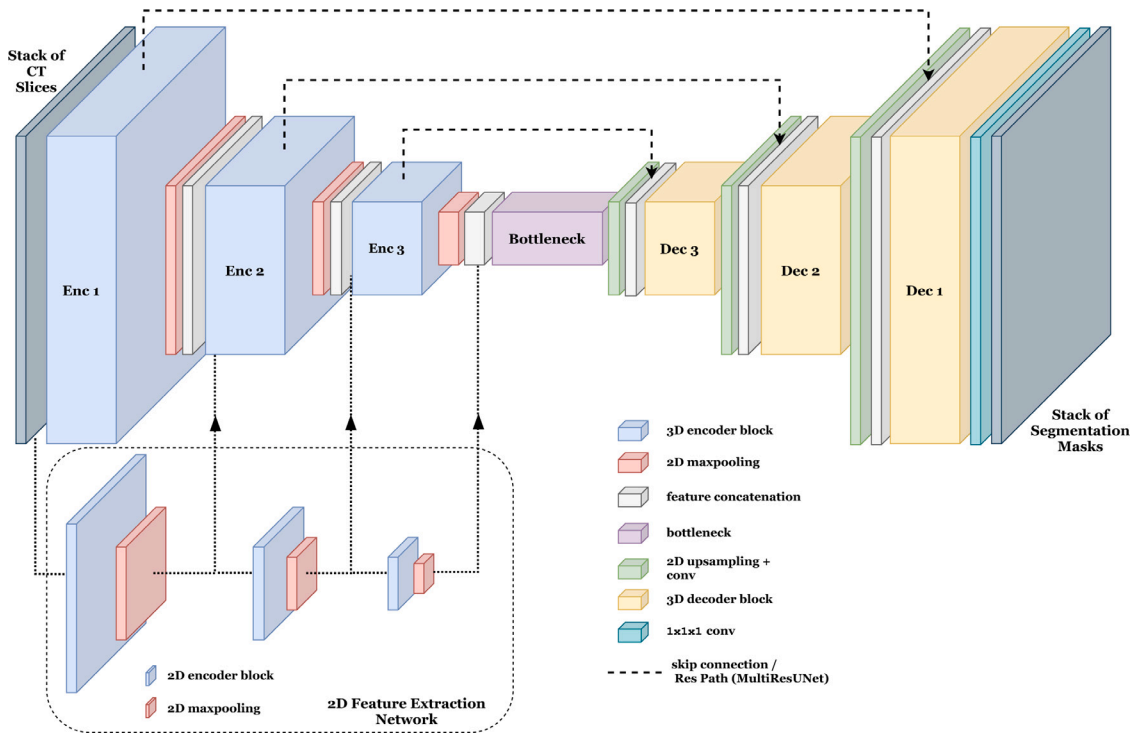

Fig. 5. Architectural diagram of the proposed spatial feature fusion network.

Table 2
Summary of proposed models based on spatial feature fusion.

Proposed model	Based on	Modifications	2D feature extraction network
SFF-3D-UNet	2D-UNet	3D convolutions in place of 2D convolutions. 2D max-pooling and 2D upsampling convolutions used for scaling up/down. Spatial feature fusion at encoders.	2D-UNet with 3 levels pretrained on 2D data
SFF-3D-MultiResUNet	2D-MultiResUNet	Similar modifications to SFF-3D-UNet (2D pooling/upsampling). Modified 3D 'MultiRes' for encoder/decoder blocks and 3D 'ResPath' instead of skip connections. Spatial feature fusion at encoders.	2D-MultiResUNet with 3 levels pretrained on 2D data
SFF-Recurrent-3D-DenseUNet	Recurrent-3D-DenseUNet	No major modifications required. Spatial feature fusion added at encoders.	2D-DenseUNet (proposed) with 3 levels pretrained on 2D data

2.5.4. Proposed spatial feature fusion networks

The proposed approach for spatial feature fusion consists of a 3D autoencoder network, which also makes use of the 2D feature maps produced in the previous step. The input to the network is denoted x, where $x = [s_i, s_{(i+1)}, \ldots, s_{(i+d)}]$, and ϕ_1, ϕ_2, ϕ_3 are the spatial feature maps at different levels of the network produced by the feature extraction network Φ. The proposed network concatenates the spatial features with the corresponding features at different levels of the 3D network (x_1, x_2, x_3). Let, the 3D encoder blocks be defined as $E_{3D_1}, E_{3D_2}, E_{3D_3}, B_{3D}$ and the 3D decoder blocks be $D_{3D_1}, D_{3D_2}, D_{3D_3}$. P_{2D} defines the 2D max-pooling operation and $U_{2D_1}, U_{2D_2}, U_{2D_3}$ determine the 2D upsampling convolution operations. If O_{3D} is the final 1 × 1 × 1 3D convolutional layer, then the network is mathematically expressed as follows:

$$\begin{aligned} x_1 &= E_{3D_1}(x) \\ x_2 &= E_{3D_2}(P_{2D}(x_1) + \phi_1) \\ x_3 &= E_{3D_3}(P_{2D}(x_2) + \phi_2) \\ b &= B_{3D}(P_{2D}(x_3) + \phi_3) \\ y_3 &= D_{3D_3}(x_3 + U_{2D_1}(b)) \\ y_2 &= D_{3D_2}(x_2 + U_{2D_2}(y_3)) \\ y_1 &= D_{3D_1}(x_1 + U_{2D_3}(y_2)) \\ y &= O_{3D}(y_1) \end{aligned} \tag{7}$$

Fig. 5 shows a brief overview of the proposed spatial feature fusion networks. Our proposed method has been used to create three architectures — the SFF-3D-UNet architecture, the SFF-3D-MultiResUNet architecture, and the SFF-Recurrent-3D-DenseUNet Architecture. A brief summary of the proposed models based on spatial feature fusion is shown in Table 2.

Since there is no compatible 2D version of the Recurrent-3D-DenseUNet network, we had to design a 2D counterpart for this network from scratch to use as a feature extraction network. A 2D-UNet network was modified to contain dense connections in the encoder and decoder blocks. ConvLSTM blocks are not used since they are not relevant for 2D images. A densely connected encoder block is used as the bottleneck instead. We call this the 2D-DenseUNet network. This network is used to extract the relevant features for use in the SFF-Recurrent-3D-DenseUNet architecture.

2.6. Training setup

All deep learning models are built and trained using the Tensorflow [43] framework. The models are trained on a cloud server with an Intel Xeon CPU, 12 Gigabytes of RAM, and an Nvidia Tesla V100 GPU. The Adam optimizer was used to train the final models since it outperformed the stochastic gradient descent optimizer for training. We used binary cross-entropy as the loss function since it has outperformed other loss functions for this dataset [34]. Training data has to be

carefully selected for training both the 2D and 3D networks since the class imbalance of the training data might heavily skew the training process.

The baseline 2D networks must be trained first before we can build the proposed networks. Following the preprocessing steps mentioned in Section 2.2, normalized 2D CT scan slices and corresponding binarized segmentation masks are generated. Since the 2D networks will mainly be utilized to generate tumor features, only tumor-containing slices are taken for training the 2D segmentation network. The 2D feature extractor of a spatial feature fusion network is obtained from the baseline 2D segmentation network for the corresponding architecture. For example, a 2D-UNet is trained, and the contracting path of the network is used as a feature extractor for the SFF-3D-UNet architecture. Similarly, the 2D-MultiResUNet and the 2D-DenseUNet architectures are utilized as feature extractors for the SFF-3D-MultiResUNet and the SFF-Recurrent-3D-DenseUNet architecture. The 2D feature extraction network, which is part of the spatial feature fusion network now is frozen (set to not-trainable) while training to preserve 2D features and avoid issues with inconsistent gradients. To reduce model complexity and computational expense while preserving spatial resolution, stacks of eight consecutive 3D CT scans are taken instead of the whole 3D CT volume. We only take slices that contain at least one tumor to train the 3D networks to reduce a bias towards false negatives.

2.7. Segmentation mask generation

The 3D segmentation networks are trained to produce results on stacks of eight consecutive slices. To produce the segmentation mask on the whole CT image, overlapping stacks of CT slices are passed to the segmentation networks. The result is a set of overlapping segmentation masks. The overlapping masks are averaged out which serves as a step to remove noise in the output. The predicted segmentation masks are outputs of a sigmoid function which give a probability that a pixel contains a tumor or not between 0 and 1. A binary thresholding operation is usually performed to produce binary segmentation masks from the predictions. The thresholding operation also removes unwanted noise and reduces false positives. We also experiment with a novel two-step thresholding approach to improve detection and segmentation performance. This approach involves taking a higher threshold value to first filter out false-positive slices from the scan. Later, a lower threshold is applied to the remaining 2D slices to generate the final tumor volume. The rationale behind this two-step approach is discussed in detail in Section 3.3.2.

3. Results and discussion

3.1. Evaluation metrics

The dice coefficient has been used as the main metric for evaluating the segmentation performance between the generated masks and the ground truth for all test images. The dice coefficient quantifies the relative overlap of two sets between 0 and 1, where 0 represents no overlap and 1 represents perfect overlap. The dice coefficient is calculated using the following formula:

$$D = \frac{2 * |X \cap Y|}{|X| + |Y|} \tag{8}$$

Here X and Y represent the two sets corresponding to the binary segmentation masks of the ground truth and the prediction respectively. Similar to [34,35], the following conventions are used to compute the dice coefficient for true-negative and false-positive cases:

1. For True-Negative cases (Model successfully detects that no tumor is present), the dice coefficient is 1.
2. For False-Positive cases (Model mistakenly classifies tumor), the dice coefficient is 0.

Table 3
Dice coefficients (validation set) for different optimizers and learning rates for the 2D models.

Model	Optimizer	Learning rate	Dice coefficient (Validation)
2D-UNet	SGD	0.1	0.5327
2D-UNet	SGD	0.01	0.4799
2D-UNet	Adam	0.01	0.5327
2D-UNet	Adam	0.001	0.5950
2D-MultiResUNet	Adam	0.01	0.6066
2D-MultiResUNet	Adam	0.001	0.5436
2D-DenseUNet	Adam	0.01	0.6088
2D-DenseUNet	Adam	0.001	0.5926

Table 4
Dice coefficients (test set) for the 2D models.

Model	Optimizer	Learning rate	Threshold	Dice coefficient (Test set)
2D-UNet	Adam	0.001	0.5	0.5886
			0.7	0.6510
2D-MultiResUNet	Adam	0.01	0.5	0.6706
			0.7	0.7158
2D-DenseUNet	Adam	0.01	0.5	0.6098
			0.7	0.6911

We also report the 3D dice score for the 3D segmentation networks which is the 3D dice coefficient of the predicted tumor volumes with respect to the tumor volumes present in the ground truth. To evaluate the detection performance of our proposed networks, we also report the F1-score and Matthew's Correlation Coefficient (MCC). If the number of True-Positive, False-Positive, True-Negative, and False-Negative cases predicted by our networks are represented by TP, FP, TN, and FN respectively, then the F1-Score and MCC can be calculated as follows:

$$F1_{score} = \frac{2 * TP}{2 * TP + FP + FN} \tag{9}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{10}$$

3.2. Baseline 2D networks

3.2.1. Selection of training parameters

The baseline 2D networks need to be trained on 2D slices containing tumors from the dataset before we can utilize them for training the proposed 3D segmentation networks. We experiment with different optimizers and learning rates to fine-tune our baseline 2D networks — 2D-UNet, 2D-MultiResUNet, and our proposed 2D-DenseUNet architecture. Table 3 shows the different dice coefficients (2D) on the validation set resulting from different optimizers and learning rates. For the 2D-UNet architecture, we found that the Adam optimizer performed significantly better than Stochastic Gradient Descent (SGD). Therefore, Adam is chosen as the optimizer of choice for further experimentation. We experimented with different learning rates for the architectures and settled on a learning rate of 0.001 for 2D-UNet, and 0.01 for 2D-MultiResUNet and 2D-DenseUNet architectures.

3.2.2. Model performance

After training the baseline 2D networks with their optimal parameters, the trained models are evaluated against the test set to measure their performance in terms of 2D dice. Table 4 shows the dice coefficients for all the 2D models at different thresholds. The 2D-MultiResUNet model shows a better score in terms of the 2D dice coefficient among the different models. From the results, we can conclude that all the models can segment the 2D slices with a respectable degree of accuracy and are ready to be used in the next step for the formation of the proposed models.

Table 5
3D dice coefficients (validation set) for different learning rates for the 3D models.

Model	Optimizer	Learning rate	3D dice (Validation)
3D-UNet	Adam	0.001	0.5942
3D-UNet	Adam	0.0001	0.5955
SFF-3D-UNet	Adam	0.001	0.6550
SFF-3D-UNet	Adam	0.0001	0.6568
3D-MultiResUNet	Adam	0.001	0.5615
SFF-3D-MultiResUNet	Adam	0.001	0.6175
Recurrent-3D-DenseUNet	Adam	0.0001	0.5979
SFF-Recurrent-3D-DenseUNet	Adam	0.001	0.6458

3.3. 3D networks

3.3.1. Selection of training parameters

The Adam optimizer is used to train all the 3D models. We experimented with different learning rates for training the models. Table 5 shows the 3D dice coefficients on the validation set for different model configurations. The 3D models are larger in terms of the number of parameters, and when the learning rate is lowered compared to the learning rate of their corresponding 2D networks, the networks usually show better performance. From our experimentation, we can see that the 3D-UNet and SFF-3D-UNet models show marginally better performance at a learning rate of 0.0001. The 3D-MultiResUNet and SFF-3D-MultiResUNet models did not converge for a learning rate of 0.0001 and therefore the learning rate of 0.001 was an ideal choice. We chose a learning rate of 0.0001 for the Recurrent-3D-DenseUNet according to [34]. The SFF-Recurrent-3D-DenseUNet model showed better performance at a learning rate of 0.001.

3.3.2. Model performance

The 3D models such as 3D-UNet, 3D-MultiResUNet, and Recurrent-3D-DenseUNet, along with their counterparts with spatial feature fusion are trained using the Adam optimizer and the optimal learning rates found in the previous section. The best models are selected based on their performance on the validation set. The detection and segmentation performance of the best models are evaluated on the separate test set which contains data from 40 test subjects. We have experimented with different thresholding techniques while evaluating the models' performance.

The generated prediction masks of the segmentation networks are evaluated against the ground truth to benchmark their segmentation performance. The segmentation performance is evaluated mainly in terms of the mean dice coefficient (2D). To get a better idea of the volumetric segmentation performance of the segmentation networks, we also report the 3D dice coefficient which has not been reported before on this dataset. Before calculating the dice coefficients, the predicted segmentation masks are first binarized with binary thresholding techniques. The models are evaluated at different threshold values between 0.4 and 0.9 on the validation set. The threshold value of 0.7 provided the best 2D dice coefficient whereas the threshold value of 0.5 provided the best 3D dice coefficient on the validation set. The models are then evaluated for these threshold values on the test set and the results are reported in Table 6.

We can notice a common trend in the segmentation (Table 6) and detection (Table 7) performance for different thresholds. A lower threshold (0.5) generates a more accurate delineation of the 3D tumor volumes at the cost of more false positives and a lower overall 2D dice score. A higher threshold (0.7) gives fewer false positives and improves the overall dice, but the performance of 3D delineation falls significantly. To combat the issue and balance the trade-offs between false positives and segmentation accuracy, we have introduced a two-step thresholding approach discussed in Section 2.7. This method improves the overall dice by reducing the false positives and improving the delineation of the 3D volume at the same time. The segmentation performance of the different models with and without two-step thresholding is also reported in Table 6. We can see from the results that the two-step thresholding approach improves the 2D dice score by 1.30% on average.

It can also be concluded from the results that each of the proposed SFF models always performs better compared to their baseline 3D models in terms of both 2D and 3D dice coefficients. The 3D-UNet, the 3D-MultiResUNet, and the Recurrent-3D-DenseUNet - all show an improvement in both spatial and volumetric segmentation performance when spatial feature fusion is introduced. According to the mean 2D dice coefficient, the models respectively show a 1.61%, 2.25%, and 2.42% increase in performance when our proposed modifications are introduced. In terms of the 3D dice coefficient, the proposed models respectively show a performance improvement of 7.58%, 2.32%, and 4.28%. The best-performing model in terms of both 2D and 3D dice is our proposed *SFF-3D-MultiResUNet* architecture which achieves an impressive mean 2D dice score of 0.8669 and a mean 3D dice score of 0.5938. This is the best 2D dice score reported on the dataset compared to all previous works [33–35]. At the same time, this architecture is most efficient in terms of computational complexity (Table 9).

To detect the presence of a tumor, we use the predicted segmentation mask after thresholding. The detection is considered on a slice-by-slice basis, where if any pixel of the segmentation mask corresponding to the slice is greater than the threshold, we consider that the slice contains a tumor. Otherwise, the slice is classified as non-tumor. We experiment with different thresholds (0.5, 0.7) to evaluate the detection performance. Table 7 reports the various detection metrics like the number of true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), F1-score, and MCC (Matthew's Correlation Coefficient) for the different 3D models. From the detection performance of the different models, we can see that there is a tradeoff between the number of true positives and false positives with the change in threshold values. For a threshold value of 0.5, the models perform better in terms of true positive values. However, this increases the number of false positives, which reduces the overall detection and segmentation performance (see Table 6). The threshold value of 0.7 seems to give an optimum detection performance and results in the highest F1 and MCC scores. We can also see that the modified version of each of the 3D models performs better in terms of both F1 score and MCC. For threshold values of 0.7, the modified versions of the 3D-UNet, 3D-MultiResUNet, and Recurrent-3D-DenseUNet models with spatial feature fusion respectively report 4.71%, 7.73%, and 6.95% improvement in terms of the F1 score compared to their baseline. A similar trend is visible for the MCC values as well. Here, the best performing model in terms of detection is our proposed *SFF-3D-MultiResUNet* model with an F1 score of 0.7194. The detection performance of the model is significantly better compared to the results reported by other works on the same dataset [34,35]. It is to note that our proposed two-step thresholding approach gives detection scores similar to that with a threshold of 0.7, therefore it is not separately shown in the table.

It is evident from the quantitative analysis that our proposed models with spatial feature fusion perform significantly better in terms of detection and segmentation compared to their baseline models. It is worth mentioning that out of the 40 patients in the validation set, our proposed best model is able to detect the presence of tumors in 38 of them and provides a mean 2D dice coefficient greater than 0.80 for 33 out of 40 scans, which signifies a good overall segmentation score.

3.4. Computational overhead and training time

Naturally, 3D segmentation networks are computationally more expensive compared to 2D segmentation networks. However, the use of 2D upsampling convolutions in the decoder instead of 3D upsampling convolutions helps keep the computational costs tolerable. One might argue that the introduction of the 2D feature extraction network might add significant computational overhead while training. However, the

Table 6
Comparison of the dice coefficients (Test set) for different models at different thresholds.

Model	Threshold: 0.5		Threshold: 0.7		Two-step threshold	
	2D dice	3D dice	2D dice	3D dice	2D dice	3D dice
3D-UNet	0.7874	0.5460	0.8056	0.5102	0.8144	0.5440
SFF-3D-UNet	0.7914	0.5886	0.8178	0.5504	0.8275	0.5853
3D-MultiResUNet	0.8304	0.5844	0.8365	0.5368	0.8478	0.5803
SFF-3D-MultiResUNet	0.8437	0.5992	0.8555	0.5506	0.8669	0.5938
Recurrent-3D-DenseUNet	0.7777	0.5715	0.7984	0.5147	0.8080	0.5634
SFF-Recurrent-3D-DenseUNet	0.7916	0.5971	0.8143	0.5386	0.8276	0.5874

Table 7
Detection performance (Test set) of the different 3D models at different thresholds.

Model	Threshold	TP	FP	TN	FN	F1 score	MCC
3D-UNet	0.5	603	488	3416	245	0.6219	0.5264
	0.7	549	367	3267	299	0.6224	0.5307
SFF-3D-UNet	0.5	639	520	3114	209	0.6367	0.5460
	0.7	583	358	3276	265	0.6517	0.5664
3D-MultiResUNet	0.5	611	319	3315	237	0.6872	0.6111
	0.7	546	241	3393	302	0.6678	0.5945
SFF-3D- MultiResUNet	0.5	633	283	3351	215	0.7176	0.6493
	0.7	577	179	3455	271	0.7194	0.6601
Recurrent-3D-DenseUNet	0.5	637	539	3095	211	0.6294	0.5367
	0.7	566	403	3231	282	0.6230	0.5295
SFF-Recurrent-3D-DenseUNet	0.5	664	526	3108	184	0.6516	0.5661
	0.7	606	365	3269	242	0.6663	0.5893

Table 8
Comparison of different 3D models in terms of computational overhead.

Model	Number of parameters	Trainable parameters	Epochs to converge	Training time per epoch (min.)	Testing time (min.)	Dice score (2D)
3D-UNet	5.433×10^6	5.430×10^6	29	13:43	4:02	0.8144
SFF-3D-UNet	6.882×10^6	6.591×10^6	20	14:18	4:10	0.8275
3D-MultiResUNet	4.297×10^6	4.285×10^6	30	22:18	6:27	0.8478
SFF-3D-MultiResUNet	5.135×10^6	4.932×10^6	17	22:48	7:02	0.8669
Recurrent-3D-DenseUNet	19.220×10^6	19.216×10^6	29	29:04	8:01	0.8080
SFF-Recurrent-3D-DenseUNet	25.012×10^6	24.551×10^6	28	32:55	9:47	0.8276

baseline 2D networks that we selected for feature extraction are scaled-down and light-weight and we make efficient use of them by using only the first three encoder levels to generate features. Additionally, these layers are frozen (not trainable) at the time of training and do not affect the backward pass of the training step. Table 8 shows a brief comparison of the number of parameters between the various models. We can see that the SFF versions of the 3D-UNet and 3D-MultiResUNet architectures do not add significant computational complexity (number of parameters) compared to the baseline architectures. The same is reflected in the reported training times and testing times where the modified networks show only a marginal increase. The training and testing times are reported for a batch size of 2 with an Nvidia Tesla V100 GPU. It is to note that the Recurrent-3D-DenseUNet architecture, which we have used as-is from its original implementation [34] and its modified counterpart both are inherently computationally expensive due to the dense and recurrent connections present in the network.

The convergence of training with our proposed models is also in general faster compared to the baseline models. Where the baseline models usually take around 30 iterations to converge, our proposed SFF-3D-UNet model converges after 20 iterations of training. Convergence is even faster for the best performing *SFF-3D-MultiResUNet* model. Although an additional step in training is required to train the baseline 2D network for feature extraction, the overall training time is cut down by a significant margin for the proposed networks using spatial feature fusion. However, the SFF-Recurrent-3D-DenseUNet model takes longer to converge compared to the other models. This might be due to the unusually large number of parameters.

3.5. Comparison with other models

Table 9 reports a comparative analysis of the different models reported in the literature. Our proposed architectures with spatial feature fusion show excellent performance in terms of mean dice coefficient (2D). The Deeply Supervised MultiResUNet [35] with Test Time Augmentation (50 rotations) is the only model that performs better than our proposed SFF-3D-UNet and SFF-Recurrent-3D-DenseUNet models. Considering we have not implemented deep supervision or TTA with our models, the performance of these models in terms of dice coefficient is respectable. However, our best performing model, the *SFF-3D-MultiResUNet* shows significantly better performance both in terms of segmentation and detection compared to all the other models. This model reports the best 2D dice coefficient (0.8669) on the dataset which is a significant improvement over the previous best model, the Deeply Supervised MultiResUNet with TTA (dice coefficient: 0.8472). In terms of detection performance, our best model reports the best F1-Score (0.7194) and MCC (0.6601) as well. The Deeply Supervised MultiResUNet reports 123 False positives which is less than our model (179). However, our model performs much better in terms of true positives (577 compared to their 505) which is a more important measure when it comes to the diagnosis and detection of lung cancer. It is also worth mentioning that our proposed models can achieve good detection scores without sacrificing volumetric segmentation performance.

From Table 9, we can see that our best performing model, the SFF-3D-MultiResUNet model, is also the most efficient in terms of model complexity (number of parameters) among the models which have a competitive mean dice coefficient. This model uses significantly fewer parameters while providing the best performance in terms of both segmentation and detection. The Recurrent-3D-DenseUNet and its modified version are both computationally expensive. However, the introduction of spatial feature fusion does significantly boost the performance of the model. Our proposed SFF-3D-UNet model can achieve similar dice scores without any major architectural overhaul at a much lower computational complexity. The next best-performing model in

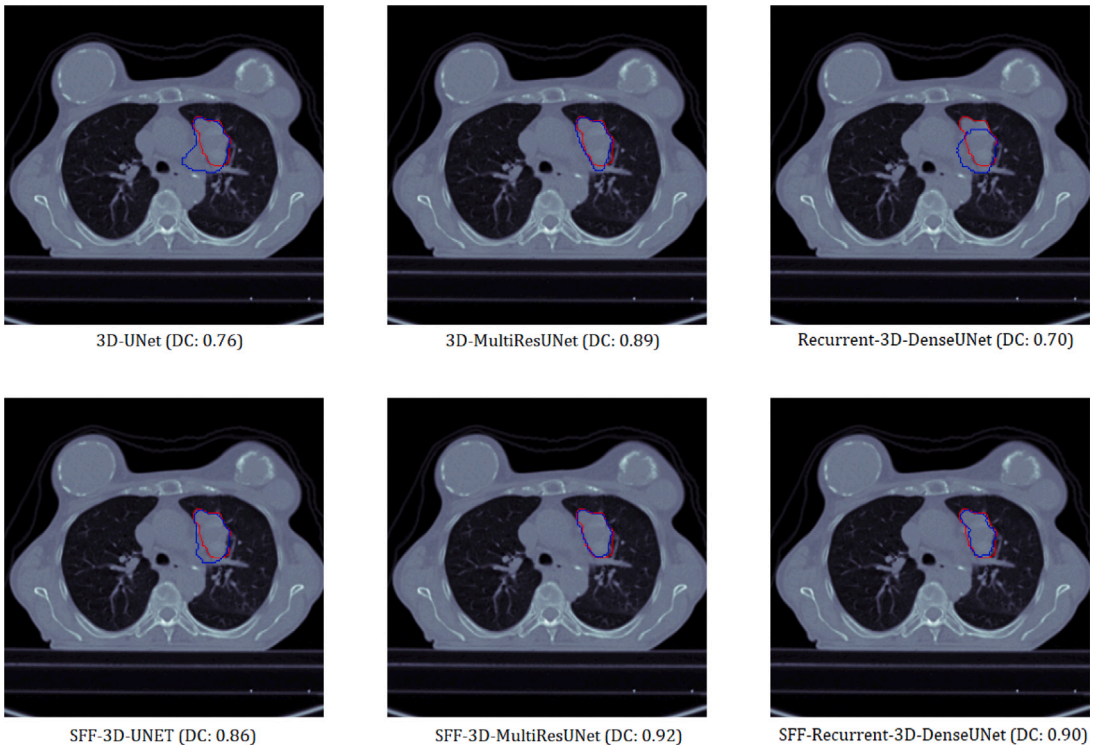

Fig. 6. Illustration of the ground truth (red) and predicted tumor boundaries (blue) for various models. Proposed models show significantly better segmentation performance (DC: Dice Coefficient).

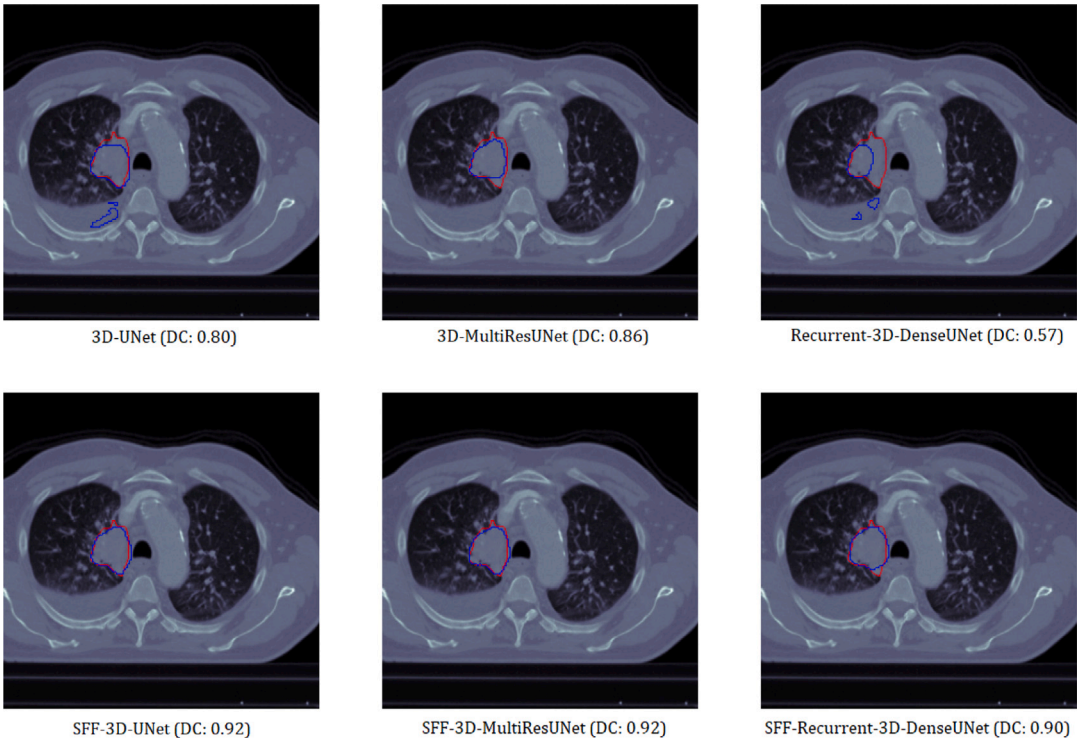

Fig. 7. Illustration of the ground-truth (red) and predicted tumor boundaries (blue) for various models. Our proposed models generate more accurate segmentation masks relative to the baseline models (DC: Dice Coefficient).

the literature, the Deeply Supervised MultiResUNet architecture with TTA has a computational complexity that is higher than both the SFF-3D-UNet and the SFF-3D-MultiResUNet architectures. It is also worth mentioning that this model performs 2D segmentation whereas our proposed models perform volumetric segmentation of eight consecutive CT slices at once. Also, the authors have utilized TTA with 50 rotations for evaluating the Deeply Supervised MultiResUNet model, which increases the inference time of their model 50-fold. Our proposed SFF-3D-MultiResUNet model achieves better performance without any such alterations at a significantly lower computational cost.

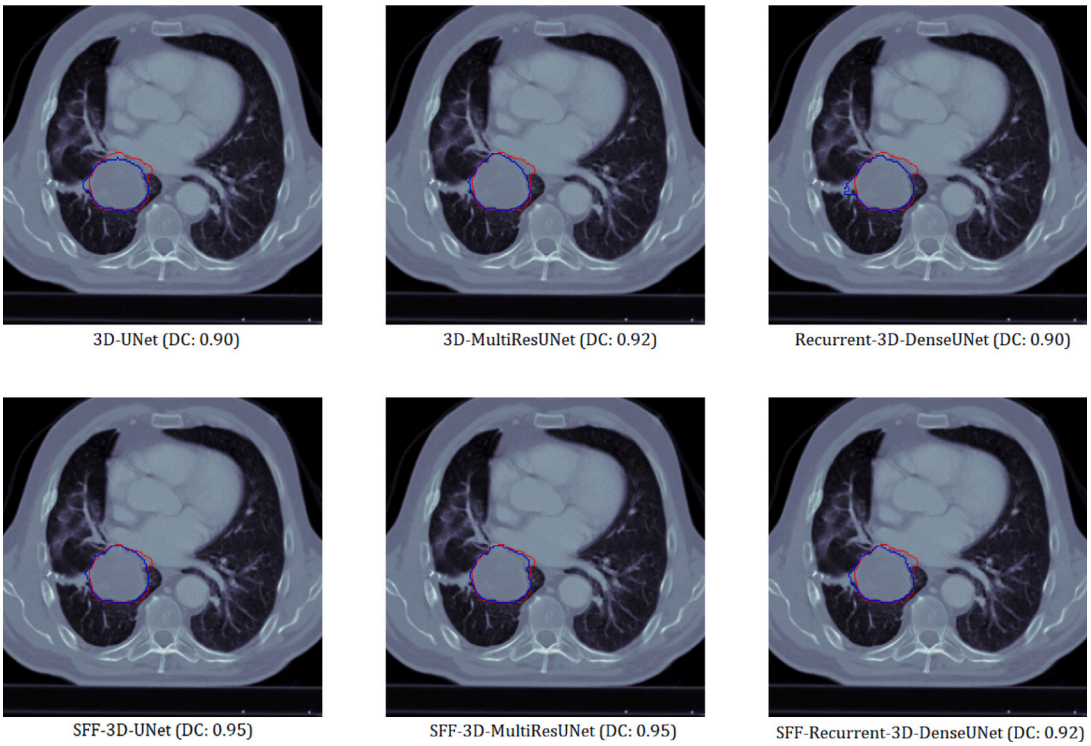

Fig. 8. Illustration of the ground-truth (red) and predicted tumor boundaries (blue) for various models. The proposed models generate finer and more accurate segmentation boundaries (DC: Dice Coefficient).

Table 9
Comparison of mean dice coefficient (2D) with other models.

Model	Mean dice coefficient (2D)	Number of parameters
2D-LungNet [33]	0.6267	1.30×10^5
3D-LungNet [33]	0.6577	4.03×10^5
3D-DenseNet [34]	0.6884	14×10^6
Recurrent-3D-DenseUNet [34]	0.7228	19.22×10^6
Deeply-Supervised-MultiResUNet [35]	0.8472	7.28×10^6
SFF-3D-UNet	0.8275	6.59×10^6
SFF-3D-MultiResUNet	0.8669	5.13×10^6
SFF-Recurrent-3D-DenseUNet	0.8276	25.01×10^6

3.6. Visual analysis

From our detailed quantitative analysis in Section 3.3.2, it is evident that our proposed networks are able to perform better in terms of both segmentation and detection. In all cases, our proposed models have shown better performance compared to their baseline architectures. This is also evident from a visual analysis of the generated segmentation boundaries on various 2D CT slices from the dataset. A few examples of the generated segmentation boundaries of the various 3D and SFF models are illustrated in Figs. 6–8. We can see from Fig. 6 that the segmentation performance is significantly better for the models modified with our proposed spatial feature fusion. Here, our proposed models show a relative improvement in terms of classifying the tumor and its neighboring regions. A similar trend can be seen in Fig. 7 where our proposed models with spatial feature fusion provide a more accurate segmentation mask. In cases where the tumor volume is clearly visible and easy to delineate, the baseline models do a good job. However, our proposed models are able to generate finer and more accurate segmentation boundaries in such cases as illustrated in Fig. 8. Our proposed best model *SFF-3D-MultiResUNet* shows impressive segmentation performance in all of the cases.

3.7. Limitations and future work

Although our proposed models perform well in most cases, there are some cases where our segmentation networks fail to perform proper segmentation of the tumor volume. Some of these cases for the 3D-MultiResUNet and the SFF-3D-MultiResUNet models are illustrated in Fig. 9. The models sometimes falsely identify lung nodules and tumor-like masses present in the lung as tumors (Fig. 9a, b). The models also occasionally fails to accurately detect tumor regions with erratic boundaries (Fig. 9b, c) or when the tumor region is very small (Fig. 9d). However, it is evident from a visual analysis of the results that the proposed model based on spatial feature fusion shows better performance compared to the baseline model even during these edge cases. Our networks might be able to better perform on such edge cases if exposed to a more diverse and larger training set. Publicly available biomedical segmentation data is a scarce resource, and our training sample is sufficiently small compared to the complexity of the segmentation task. Extensive collection and careful curation of more training samples might help our models learn from manifold examples and generalize better.

The main goal of this research is to address the issues with 3D convolutional neural networks for biomedical image segmentation and provide an approach that can make efficient use of both the spatial and volumetric features present in biomedical images. Thereby, this work has not focused on the modification of any architectural building blocks of UNet and other autoencoder segmentation networks. Instead, we have taken existing architectures like the UNet, MultiResUNet, and Recurrent-3D-DenseUNet and applied spatial feature fusion to establish the effectiveness of our proposed methodology. A limitation to this approach, as also evident in our quantitative analysis is that the performance of our proposed models is somewhat limited by the performance of their underlying architecture. For example, the performance of the MultiResUNet model is proven to be better than UNet in terms of segmentation [21]. This similar trend is translated over to the performance of our proposed models as well with the SFF-3D-MultiResUNet outperforming the SFF-3D-UNet. Therefore, it is logical

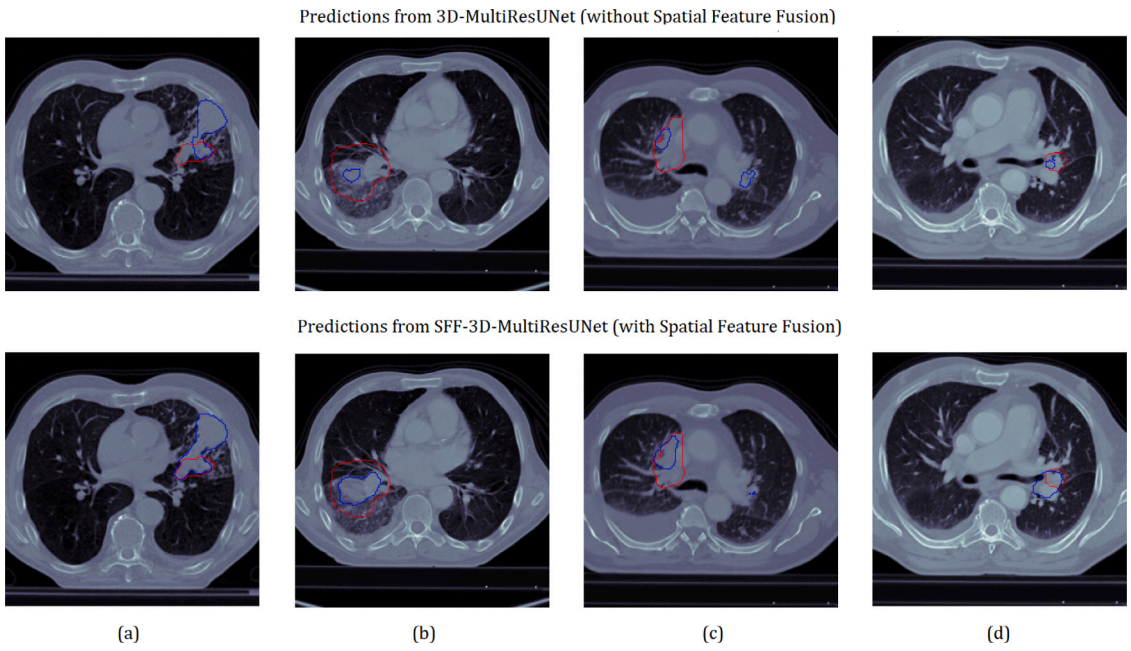

Fig. 9. Limitations of our segmentation networks. Some edge cases where our predictions are imperfect.

to presume that any architectural improvements of UNet or any of the segmentation networks of the autoencoder family can be translated into better performance with the help of our proposed methodology. Recent works on biomedical segmentation [35,44] have also utilized techniques like deep supervision which involve the evaluation of the loss at different levels of the network instead of only at the output. Further architectural modifications and advanced training strategies like deep supervision might open up room for improvement of our proposed segmentation models.

The UNet architecture and its many variants have successfully been utilized in many different volumetric biomedical image segmentation tasks involving the brain, lungs, heart, breast, liver, prostate, pancreas, etc. across various modalities like CT, MRI, Ultra Sound, etc. [45]. Although our experimentation and research mainly focus on lung tumor segmentation from CT images, the same approach might be extended to other biomedical image segmentation domains as well to improve volumetric segmentation performance. There is room for further experimentation on how our proposed architectures perform across biomedical images from different domains and modalities.

4. Conclusion

The accurate detection and segmentation of 3D biomedical images is a challenging task. In this paper, we have proposed a novel approach for the volumetric segmentation of lung tumors from 3D CT images. Our proposed method of spatial feature fusion enables existing autoencoder-based segmentation models to make use of both spatial (2D) features and volumetric information with minor architectural modification while significantly improving the performance of all the segmentation models. The proposed methodology has allowed us to achieve excellent results at the cost of minimum computational overhead, and the proposed SFF-3D-MultiResUNet network outperforms all previous approaches to achieve outstanding segmentation performance in terms of both 2D and 3D dice coefficient. Our proposed approach is able to provide more accurate detection and delineation of the tumor boundaries and we believe that it will be able to speed up the diagnostic process of lung cancer by assisting medical professionals and radiation oncologists. This might facilitate early detection of lung cancer which has the potential for saving lives. We plan to continue further research on this topic and explore different avenues to improve the overall pipeline and achieve better results. For example, we intend to explore architectural modifications of our proposed networks along with more advanced training strategies. We would also like to explore the performance of our proposed methodology on different medical imaging tasks.

CRediT authorship contribution statement

Suhail Najeeb: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Mohammed Imamul Hassan Bhuiyan:** Validation, Resources, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by The Department of Electrical and Electronic Engineering; Bangladesh University of Engineering and Technology, Dhaka, Bangladesh.

References

[1] H. Sung, J. Ferlay, R.L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, F. Bray, Global cancer statistics 2020: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries, CA: Cancer J. Clin. 71 (3) (2021) 209–249, http://dx.doi.org/10.3322/caac.21660, URL https://acsjournals.onlinelibrary.wiley.com/doi/abs/10.3322/caac.21660.

[2] C. Gridelli, A. Rossi, D.P. Carbone, J. Guarize, N. Karachaliou, T. Mok, F. Petrella, L. Spaggiari, R. Rosell, Non-small-cell lung cancer, Nat. Rev. Disease Primers 1 (1) (2015) 1–16, http://dx.doi.org/10.1038/nrdp.2015.9.

[3] D.E. Midthun, Early detection of lung cancer, F1000Research 5 (2016) http://dx.doi.org/10.12688/f1000research.7313.1.

[4] R. Siegle, D. Naishadham, A. Jemal, Cancer statistics, 2012, CA Cancer J. Clin. 62 (1) (2012) 10–29.

[5] National Lung Screening Trial Research Team, The national lung screening trial: overview and study design, Radiology 258 (1) (2011) 243–253, http://dx.doi.org/10.1148/radiol.10091808.

[6] A. Del Ciello, P. Franchi, A. Contegiacomo, G. Cicchetti, L. Bonomo, A.R. Larici, Missed lung cancer: when, where, and why? Diagn. Interv. Radiol. 23 (2) (2017) 118.

[7] B. Zhao, A.P. Reeves, D. Yankelevitz, C.I. Henschke, Three-dimensional multi-criterion automatic segmentation of pulmonary nodules of helical computed tomography images, Opt. Eng. 38 (8) (1999) 1340–1347, http://dx.doi.org/10.1117/1.602176.

[8] B. Zhao, L.H. Schwartz, C.S. Moskowitz, M.S. Ginsberg, N.A. Rizvi, M.G. Kris, Lung cancer: Computerized quantification of tumor response—Initial results, Radiology 241 (3) (2006) 892–898, http://dx.doi.org/10.1148/radiol.2413051887.

[9] D. Sharma, G. Jindal, Identifying lung cancer using image processing techniques, in: International Conference on Computational Techniques and Artificial Intelligence (ICCTAI), Vol. 17, 2011, pp. 872–880.

[10] A. Chaudhary, S.S. Singh, Lung cancer detection on CT images by using image processing, in: 2012 International Conference on Computing Sciences, 2012, pp. 142–146, http://dx.doi.org/10.1109/ICCS.2012.43.

[11] H.J. Aerts, E.R. Velazquez, R.T. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, et al., Decoding tumour phenotype by noninvasive imaging using a quantitative radiomics approach, Nature Commun. 5 (1) (2014) 1–9, http://dx.doi.org/10.1038/ncomms5006.

[12] P. Lambin, E. Rios-Velazquez, R. Leijenaar, S. Carvalho, R.G. van Stiphout, P. Granton, C.M. Zegers, R. Gillies, R. Boellard, A. Dekker, H.J. Aerts, Radiomics: Extracting more information from medical images using advanced feature analysis, Eur. J. Cancer 48 (4) (2012) 441–446, http://dx.doi.org/10.1016/j.ejca.2011.11.036, URL https://www.sciencedirect.com/science/article/pii/S0959804911009993.

[13] P. Sangamithraa, S. Govindaraju, Lung tumour detection and classification using EK-Mean clustering, in: 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), 2016, pp. 2201–2206, http://dx.doi.org/10.1109/WiSPNET.2016.7566533.

[14] S. Makaju, P. Prasad, A. Alsadoon, A. Singh, A. Elchouemi, Lung cancer detection using CT scan images, Procedia Comput. Sci. 125 (2018) 107–114, http://dx.doi.org/10.1016/j.procs.2017.12.016, The 6th International Conference on Smart Computing and Communications. URL https://www.sciencedirect.com/science/article/pii/S1877050917327801.

[15] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444, http://dx.doi.org/10.1038/nature14539.

[16] I.M. Nasser, S.S. Abu-Naser, Lung cancer detection using artificial neural network, Int. J. Eng. Inf. Syst. (IJEAIS) 3 (3) (2019) 17–23.

[17] S. Bhatia, Y. Sinha, L. Goel, Lung cancer detection: A deep learning approach, in: J.C. Bansal, K.N. Das, A. Nagar, K. Deep, A.K. Ojha (Eds.), Soft Computing for Problem Solving, Springer Singapore, Singapore, 2019, pp. 699–705, http://dx.doi.org/10.1007/978-981-13-1595-4_55.

[18] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[19] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[20] D. Jha, P.H. Smedsrud, M.A. Riegler, D. Johansen, T.D. Lange, P. Halvorsen, H. D. Johansen, ResUNet++: An advanced architecture for medical image segmentation, in: 2019 IEEE International Symposium on Multimedia (ISM), 2019, pp. 225–2255, http://dx.doi.org/10.1109/ISM46123.2019.00049.

[21] N. Ibtehaz, M.S. Rahman, MultiResUNet : Rethinking the U-Net architecture for multimodal biomedical image segmentation, Neural Netw. 121 (2020) 74–87, http://dx.doi.org/10.1016/j.neunet.2019.08.025, URL https://www.sciencedirect.com/science/article/pii/S0893608019302503.

[22] M. Ye, S. Xu, T. Cao, Q. Chen, Drinet: A dual-representation iterative learning network for point cloud segmentation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 7447–7456.

[23] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, S. Mougiakakou, Semantic segmentation of pathological lung tissue with dilated fully convolutional networks, IEEE J. Biomed. Health Inf. 23 (2) (2018) 714–722, http://dx.doi.org/10.1109/JBHI.2018.2818620.

[24] O. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, O. Ronneberger, 3D U-Net: Learning dense volumetric segmentation from sparse annotation, in: S. Ourselin, L. Joskowicz, M.R. Sabuncu, G. Unal, W. Wells (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016, Springer International Publishing, Cham, 2016, pp. 424–432, http://dx.doi.org/10.1007/978-3-319-46723-8_49.

[25] F. Milletari, N. Navab, S.-A. Ahmadi, V-Net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 Fourth International Conference on 3D Vision (3DV), 2016, pp. 565–571, http://dx.doi.org/10.1109/3DV.2016.79.

[26] H. Chen, Q. Dou, L. Yu, J. Qin, P.-A. Heng, VoxResNet: Deep voxelwise residual networks for brain segmentation from 3D MR images, NeuroImage 170 (2018) 446–455, http://dx.doi.org/10.1016/j.neuroimage.2017.04.041, Segmenting the Brain. URL https://www.sciencedirect.com/science/article/pii/S1053811917303348.

[27] S.G. Armato III, G. McLennan, L. Bidaut, M.F. McNitt-Gray, C.R. Meyer, A.P. Reeves, B. Zhao, D.R. Aberle, C.I. Henschke, E.A. Hoffman, E.A. Kazerooni, H. MacMahon, E.J.R. van Beek, D. Yankelevitz, A.M. Biancardi, P.H. Bland, M.S. Brown, R.M. Engelmann, G.E. Laderach, D. Max, R.C. Pais, D.P.-Y. Qing, R.Y. Roberts, A.R. Smith, A. Starkey, P. Batra, P. Caligiuri, A. Farooqi, G.W. Gladish, C.M. Jude, R.F. Munden, I. Petkovska, L.E. Quint, L.H. Schwartz, B. Sundaram, L.E. Dodd, C. Fenimore, D. Gur, N. Petrick, J. Freymann, J. Kirby, B. Hughes, A. Vande Casteele, S. Gupte, M. Sallam, M.D. Heath, M.H. Kuhn, E. Dharaiya, R. Burns, D.S. Fryd, M. Salganicoff, V. Anand, U. Shreter, S. Vastagh, B.Y. Croft, L.P. Clarke, The lung image database consortium (LIDC) and image database resource initiative (IDRI): A completed reference database of lung nodules on CT scans, Med. Phys. 38 (2) (2011) 915–931, http://dx.doi.org/10.1118/1.3528204, arXiv: https://aapm.onlinelibrary.wiley.com/doi/pdf/10.1118/1.3528204. URL https://aapm.onlinelibrary.wiley.com/doi/abs/10.1118/1.3528204.

[28] A.A.A. Setio, A. Traverso, T. de Bel, M.S. Berens, C. van den Bogaard, P. Cerello, H. Chen, Q. Dou, M.E. Fantacci, B. Geurts, R. van der Gugten, P.A. Heng, B. Jansen, M.M. de Kaste, V. Kotov, J.Y.-H. Lin, J.T. Manders, A. Sóñora-Mengana, J.C. García-Naranjo, E. Papavasileiou, M. Prokop, M. Saletta, C.M. Schaefer-Prokop, E.T. Scholten, L. Scholten, M.M. Snoeren, E.L. Torres, J. Vandemeulebroucke, N. Walasek, G.C. Zuidhof, B. van Ginneken, C. Jacobs, Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge, Med. Image Anal. 42 (2017) 1–13, http://dx.doi.org/10.1016/j.media.2017.06.015, URL https://www.sciencedirect.com/science/article/pii/S1361841517301020.

[29] Data science bowl 2017, 2017, URL https://www.kaggle.com/c/data-science-bowl-2017.

[30] A. Mohammadi, P. Afshar, A. Asif, K. Farahani, J. Kirby, A. Oikonomou, K.N. Plataniotis, Lung cancer radiomics: Highlights from the IEEE video and image processing cup 2018 student competition [SP competitions], IEEE Signal Process. Mag. 36 (1) (2019) 164–173, http://dx.doi.org/10.1109/MSP.2018.2877123.

[31] H.J.W.L. Aerts, L. Wee, E. Rios Velazquez, R.T.H. Leijenaar, C. Parmar, P. Grossmann, S. Carvalho, J. Bussink, R. Monshouwer, B. Haibe-Kains, D. Rietveld, F. Hoebers, M.M. Rietbergen, C.R. Leemans, A. Dekker, J. Quackenbush, R.J. Gillies, P. Lambin, Data from NSCLC-radiomics, 2019, http://dx.doi.org/10.7937/K9/TCIA.2015.PF0M9REI, URL https://wiki.cancerimagingarchive.net/x/FgL1.

[32] P. Afshar, A. Mohammadi, K.N. Plataniotis, K. Farahani, J. Kirby, A. Oikonomou, A. Asif, L. Wee, A. Dekker, X. Wu, et al., Lung-originated tumor segmentation from computed tomography scan (LOTUS) benchmark, 2022, arXiv preprint arXiv:2201.00458.

[33] S. Hossain, S. Najeeb, A. Shahriyar, Z.R. Abdullah, M. Ariful Haque, A pipeline for lung tumor detection and segmentation from CT scans using dilated convolutional neural networks, in: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 1348–1352, http://dx.doi.org/10.1109/ICASSP.2019.8683802.

[34] U. Kamal, A.M. Rafi, R. Hoque, J. Wu, M.K. Hasan, Lung Cancer Tumor Region segmentation using recurrent 3D-DenseUNet, in: J. Petersen, R. San José Estépar, A. Schmidt-Richberg, S. Gerard, B. Lassen-Schmidt, C. Jacobs, R. Beichel, K. Mori (Eds.), Thoracic Image Analysis, Springer International Publishing, Cham, 2020, pp. 36–47, http://dx.doi.org/10.1007/978-3-030-62469-9_4.

[35] F. Farheen, M.S. Shamil, N. Ibtehaz, M.S. Rahman, Revisiting segmentation of lung tumors from CT images, Comput. Biol. Med. 144 (2022) 105385, http://dx.doi.org/10.1016/j.compbiomed.2022.105385, URL https://www.sciencedirect.com/science/article/pii/S0010482522001779.

[36] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, P.-A. Heng, H-DenseUNet: hybrid densely connected UNet for liver and tumor segmentation from CT volumes, IEEE Trans. Med. Imaging 37 (12) (2018) 2663–2674, http://dx.doi.org/10.1109/TMI.2018.2845918.

[37] D. Mason, SU-E-T-33: pydicom: an open source DICOM library, Med. Phys. 38 (6Part10) (2011) 3493, http://dx.doi.org/10.1118/1.3611983.

[38] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48, http://dx.doi.org/10.1186/s40537-019-0197-0.

[39] S.-H. Wang, M.A. Khan, V. Govindaraj, S.L. Fernandes, Z. Zhu, Y.-D. Zhang, Deep rank-based average pooling network for Covid-19 recognition, Comput. Mater. Contin. 70 (2) (2022) 2797–2813, http://dx.doi.org/10.32604/cmc.2022.020140, URL http://www.techscience.com/cmc/v70n2/44681.

[40] S.-H. Wang, X. Zhang, Y.-D. Zhang, DSSAE: Deep stacked sparse autoencoder analytical model for COVID-19 diagnosis by fractional Fourier entropy, ACM Trans. Manage. Inf. Syst. 13 (1) (2021) http://dx.doi.org/10.1145/3451357.

[41] F. Isensee, P.F. Jaeger, S.A. Kohl, J. Petersen, K.H. Maier-Hein, NnU-Net: a self-configuring method for deep learning-based biomedical image segmentation, Nature Methods 18 (2) (2021) 203–211, http://dx.doi.org/10.1038/s41592-020-01008-z.

[42] M.M. Bejani, M. Ghatee, A systematic review on overfitting control in shallow and deep neural networks, Artif. Intell. Rev. 54 (8) (2021) 6391–6438, http://dx.doi.org/10.1007/s10462-021-09975-1.

[43] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, http://dx.doi.org/10.5281/zenodo.4724125, Software available from tensorflow.org. URL https://www.tensorflow.org/.

[44] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, UNet++: A nested U-net architecture for medical image segmentation, in: D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J.M.R. Tavares, A. Bradley, J.P. Papa, V. Belagiannis, J.C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, A. Madabhushi (Eds.), Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support, Springer International Publishing, Cham, 2018, pp. 3–11, http://dx.doi.org/10.1007/978-3-030-00889-5_1.

[45] I.R.I. Haque, J. Neubert, Deep learning approaches to biomedical image segmentation, Inform. Med. Unlocked 18 (2020) 100297, http://dx.doi.org/10.1016/j.imu.2020.100297, URL https://www.sciencedirect.com/science/article/pii/S235291481930214X.