Revisiting Segmentation of Lung Tumors from CT Images*

Farhanaz Farheen[1,2], Md. Salman Shamil[1,2], Nabil Ibtehaz[1], and M. Sohel Rahman †[1]

[1]Department of CSE, BUET, ECE Building, West Palasi, Dhaka-1230, Bangladesh
[2]Department of CSE, United International University, Dhaka, Bangladesh

Abstract

Lung cancer is a leading cause of death throughout the world. Because the prompt diagnosis of tumors allows oncologists to discern their nature, type, and mode of treatment, tumor detection and segmentation from CT scan images is a crucial field of study. This paper investigates lung tumor segmentation via a two-dimensional Discrete Wavelet Transform (DWT) on the LOTUS dataset (31,247 training, and 4,458 testing samples) and a Deeply Supervised MultiResUNet model. Coupling the DWT, which is used to achieve a more meticulous textural analysis while integrating information from neighboring CT slices, with the deep supervision of the model architecture results in an improved dice coefficient of 0.8472. A key characteristic of our approach is its avoidance of 3D kernels (despite being used for a 3D segmentation task), thereby making it quite lightweight.

Keywords: Lung tumor, CT scan images, deep learning, discrete wavelet transform, MultiResUNet.

1 Introduction

Cancer is a potentially fatal affliction and a leading cause of death worldwide. According to the Global Cancer Statistics 2018 [1], lung cancer is the most commonly diagnosed cancer in the world, with instances constituting 11.6% of total cancer cases, and also the prime cause of cancer-related fatalities, covering

*Farheen and Shamil contributed equally.

†Corresponding Author: msrahman@cse.buet.ac.bd

DOI: https://doi.org/10.1016/j.compbiomed.2022.105385

18.4% thereof. In addition to facing distressing chest pain, weight loss, and bouts of coughing, lung cancer patients are more likely to suffer from chronic obstructive pulmonary disease [2]. Mutations of lung cells - often prompted by smoking, which is the primary cause of lung cancer, or simply due to exposure to radon gas [2] - followed by their disorderly growth, results in the formation of tumors. In other words, lung tumors are mainly aberrant clusters of tissue resulting from atypical cell division in the lungs [3], [4]. Diagnosis of malignant tumors is often not possible in the early stages due to the lack of symptoms or because the symptoms are indistinguishable from those of a respiratory infection.

Thus, most lung cancer cases are diagnosed at an advanced stage [5], increasing the probability of the patient's death. Therefore, it is urgent that an effort be made to detect it during the early stages. In fact, if an early diagnosis can be made before the existing tumors have spread to neighboring tissues, the oncologist will have more options for the patient's treatment, including radiotherapy and chemotherapy. An important observation in this regard is that lung CT scan images have the capability of revealing tumors during the early stages. The challenge, however, is that undertaking such a task is tedious, time-consuming, and relatively error-prone, in general [6], [7].

Lung cancer prediction via CT scan images goes through several stages, including image pre-processing, segmentation, feature extraction, and classification. The pre-processing technique used in [8] involves removing unwanted artifacts with the help of Median and Wiener filters. Next, a K-Means clustering method is applied for segmentation, followed by EK-Means clustering. In [9], a Support Vector Machine (SVM) classifier is used for the early detection of lung cancer. After pre-processing a CT scan image of the lungs, the Region of Interest (ROI) is segmented, retained, and compressed using the DWT. SVMs are also used for lung cancer classification in [10]. Patel et al. used the Local Energy-Based Shape Histogram (LESH) feature extraction technique and a Sensitivity Analysis (SA) to detect lung cancer [11]. In [12], an enhanced artificial bee colony optimization technique is used for lung cancer detection and classification. Most common procedures for lung cancer detection have relied significantly on typical image processing methods, machine learning models based on handcrafted features, and soft computing techniques.

In recent times, deep learning techniques have revolutionized the field of medical image processing and analysis. In [13]. Haque et al. discuss the superiority of deep learning techniques in medical image segmentation. Traditional machine learning approaches usually cannot process natural data in their raw form and are incapable of dynamically adapting to new information [13]. However, deep learning techniques handle these items well, so they are widely used for such purposes. Moreover, with the advent of advanced Central Processing Units (CPUs) and Graphic Processing Units (GPUs), training and execution times have been reduced, making it easier to use algorithms based on deep learning. Therefore, advanced deep learning techniques [14], including Convolutional Neural Networks (CNNs) [15], have recently been used in the area of medical image segmentation. For example, in [16], lung cancer detection and classification is done using 3D CNNs. The Kaggle Data Science Bowl 2017 (KDSB17)

[17] challenge featured multitudinous applications of CNNs for the KDSB17 datasets, which were accompanied by the LUNA16 dataset in some cases [18]. On the other hand, the 2018 VIP-CUP Challenge presented the problem of segmentation and prediction of lung tumor regions, which was addressed with methodical procedures such as the Recurrent 3D-DenseUNet architecture with a Tversky loss function [19]. Furthermore, dilated hybrid 3D CNNs have also been applied in this context in addition to the LungNet and U-Net models [20]. All these approaches reported excellent results.

In [21], a study of automatic extraction of features leveraging deep learning techniques is conducted. A CAD system is proposed in [22] that uses deep features extracted from an autoencoder to classify lung nodules as malignant or benign. Moreover, three deep learning approaches, namely, CNNs, Deep Belief Networks (DBNs), and Stacked Denoising Autoencoders (SDAEs), are applied in [23] to demonstrate the use of deep learning in lung cancer diagnosis using the Lung Image Database Consortium (LIDC) database.

1.1 Our Contributions

The main contribution of this paper is a newly developed pipeline comprising a pre-processing phase with efficient feature engineering, an existing deep learning architecture with some useful enhancements, and some post-processing techniques for further improvement of the results. In particular, our technical contribution revolves around: (a) applying a two-dimensional Discrete Wavelet Transform (DWT) for improved textural inference, (b) analyzing neighboring CT slices for improved deductions, and (c) adopting deep supervision [24] via the MultiResUNet framework - a considerable improvement over the state-of-the-art U-Net models [25]. Additionally, the concerning bottleneck posed by a small dataset has been handled by data augmentation via rotations by differing angles. The combined impact of all these phases on accurate segmentation is the most significant contribution of our project, resulting in a dice coefficient of 0.8472, which exceeds all scores reported by previous works.

One characteristic feature of our pipeline is its avoidance of 3D kernels, despite the fact that a 3D segmentation task is being addressed. The dataset in our study (to be described shortly), while it is the most appropriate for our purpose, is rather small. Now, 3D CNNs, which are quite compute- and memory-intensive, may lead to overfitting with small datasets due to the larger numbers of parameters involved. As a result, it often becomes necessary to conceptualize the 3D space as a collection of 2D planes (c.f., Section 2.2.1 for further elaboration). This issue motivated us to transform the problem so that 2D kernels could be used with the designed feature map to achieve a much better performance while maintaining efficiency in terms of memory usage and run-time. This approach, along with other components of our pipeline, had a profound effect on the overall performance, resulting in a 17.21% improvement in the dice coefficient over that in [19], the highest result obtained on the same dataset hitherto in the literature.

2 Materials and Methods

2.1 Dataset

Our experiments have been conducted on the LOTUS Dataset (supplied by MAASTRO clinic), which is an amended adaptation of the NSCLC-Radiomics Dataset. This dataset consists of images and annotations from 300 patients in total. Among them, images from 260 patients are used for the training set and the rest were used for validation [26]. However, we have used the validation set as an independent test set since the original test set was unlabeled, rendering it useless for our evaluation. The ratio of the training and testing set is approximately 7:1 in terms of slices. The annotations of this dataset cover a broad region surrounding the tumor area, including both the right and the left lungs. Tumor volumes - Gross Tumor Volume (GTV), Planning Tumor Volume (PTV), and Clinical Target Volume (CTV) - have also been annotated [26]. The dataset is a compact collection of DICOM files containing several medical details on each patient. Our pre-processing methods extract the 512×512 CT scan slices from these DICOM objects that are sent into the pipeline after some further partitioning and refinements. Another essential detail is that the CT scans were produced by two different manufacturers, namely, CMS Imaging Inc. and SIEMENS. The relevant statistics have been reported in Table 1 along with a summary of the dataset [26].

Table 1: Number of tumor and non-tumor slices in the dataset along with dataset statistics with respect to different manufacturers

Dataset	Number of Subjects	Tumor Slices	Non-Tumor Slices	SIEMENS (#subjects)	CMS Imaging Inc. (#subjects)
Training Set	260	4296	26951	200	60
Test Set	40	848	3610	6	34

The dataset contains folders having the slices and annotations for each patient. We have dropped a few of these owing to missing or inconsistent annotations. Apart from those, all the other folders contain several DICOM files along with their corresponding contours. We have addressed some minor inconsistencies involving the identifiers of the annotation files by examining the coordinates of the original CT slices.

2.2 Proposed Approach

2.2.1 Overview of the pipeline

Our study makes use of two-dimensional CT slices for the segmentation task. We have applied 2D DWT on the original input images. While preparing the input instances, neighboring CT slices have also been incorporated with each image. U-Net and MultiResUNet models are then trained with the pre-processed data.

We have implemented both of these models with/without deep supervision. Test Time Augmentation (TTA) has also been applied in our experiments.

Challenges in medical image segmentation include variation in the shape, size, and texture of the ROI [13]. Our dataset contains a very small ROI in the two-dimensional CT slices. Usage of three-dimensional slices would have increased the proportion of non-tumor pixels considerably. This could make the models biased towards non-tumor regions, ultimately affecting their performance to a great extent. Moreover, the use of three-dimensional slices would have required a huge number of parameters, making the task more computationally expensive. On the contrary, we have used two-dimensional slices, which demand fewer parameters. To ensure that the minimum spatial information is included in the input instances, we have added the neighboring CT slices with each original slice. However, unlike a three-dimensional approach, it does not significantly inflate the proportion of blank spaces in the input, nor does it increase the parameter usage substantially.

Since the ROI is already quite small, capturing the texture of the tumors in the dataset is a rather arduous task, and as mentioned before, texture variation in ROI can be a challenge. Wavelet transforms can perform this task of texture analysis very well as is evident from several studies in the literature (e.g., [27], [28], [29], [30] etc.). It allows the model to identify minute details in the image. For this, we have chosen to use this in our pre-processing stage.

2.2.2 Pre-processing

Based on the manufacturer, the DICOM files have initially been partitioned into two groups. There is a stark difference in their span of intensity values in the Hounsfield Unit (HU), ranging from -1024 to 3071 for CMS Imaging Inc. and from 0 to 4095 for SIEMENS. Each partition has been normalized to establish relative uniformity in the dataset.

The masks generated have a lot of blank spaces surrounding the tumor regions. This makes the tumor reasonably insignificant compared to the empty area. Therefore, the images have been cropped to a smaller window to ensure that an adequately amplified tumor portion is supplied to the model. We have used the same coordinates to crop both the training and test sets. These coordinates have been obtained by analyzing the masks of the training set.

The pre-processing step also involves gathering more information from neighboring CT slices. A tumor is a three-dimensional structure, while the images that we are using are two-dimensional slices of them. Clearly, the adjacent slices of any given two-dimensional CT slice would carry essential information about whether a tumor exists in the given slice or not. To incorporate this context and account for dependencies between consecutive slices, the neighbors are added to each slice to prepare the input instance.

A 2D DWT is applied on each CT slice twice, generating the first and the second approximations of the image [31]. This operation simplifies images by leaving out some sharp horizontal, vertical, and diagonal details. By analyzing the original as well as the transformed slices, the model can perceive tumor

textures more minutely by identifying where these details are lost. In our study, the 512×512 images have been resized to 128×128.

The final feature set contains five slices concatenated one after another. These slices represent the original image, the previous and the following images, and the first and the second approximations of the original image (by 2D DWT). The training set has been enlarged, or more accurately, doubled by augmentation, i.e., flipping or rotating each image based on a probability. Figure 1 shows a flipped and a rotated sample for an image. After augmentation, we have obtained a total of 62,494 training samples. Figure 2 illustrates the pre-processing steps to prepare the input instances.

(a) Original (b) Flipped (c) Rotated

Figure 1: A CT slice with its corresponding flipped and rotated instances

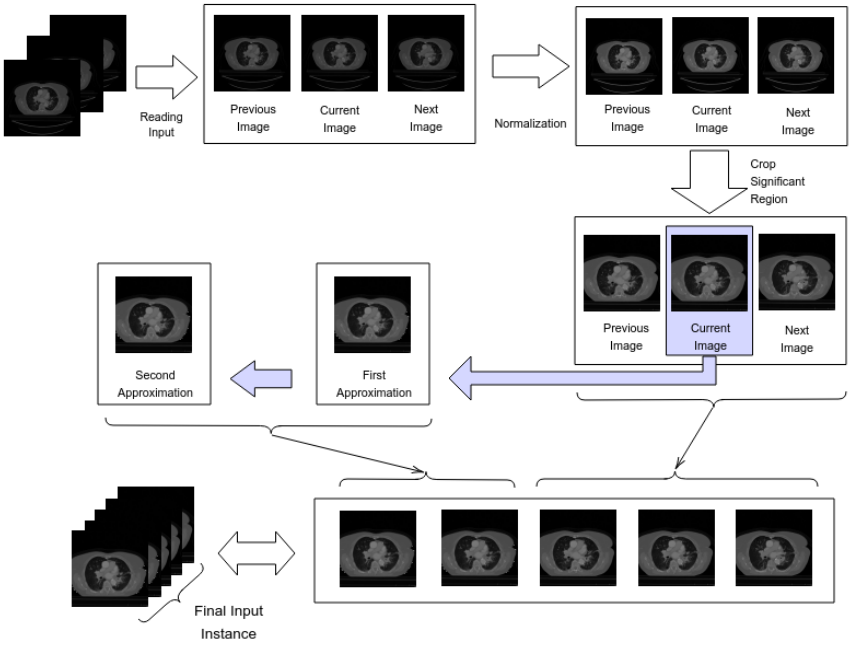


Figure 2: Input Instance Preparation

2.2.3 Description of the Models

We have experimented with two-dimensional U-Net [32], MultiResUNet [25], and Deeply Supervised [24] networks. Input instances of the shape (128, 128, 5) have been fed into all the architectures separately. Here, the (height, width) of the input images is (128, 128) after resizing the original images of size (512, 512). Also, each input instance contains five different channels, including the original image, wavelet transforms, and adjacent slices. Below we give a brief description of the models used in our study.

1. **U-Net**
   The U-Net framework consists of an encoding phase and a decoding phase. Each layer covers two convolutions and a max-pooling operation in the encoding phase, whereas the decoding phase replaces the max-pooling operation with an up-sampling. As the data progresses down the encoders, it gradually secures more context while leaving out details about the location. Consequently, spatial information is prepended to the contextual data of the decoders via skip connections.

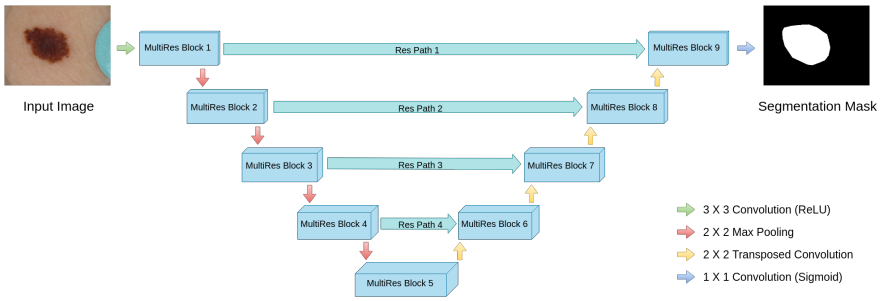
2. **MultiResUNet**
   The generic design of this network is quite similar to that of U-Net, except for slight differences in the nature of convolutions in each layer and the residual path from the encoders to the decoders (Figure 3). Each layer of U-Net is replaced by a MultiRes block that contains four convolutions (Figure 3b). The skip connections of U-Net are replaced by Res Paths (Figure 3c) as described in [25]. The motivation for adopting this network comes from the idea that an ideal architecture should be able to assess images having diversified scales in medical image segmentation procedures. For further details, readers are referred to [25].

3. **Deep Supervision**
   We have applied deep supervision on both U-Net and MultiResUNet models. Rather than only evaluating the output of the top layer's decoder, this model associates loss weights with the outputs of all five layers, in descending order from top to bottom. In order to achieve this, a simple 1 x 1 convolution is applied in all the layers after the up-sampling and convolution operations. Figure 4 presents Deeply Supervised U-Net and MultiResUNet models.

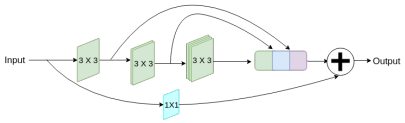
Notably, we have chosen U-Net [32] as our baseline model and applied deep supervision on U-Net and MultiResUNet in our experiments. Since the validation set of the original dataset is used for testing, a 5-fold cross-validation has been implemented.

2.2.4 Post-processing

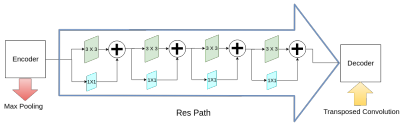
We have taken the predicted segmentation masks and constructed the final tumor masks through some useful post-processing steps. As the output masks'



(a) Network architecture



(b) MultiRes block



(c) Res Path

Figure 3: MultiResUNet architecture. Figure borrowed from [25]

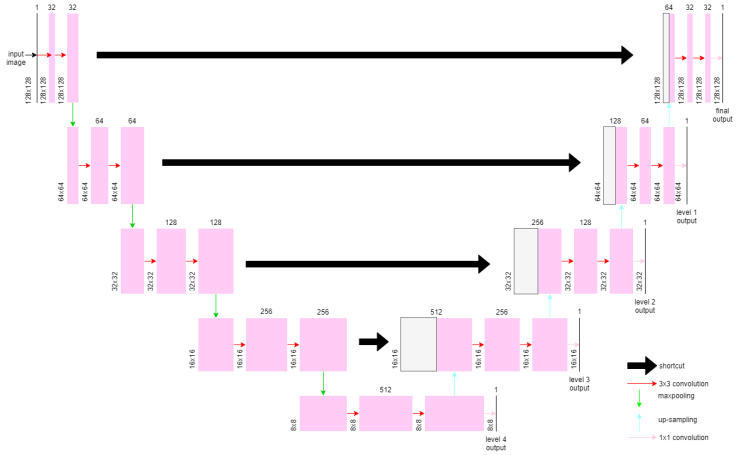
pixel values are originated from a sigmoid function, these values indicate the probabilities of a pixel to be tumorous. We needed to binarize the values to provide binary segmentation masks similar to the ground truth labels. For this purpose, we have experimented with suitable thresholds. The pixels which, after prediction, have values higher than that of the threshold, are considered tumorous. Moreover, we have performed TTA to achieve an ensemble effect, which involved creating multiple instances for a single test image through random rotations, as shown in Figure 1. All of the rotated images, along with the original ones, are fed into the trained models, and the output masks are taken to rotate back to the original orientation. These output masks are averaged and then binarized to produce the final output segmentation masks. The threshold values for binarization and the number of rotations for TTA are mentioned in Table 4 and Table 5.

2.3 Evaluation Metric

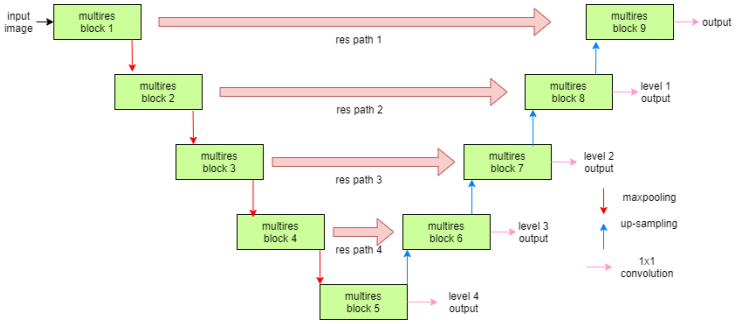
We have mainly used dice coefficient for performance evaluation and comparison. The value of dice coefficient is always in the range between 0 and 1 (inclusive) and is calculated using the following formula:

$$\text{dice coefficient} = \frac{2 * (|x \cap y|)}{|x| + |y|}$$

Here, x and y represent the tumor areas of the ground truth and the predic-



(a) Deeply Supervised U-Net



(b) Deeply Supervised MultiResUNet

Figure 4: Deeply Supervised Networks

tion, respectively. Following the literature [19], the dice coefficient is computed as follows: (i) For True-Negatives (i.e., there is no tumor, and the processing algorithm correctly detected that), the dice coefficient would be 1; and (ii) For False-Positives (i.e., there is no tumor, but the processing algorithm mistakenly segmented the tumor), the dice coefficient would be 0.

We have also reported the F1 score and Matthew's Correlation Coefficient (MCC), which are calculated using the following formulae.

$$\text{F1 score} = \frac{TP}{TP + \frac{FP+FN}{2}}$$

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

Here, TP, FP, FN, and TN represent True-Positive, False-Positive, False-Negative, and True-Negative, respectively.

2.4 Coding, Environment, and Availability

Our experiments have been conducted in a desktop computer having Intel Core i7-7700 processor (3.6 GHz, 8 MB cache) CPU, 16 GB RAM, and NVIDIA TITAN XP (12 GB, 1582 MHz) GPU. We have used the Python3 programming language [33]. For implementing the models, we have used Keras [34] with Tensorflow backend [35]. We have made the source code available at `https://github.com/farhanaz-farheen/LungTumorSegmentation`.

3 Results

3.1 Tuning of Model Parameters

We have assessed the results by varying different parameters in our model – optimization algorithm, learning rate, and decay of the network. This has led to various dice coefficients (training accuracy in this case), as reported in Table 2.

It can be observed from Table 2 that Adam has performed significantly better than Stochastic Gradient Descent (SGD) for both U-Net and MultiResUNet. Besides, for both the models, a learning rate of 0.01 has outperformed any other learning rate. This can be inferred from the table too where with an increase of the learning rate from 0.001 to 0.01, the dice coefficient increases. However, it falls once the learning rate is raised to 0.1.

Now that we have fixed the Adam optimizer (as the optimization algorithm) and an initial learning rate of 0.01, we can focus on deep supervision. Performance of deep supervision on U-Net and MultiResUNet models involves consideration of loss weights associated with the outputs from each level. Table 3 presents the dice coefficients achieved during the training for various loss weights in different layers.

Table 2: Dice Coefficients for different optimizers, learning rates, and decay for U-Net and MultiResUNet models

Model	**Optimization Algorithm**	**Learning Rate**	**Decay**	**Dice Coefficient (Training Accuracy)**
U-Net	Adam	0.1	0.1/150	0.7218
U-Net	Adam	0.01	0.01/150	0.7636
U-Net	Adam	0.001	0.001/150	0.7591
U-Net	SGD	0.1	0.1/150	0.5780
U-Net	SGD	0.01	0.01/150	0.4525
MultiResUNet	Adam	0.1	0.1/150	0.7437
MultiResUNet	Adam	0.01	0.01/150	0.7533
MultiResUNet	Adam	0.001	0.001/150	0.7435
MultiResUNet	SGD	0.5	0.5/150	0.6556
MultiResUNet	SGD	0.1	0.1/150	0.5916

Table 3: Dice Coefficients for deep supervision for various loss weights in different layers

Model	**Loss Weight (Final layer)**	**Loss Weight (Level 1)**	**Loss Weight (Level 2)**	**Loss Weight (Level 3)**	**Loss Weight (Level 4)**	**Dice Coefficient (Training Accuracy)**
Deeply Supervised U-Net	1.00	0.8	0.6	0.4	0.2	0.7574
Deeply Supervised U-Net	1.00	0.7	0.5	0.3	0.1	0.7582
Deeply Supervised U-Net	1.00	0.6	0.4	0.2	0.0	0.7578
Deeply Supervised U-Net	1.00	0.5	0.3	0.1	0.0	0.7578
Deeply Supervised MultiResUNet	1.00	0.8	0.6	0.4	0.2	0.7482
Deeply Supervised MultiResUNet	1.00	0.7	0.5	0.3	0.1	0.7421
Deeply Supervised MultiResUNet	1.00	0.6	0.4	0.2	0.0	0.7430
Deeply Supervised MultiResUNet	1.00	0.5	0.3	0.1	0.0	0.7402

3.2 Models' Performance

We have used our test set which contains data from 40 subjects. Various threshold values and numbers of rotations have been experimented with while performing TTA.

3.2.1 Tumor Detection

Since the probability of a pixel to be part of a tumor is between 0 and 1, we have binarized these predicted probability values based on certain thresholds. For example, if the probability of a pixel for being a tumor is x and the threshold is t then for all $x > t$, our model predicts it to be a tumor, whereas for all $x \leq t$, the model does not recognize it as a tumor pixel. Table 4 reports the number of true positives, false positives, true negatives, false negatives, F1 score, and MCC for all the model settings.

Table 4: All quantitative results related to tumor detection for different numbers of rotations and thresholds using the test set

Model	**Number of Rotations**	**Thres -hold**	**True Positive**	**False Positive**	**True Negative**	**False Negative**	**F1 Score**	**Matthew's correlation coefficient (MCC)**
U-Net	20	0.4	104	130	3502	744	0.1922	0.1529
U-Net	50	0.5	53	58	3574	795	0.1105	0.1173
MultiResUNet	20	0.4	576	215	3417	272	0.7029	0.6370
MultiResUNet	50	0.5	447	115	3517	401	0.6340	0.5860
Deeply Supervised U-Net	20	0.4	632	520	3112	216	0.6320	0.5397
Deeply Supervised U-Net	50	0.5	501	210	3422	347	0.6427	0.5714
Deeply Supervised MultiResUNet	20	0.4	600	268	3364	248	0.6993	0.6281
Deeply Supervised MultiResUNet	50	0.5	505	123	3509	343	0.6843	0.6337

3.2.2 Tumor Segmentation

Table 5 reports the dice coefficients for the test set with various numbers of rotations applied during TTA. Moreover, in addition to performing standard binarization using a threshold of 0.5, we have experimented with different threshold values. Evidently, deep supervision has been able to improve the performance for both U-Net and MultiResUNet, and the latter has performed better. More specifically, the Deeply Supervised MultiResUNet model (with 50 rotations and 0.5 threshold) is the winner with a dice coefficient of 0.8472.

3.3 Comparison with other models

A comparative analysis involving other state-of-the-art models shows that our model outperforms all of them by a significant margin in terms of dice coefficient. This is presented in Table 6 which shows the number of parameters required

Table 5: Dice coefficients for different numbers of rotation and thresholds using the test set

Model	**Number of Rotations**	**Threshold**	**Dice Coefficient (Test Accuracy)**
U-Net	20	0.4	0.7882
U-Net	50	0.4	0.7890
U-Net	20	0.5	0.7984
U-Net	50	0.5	0.8003
MultiResUNet	20	0.4	0.8327
MultiResUNet	50	0.4	0.8363
MultiResUNet	20	0.5	0.8373
MultiResUNet	50	0.5	0.8382
Deeply Supervised U-Net	20	0.4	0.7645
Deeply Supervised U-Net	50	0.4	0.7858
Deeply Supervised U-Net	20	0.5	0.8158
Deeply Supervised U-Net	50	0.5	0.8224
Deeply Supervised MultiResUNet	20	0.4	0.8294
Deeply Supervised MultiResUNet	50	0.4	0.8324
Deeply Supervised MultiResUNet	20	0.5	0.8434
Deeply Supervised MultiResUNet	50	0.5	0.8472

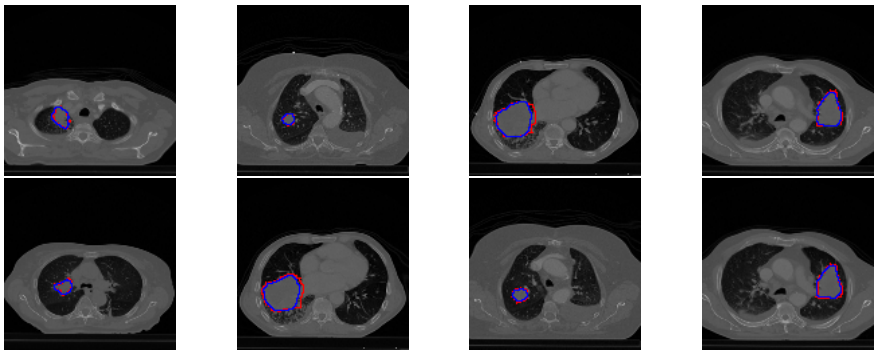
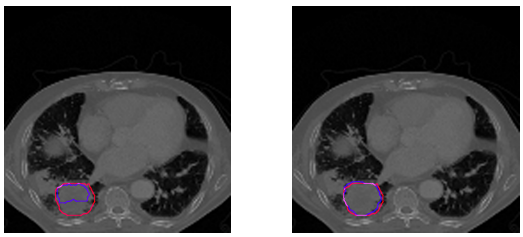


Figure 5: Strength of our model. The ground truth (in red) and prediction (in blue) for Deeply Supervised MultiResUNet model



(a) Prediction with MultiResUNet (Image 1)

(b) Prediction with Deeply Supervised MultiResUNet (Image 1)

(c) Prediction with MultiResUNet (Image 2)

(d) Prediction with Deeply Supervised MultiResUNet (Image 2)

Figure 6: Deeply Supervised MultiResUNet outperforms MultiResUNet significantly in terms of capturing the range of the tumor. The ground truth is shown in red, whereas the prediction is given in blue

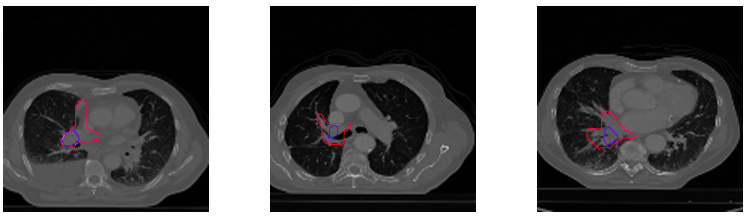


Figure 7: Limitation of our model. Tumors with tremendously erratic boundaries are predicted imperfectly. The ground truth is shown in red, whereas the prediction is given in blue

and the mean dice coefficient obtained by our model and several others. It can be seen that Recurrent 3D-DenseUNet [19], which gave a dice coefficient closest to the result obtained by our model, requires more parameters than Deeply Supervised MultiResUNet. Moreover, although 2D-LungNet [36] and 3D-LungNet [20] require fewer parameters than our model, our dice coefficient exceeds them by a large margin, and this improvement is worth the cost of this additional complexity.

At this point, a brief discussion on the (unusual) dice coefficient calculation of Hossain et al. [20] is in order, which differs from the calculation followed in [19] (and this work). The test set in [20] contains 4478 slices from 40 subjects. Their front-end binary classifier produces 1158 false negatives out of those, but this huge number of misclassifications does not affect the dice coefficient, as only the slices with tumor pixels in the ground truth are considered during their dice coefficient calculation. Although they have mentioned that the number of false positives is reduced by approximately 50% through segmentation and post-processing, the paper makes no attempt to address what happens to the true positives reported by the binary classifier after passing them through the segmentation model. Therefore, the possibility of a reduction in the number of true positives is conveniently ignored.

Let us now analyze the quantitative results reported in [19]. The highest dice coefficient achieved by them is 0.7228. For this result, the false positives and false negatives are 321 and 331 in number, respectively [19]. In our case, the best dice coefficient (of 0.8472) is achieved by the Deeply Supervised MultiResUNet, which gives 123 false positives (much less than [19]) and 343 false negatives (close to [19]). Notably, we have already achieved much fewer false negatives for Deeply Supervised MultiResUNet with 20 rotations and 0.4 threshold (248 in number) at the cost of a slightly worse dice score (i.e., 0.8294) and false positives (268 in number) - both far better than that of [19]. However, we still plan on exploring ways to reduce both false positives and negatives further while maintaining a high dice coefficient.

3.4 Ablation Study

3.4.1 Impact of the wavelet transforms

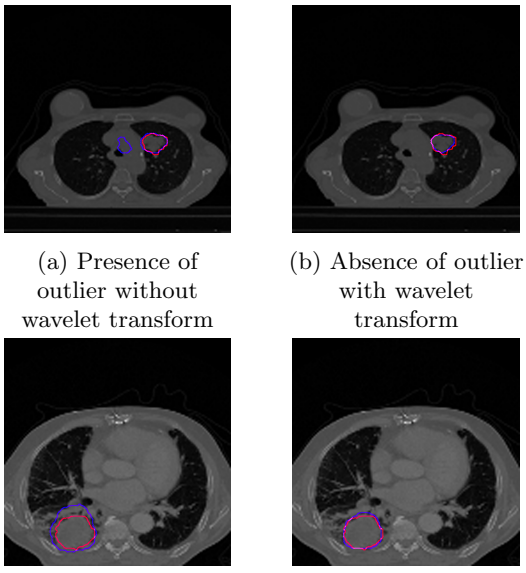
We have conducted an ablation study to analyze the performance of the wavelet transforms. To do this, we have pre-processed the images without wavelet transforms and have only kept the neighboring slices along with the original image. This gives us a 3-channel input instead of a 5-channel one (original slice and two neighboring slices), wherewith we then train the Deeply Supervised MultiResUNet. This results in a degraded performance of the (ablated) model on the test set, registering a dice coefficient of 0.7649 as opposed to the original value of 0.8294 (Table 7).

Table 6: Dice coefficients obtained using different models in lung tumor segmentation task. Dice scores for the other models have been taken from the respective papers ([20] and [19]). Short description and the number of parameters for each model are also shown

Model	**Mean Dice Coefficient**	**Description**	**Number of Parameters**	**Shape of input (resized) to the model** (height,width,channels)
2D-LungNet [36]	0.6267	Original LungNet architecture that consists of solely convolutional layers with dilated kernels. It uses 2D kernels and works on 2D input slices independently.	$1 \cdot 30 \times 10^5$	(224,224,1)
3D-LungNet [20]	0.6577	Uses the LungNet as the feature extractor on 2D input slices and concatenates the feature maps for 9 consecutive slices, which is followed by 3D convolutional layers. The method is enhanced by a front-end binary classifier and some post-processing techniques.	$4 \cdot 03 \times 10^5$	(224,224,9)
3D-DenseUNet [19], [37]	0.6884	Based on original U-Net's 3D version with additional interconnections between layers.	14×10^6	(256,256,8)
Recurrent 3D-DenseUNet [19]	0.7228	A combination of DenseNet, CNN and RNN. Consists of a 3D encoder block for feature extraction from 8 consecutive slices, a recurrent block of multiple ConvLSTM layers and a 3D decoder block to generate segmentation masks, followed by selective thresholding and dilation for post-processing.	19×10^6	(256,256,8)
Deeply Supervised MultiResUNet	**0.8472**	Our proposed method in this study.	$7 \cdot 28 \times 10^6$	(128,128,5)

Table 7: Experimental setup with results for the ablation study to evaluate the performance of wavelet transforms

Wavelet Transform Applied?	**Model**	**Loss Weights Top to Bottom Layer**	**Number of Rotations**	**Threshold**	**Dice Coefficient (Test Set)**
Yes	Deeply Supervised MultiResUNet	1.0, 0.8, 0.6, 0.4, 0.2	20	0.4	0.8294
No	Deeply Supervised MultiResUNet	1.0, 0.8, 0.6, 0.4, 0.2	20	0.4	0.7649



(a) Presence of outlier without wavelet transform

(b) Absence of outlier with wavelet transform

(c) Enlarged prediction without wavelet transform

(d) Better edge prediction with wavelet transform

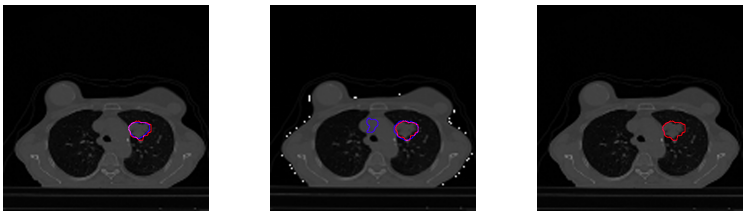
Figure 8: Application of wavelet transform reduces the frequency of outliers and avoids an inflated predicted tumor size. The ground truth is shown in red, whereas the prediction is given in blue

3.4.2 Impact of using the neighboring slices

A second ablation study has been performed to evaluate the impact of using the neighboring slices as part of the input instances. When we remove the neighboring slices from the input instance, we get a 3-channel input instead of a 5-channel one. In the same way as the previous case, we then train the Deeply Supervised MultiResUNet in this setting. This also results in a degraded performance of the (ablated) model on the test set, registering a dice coefficient of 0.6257 as opposed to the original value of 0.8294.

Table 8: Experimental setup for the ablation study to evaluate the performance of having neighboring slices

Neighboring Slices Added?	**Model**	**Loss Weights Top to Bottom Layer**	**Number of Rotations**	**Threshold**	**Dice Coefficient (Test Set)**
Yes	Deeply Supervised MultiResUNet	1.0, 0.8, 0.6, 0.4, 0.2	20	0.4	0.8294
No	Deeply Supervised MultiResUNet	1.0, 0.8, 0.6, 0.4, 0.2	20	0.4	0.6257



(a) Accurate prediction with 2 neighboring slices

(b) Outliers with no neighboring slices

(c) Inaccurate prediction with 4 neighboring slices

Figure 9: Adding neighboring slices causes tumor prediction to be more accurate. However, increasing the number of neighboring slices can cause performance to fall, and in this case, the model fails to detect any tumor at all. These predictions have been obtained by using the test samples

3.5 Increasing the number of neighboring slices

Since including neighboring slices improves performance significantly (as is evident from our ablation study), one obvious idea would be to include even more neighboring slices. To investigate this, we create a 7-channel input instance by placing two neighboring slices before and two neighboring slices after the original slice. Unfortunately, this does not improve the results; instead, this results in a significantly degraded dice coefficient of 0.6333 (Table 9).

Table 9: Experimental setup and results to evaluate the performance of having more neighboring slices

Number of Neighboring Slices	Model	Loss Weights Top to Bottom Layer	Number of Rotations	Threshold	Dice Coefficient (Test Set)
4 (2 before and 2 after)	Deeply Supervised MultiResUNet	1.0, 0.8, 0.6, 0.4, 0.2	20	0.4	0.6333
2 (1 before and 1 after)	Deeply Supervised MultiResUNet	1.0, 0.8, 0.6, 0.4, 0.2	20	0.4	0.8294

4 Discussion

MultiResUNet outperforms U-Net in the segmentation of lung tumors from CT slices, and Deeply Supervised MultiResUNet performs even better. In particular, due to deep supervision, the best MultiResUNet model registers an improvement from 0.8382 to 0.8472 in the dice coefficient. Since deep supervision considers the outputs from the lower levels of the decoder, the reconstruction phase of the model begins at a very deep level, positively affecting the final dice coefficient calculated on the predictions. Besides, we believe that the combination of wavelet transforms and the use of neighboring CT slices during input instance preparation has allowed our model to analyze tumor textures more accurately.

Table 4 demonstrates various quantitative results for tumor detection. A crucial observation is that, for every model, the application of deep supervision reduces the number of false negatives. Reducing false negatives is crucial, as failing to identify a tumor has serious detrimental effects for the patient. Tables 4 and 5 illustrate another pattern. The models with higher F1 scores also yield better dice coefficients, although the margin may not be very high.

Table 5 shows an increasing pattern in dice coefficients as we increase the thresholds and number of rotations. For each segmentation model, as the threshold is held constant, when the number of rotations increases, the dice coefficient increases as well. This phenomenon can be explained as follows. When the number of rotations is low, it is difficult to filter out the false-positive pixels, whereas, for a greater number of rotations, such pixels have very low probabilities of being parts of tumors. In other words, when rotations increase, the ability to confidently claim that a particular pixel is part of a tumor increases. Setting a higher threshold has a similar effect on the predictions because a pixel is only considered to be part of a tumor if the predicted probability is quite high. Otherwise, it is discarded, thus effectively filtering out the false-positive pixels.

Another observation arising from Table 5 is that, with the increase in threshold, the deep supervision performance improves for both U-Net and MultiResUNet. As the number of rotations is held constant, an increase in threshold results in an improved dice coefficient for deep supervision. We observe that the

Deeply Supervised versions of the model almost consistently have probability scores higher than 0.5 in the tumor regions. So, when we consider threshold values less than 0.5 (e.g., 0.4), very few additional actual tumor pixels are considered compared to the background pixels with scores in the range of 0.4-0.5. As a result, there is a slight dip in performance.

Table 5 also presents slightly better dice scores for models without deep supervision and a threshold value of 0.4, as opposed to the case when the threshold value is 0.5. The rationale behind this change in outcome comes from the continuation of the observation in the previous paragraph. For models without deep supervision, there is a larger ratio of tumor pixels in the 0.4-0.5 range; thus, their performances beat those of the Deeply Supervised models. Our takeaway point is that, apparently, for Deeply Supervised models, the separation between background and foreground is a bit better.

A qualitative analysis of our results (c.f., Section 3.2.2) reveals the strengths and weaknesses of our proposed model. In most cases, regardless of the size of the ground truth, our model predicts the tumor shapes very well, which is evident from Figure 5, where the ground truth and prediction (by Deeply Supervised MultiResUNet) are shown in red and blue, respectively. Although the tumors are diverse in size and appear in arbitrary locations within the lung, the red and blue margins appear to coincide almost perfectly.

Figure 6 shows a comparison between predictions by the MultiResUNet model and Deeply Supervised MultiResUNet model. Note that the latter can delineate the tumor edges more accurately than the former. The ground truth (shown in red) and the prediction (shown in blue) are more consistently aligned with each other in the Deeply Supervised MultiResUNet's prediction.

However, in a few rogue cases where the ground truth shapes are exceedingly uncoordinated, perhaps with fissures in the middle or with erratic outlines, it can be seen that the predicted tumor regions are prone to imperfections. Although, for these cases, our model can predict the existence of tumors, the shapes are often disfigured, as demonstrated in Figure 7. We plan to address this issue in our future work.

The ablation study reported in Section 3.4 also results in some interesting insights, which are illustrated in Figures 8 and 9. We observe two things from Figure 8. First, in the absence of the wavelet transform, the predicted tumors seem to have frequent outliers. Second, in their absence, the predicted tumors seem to be enlarged in most cases and the edges of the tumors are not detected properly. The intuition concerning the former observation is that without the wavelet transforms, the textural information is mostly lost, and the model perceives similarly shaped structures to be tumors as well. The second observation is also related to the fact that wavelet transforms can capture texture better. In its absence, the model cannot predict or justifiably limit the extent of the area that a tumor boundary should encompass, resulting in inflated tumor areas and inaccurate tumor sizes.

Figure 9, on the other hand, reveals an insight into the impact of the neighboring slices. In particular, it reveals that removal of neighboring slices results in incorrect tumor prediction and outliers (Figure 9(b)). Although 2D slices

are being considered, tumors are 3D structures. As such, allowing the model to analyze neighboring slices can help it predict a tumor with more confidence. Table 8 establishes this insight quantitatively, and Figure 9 reinforces it qualitatively. In addition, the model fails when too many neighboring slices are included in the input instances ((Figure 9(c)). These inaccurate predictions can be attributed to the overfitting of the model on the training set.

The mechanism proposed in this paper is unique, and it results in a procedure that returns an excellent dice coefficient value of 0.8472. The technique of combining neighboring slices for more detail and applying DWT for textural analysis has not been implemented in this manner previously. The first and second approximations of the original slice, obtained from a 2D DWT, allows the model to peruse lost details, resulting in a more nuanced interpretation of the input. We have also assigned priority to the outputs of the lower layers of the model by employing deep supervision. Finally, the application of TTA has assisted the network in execution when supplied with new instances.

5 Conclusion

The battle against lung cancer is ongoing. To this end, a pivotal field of research has emerged and gained traction with the help of deep learning in recent years. The proposed methods in our paper have outperformed existing approaches in detecting lung tumors with CT scan images. The strategy that we have undertaken, which included a 2D DWT during the pre-processing step and experimenting with multiple models, led to splendid results in terms of the dice coefficient, especially when the Deeply Supervised MultiResUNet model, which ensured a more accurate delineation of faint boundaries, was employed.

We believe that our approach can make the tasks of oncologists much more manageable and help detect tumors at an early stage. We have planned further research regarding this particular topic and with regard to refining the pipeline to achieve even better results. For example, we plan to explore other preprocessing techniques apart from wavelet transforms, thereby redesigning the pipeline. Moreover, we plan to experiment with additional techniques with the potential to improve the performance of the models we used. One such interesting avenue to explore is employing an attention mechanism with the MultiResUNet model. We also plan to consider adding a front-end segmentation mechanism capable of isolating the lung within the CT image before feeding the dataset to the deep supervision model.

References

[1] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, "Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, vol. 68, no. 6, pp. 394–424, 2018.

[2] D. S. Ettinger, W. Akerley, G. Bepler, M. G. Blum, A. Chang, R. T. Cheney, L. R. Chirieac, T. A. D'Amico, T. L. Demmy, A. K. P. Ganti *et al.*, "Non–small cell lung cancer," *Journal of the national comprehensive cancer network*, vol. 8, no. 7, pp. 740–801, 2010.

[3] "Lung tumors," https://my.clevelandclinic.org/health/diseases/15023-benign-lung-tumors.

[4] "Lung cancer basics," https://www.lung.org/lung-health-diseases/lung-disease-lookup/lung-cancer/learn-about-lung-cancer/what-is-lung-cancer/lung-cancer-basics.

[5] D. E. Midthun, "Early detection of lung cancer," *F1000Research*, vol. 5, 2016.

[6] S. Makaju, P. Prasad, A. Alsadoon, A. Singh, and A. Elchouemi, "Lung cancer detection using ct scan images," *Procedia Computer Science*, vol. 125, pp. 107–114, 2018.

[7] N. S. Nadkarni and S. Borkar, "Detection of lung cancer in ct images using image processing," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019, pp. 863–866.

[8] P. Sangamithraa and S. Govindaraju, "Lung tumour detection and classification using ek-mean clustering," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2016, pp. 2201–2206.

[9] D. P. Kaucha, P. Prasad, A. Alsadoon, A. Elchouemi, and S. Sreedharan, "Early detection of lung cancer using svm classifier in biomedical image processing," in *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. IEEE, 2017, pp. 3143–3148.

[10] B. Mithuna, P. Ravikumar, and C. Arpitha, "A quantitative approach for determining lung cancer using ct scan images," in *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2018, pp. 1786–1790.

[11] T. Patel and V. Nayak, "Hybrid approach for feature extraction of lung cancer detection," in *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*. IEEE, 2018, pp. 1431–1433.

[12] S. Perumal and T. Velmurugan, "Lung cancer detection and classification on ct scan images using enhanced artificial bee colony optimization," *International Journal of Engineering & Technology*, vol. 7, no. 2.26, pp. 74–79, 2018.

[13] I. R. I. Haque and J. Neubert, "Deep learning approaches to biomedical image segmentation," *Informatics in Medicine Unlocked*, vol. 18, p. 100297, 2020.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[15] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE, 2010, pp. 253–256.

[16] W. Alakwaa, M. Nassef, and A. Badr, "Lung cancer detection and classification with 3d convolutional neural network (3d-cnn)," *Lung Cancer*, vol. 8, no. 8, p. 409, 2017.

[17] B. Kaggle, "Kaggle data science bowl 2017," 2017.

[18] K. Kuan, M. Ravaut, G. Manek, H. Chen, J. Lin, B. Nazir, C. Chen, T. C. Howe, Z. Zeng, and V. Chandrasekhar, "Deep learning for lung cancer detection: tackling the kaggle data science bowl 2017 challenge," *arXiv preprint arXiv:1705.09435*, 2017.

[19] U. Kamal, A. M. Rafi, R. Hoque, M. Hasan *et al.*, "Lung cancer tumor region segmentation using recurrent 3d-denseunet," *arXiv preprint arXiv:1812.01951*, 2018.

[20] S. Hossain, S. Najeeb, A. Shahriyar, Z. R. Abdullah, and M. A. 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)*. IEEE, 2019, pp. 1348–1352.

[21] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, I. Eric, and C. Chang, "Deep learning of feature representation with multiple instance learning for medical image analysis," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 1626–1630.

[22] D. Kumar, A. Wong, and D. A. Clausi, "Lung nodule classification using deep features in ct images," in *2015 12th Conference on Computer and Robot Vision*. IEEE, 2015, pp. 133–138.

[23] W. Sun, B. Zheng, and W. Qian, "Computer aided lung cancer diagnosis with deep learning algorithms," in *Medical imaging 2016: computer-aided diagnosis*, vol. 9785. International Society for Optics and Photonics, 2016, p. 97850Z.

[24] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Artificial intelligence and statistics*, 2015, pp. 562–570.

[25] N. Ibtehaz and M. S. Rahman, "Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation," *Neural Networks*, vol. 121, pp. 74–87, 2020.

[26] P. Afshar, A. Mohammadi, K. Plataniotis, K. Farahani, J. Kirby, and e. a. Oikonomou, A, "Lung-originated tumor segmentation from computed tomography scan (lotus) benchmark," http://i-sip.encs.concordia.ca/datasets.html#Radiomics, 2019.

[27] S. Borah, E. Hines, and M. Bhuyan, "Wavelet transform based image texture analysis for size estimation applied to the sorting of tea granules," *Journal of Food Engineering*, vol. 79, no. 2, pp. 629–639, 2007.

[28] S. Sidhu and K. Raahemifar, "Texture classification using wavelet transform and support vector machines," in *Canadian Conference on Electrical and Computer Engineering, 2005.* IEEE, 2005, pp. 941–944.

[29] S. Arivazhagan and L. Ganesan, "Texture segmentation using wavelet transform," *Pattern Recognition Letters*, vol. 24, no. 16, pp. 3197–3203, 2003.

[30] S. Livens, P. Scheunders, G. Van de Wouwer, and D. Van Dyck, "Wavelets for texture analysis, an overview," 1997.

[31] G. Lee, R. Gommers, F. Waselewski, K. Wohlfahrt, and A. O'Leary, "Pywavelets: A python package for wavelet analysis," *Journal of Open Source Software*, vol. 4, no. 36, p. 1237, 2019.

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

[33] G. Van Rossum *et al.*, "Python programming language." in *USENIX annual technical conference*, vol. 41, 2007, p. 36.

[34] F. Chollet *et al.*, "keras," 2015.

[35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[36] M. Anthimopoulos, S. Christodoulidis, L. Ebner, T. Geiser, A. Christe, and S. Mougiakakou, "Semantic segmentation of pathological lung tissue with dilated fully convolutional networks," *IEEE journal of biomedical and health informatics*, vol. 23, no. 2, pp. 714–722, 2018.

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