

## Article

# Using Double Convolution Neural Network for Lung Cancer Stage Detection

Goran Jakimovski <sup>1,\*</sup>  and Danco Davcev <sup>2</sup>

<sup>1</sup> Faculty of Electrical Engineering and Information Technology, ss Cyril and Methodius University, 1000 Skopje, Macedonia

<sup>2</sup> Faculty of Computer Science and Information Technology, ss Cyril and Methodius University, 1000 Skopje, Macedonia; dancodavcev@gmail.com

\* Correspondence: goranj@feit.ukim.edu.mk

Received: 17 December 2018; Accepted: 23 January 2019; Published: 28 January 2019



**Abstract:** Recently, deep learning is used with convolutional Neural Networks for image classification and figure recognition. In our research, we used Computed Tomography (CT) scans to train a double convolutional Deep Neural Network (CDNN) and a regular CDNN. These topologies were tested against lung cancer images to determine the Tx cancer stage in which these topologies can detect the possibility of lung cancer. The first step was to pre-classify the CT images from the initial dataset so that the training of the CDNN could be focused. Next, we built the double Convolution deep Neural Network with max pooling to perform a more thorough search. Finally, we used CT scans of different Tx cancer stages of lung cancer to determine the Tx stage in which the CDNN would detect possibility of lung cancer. We tested the regular CDNN against our double CDNN. Using this algorithm, doctors will have additional help in early lung cancer detection and early treatment. After extensive training with 100 epochs, we obtained the highest accuracy of 0.9962, whereas the regular CDNN obtained only 0.876 accuracy.

**Keywords:** computed tomography; deep neural networks; image recognition; lung cancer; medical imaging

## 1. Introduction

Medical treatment has always been done with symptoms-based analysis. This means that patients first have their symptoms analyzed and, if necessary, they are sent for a more precise analysis (specialists and scans). Nowadays, the concept of “precise medicine” tries to solve the problem of the vast but fractured state of biomedical data. This is done using patient-centric appointments and storing the digital data of the patients in shareable online databases [1]. Furthermore, the European Medical Association, the World Health Organization and the United States Association have found that there is an enormous increase in lung cancer in the United states and Europe, making lung cancer the number one cause of death in Europe and the US, [1]. Latest developments in deep learning and Deep Neural Networks (DNN) have improved the process of image recognition. Using Deep Neural Networks, we can search for patterns in an image and determine if we recognize the pattern. Furthermore, when analyzing an image, we can search for multiple patterns. Training the Neural Network often requires a dataset that is predetermined, which the network can use to learn, recognize and classify an image.

Deep Neural Networks are becoming more and more popular as they can be easily applied to image pattern recognition and image classification. Few other derivative methods have emerged, such as Template Matching, Support Vector Machine, Deep Restricted Boltzmann, Stacked Autoencoders and Deep Convolutional Networks [2,3]. Convolutional Deep Neural Networks have

presented better performance over traditional Deep Neural Networks [3]. The authors of [4] used cifar-10 and mnist datasets to test and evaluate the traditional DNN with the convolutional DNN. In [4], the authors mapped the layers of the DNN to a dynamic image and the results show how the quantity of the layers influences the dynamic image recognition. Furthermore, the authors of [5] used modified AlexNet model, where, instead of using back propagation, they used unsupervised Sparse Autoencoder Machine. By using this autoencoder, they could accelerate the success of the learning feature of the Deep Neural Network to 90.1%. In [5,6], the authors trained and tested a convolutional DNN with Synthetic Aperture Radar (SAR) images, where the algorithm classifies these images into predetermined classes.

Feature extraction and image classification with Deep Neural Networks can be applied in different medical areas. For example, the authors of [7] tried to diagnose Alzheimer's brain early. They used functional Computed Tomography (CT) data to train and test the Deep Neural Network model. They used only one convolutional layer to classify the images. A comparative study is made in [8], which includes an analysis of how the parameters influence the training process of a DNN to classify images of lung cancer. The authors used 450 images of five patients, which were predetermined by a medical personnel as cancerous or not. In [9–12], the authors used lung nodules to extract nodule images from patients and trained a DNN to classify an image. They used 20 patients with around 3500 nodule images, which they put into a feature vector to train the network. In [13], they also used lung nodule images, but used them on a multiple resolution residually connected network, where they used 304 images to train the network.

It is challenging to use “heavier” images to classify using DNN. However, in [14,15] the authors used supervised 3D Convolutional Neural Network, where the input vector was a 3D object used to train the double convolutional Deep Neural Network (CDNN) and detect lung cancer.

Although CT-type images are mainly used in medical imaging, they can come with unnecessary artifacts. In [16–19], the authors used thresholding to avoid these artifacts in the images. The thresholding technique removes unwanted peaks in the pixels of lung cancer medical images (mostly grayscale). This means that medical personnel will work with images that have fewer artifacts. In [16,17], the authors used MATLAB algorithms to pre-process the image and remove such artifacts. The authors of [18,19] performed experiments with 30 and 50 lung cancer images and, based on statistical analysis, they determined the threshold to remove noise in the image. The threshold detection technique is used in [20] to remove noise in breast cancer RGB (color) images.

Another method of threshold detection to reduce artifacts in an image is the Otsu's method of binarization. In [21,22], the authors first used the Otsu's method to determine the threshold and then applied statistical analysis to adjust and optimize the threshold value. Using this optimized threshold value, pixel peaks are removed and images have less noise (artifacts). The authors used one- and two-stage Otsu's thresholding methods to compare them with 2D and 3D images. Unlike the work in [16–22], our threshold detection is defined as determining the Tx stage where our algorithm can detect the possibility of cancer.

This paper is organized as follows. Section 2 presents state-of-the-art solutions to the main problem in our research, which is classification of medical images and detection of cancer. We present how other types of medical images are classified and present other systems for medical image classification regarding lung cancer. Section 3 presents how the data were prepared as well as the defining, training and testing of our DNN. It also presents CT images of the different layers and how they communicate to output the best result. Section 4 presents an additional dataset of 35 patients used to determine the cancer stage in T 2, 3 and 4. The cancer stage detection and the results are presented in Section 5. Section 6 concludes the paper.

## 2. State of the Art for Medical Imaging Classification Solutions

Properly trained Deep Neural Networks usually require large dataset and adjustment of many parameters to get borderline (poor) results, assuming that the algorithm does not over-fit. Since a large

number of medical images is difficult to obtain, other methods can be used to train the DNN. In [23], the authors used active learning to help with the dataset, that is, to help with selecting and classifying the images before training. They used multistage training scheme to overcome the overfitting problem, which means that they started with a smaller dataset and reduced it to the point where there is no overfitting. For each step, they predicted the amount of data they needed to send the DNN and measured if and when overfitting happens.

To train the network with “heavy” multimedia, one needs to have large set of input nodes to pass the information through the network. In [24,25], the authors used extremely large Computer-Aided Detection (CAD) 3D images of lung cancer to provide the classification. To achieve this, they used U-Net LUNA 16 labeled data nodules to pass throughout the network. They had smaller pieces of the images already divided into nodules (pieces) that were pre-labeled as malignant or not. This way, the entire image is not taken into consideration, but the small nodules that are directly mapped to the nodes of the network.

Images can be classified (using CDNN) in more ways, not just into piles of cancerous or non-cancerous. In [26,27], the authors used fluorodeoxyglucose positron emission tomography (FDG-PET) images to determine the Tx stage of the cancer. They defined four piles of T1–T4 stages of cancer and determined the outcome of the classification. As an extended research [26], the authors of [27] additionally used CAD images and compared the results with FDG-PET.

We propose a lung cancer medical image classifier that is based on a Convolutional Deep Neural Network. To train and test our system, we used CT images of lungs that were previously classified by medical specialists and put into piles of yes/no (yes, the patient is diagnosed with lung cancer; and, no, the patient is cancer-free). Similar to Stanitsas and Cherian in [23], we pre-classified the images, but, in our case, we pre-classified them into groups of slice images taken from the same angle of the lung from different patients from our training dataset. Our system was trained using these images to be able to classify a new (previously unknown) image into one of the two piles (pile of cancer or pile of cancer-free) and tested the network to determine the success rate. Similar to the authors of [24,25], we divided the image into smaller pieces (using the convolution layer). Unlike the work in [24,25], our algorithm uses the entire image (combined with the pieces) for each following layer, reduced with a max-pooling algorithm. When the initial success rate of training the network was satisfactory (fit value), the topology was saved and further asynchronously tested against an additional dataset. This additional dataset was composed of images outside of the initial dataset of CT lung images and contains medical images of predetermined lung cancer images in stages 2, 3 and 4. Our algorithm, unlike the algorithm in [26,27], uses these three stages of lung cancer and determines in which of these Tx stages our algorithm can detect the possibility of a cancer.

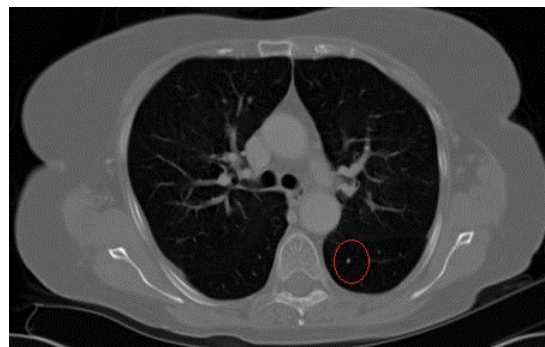
### 3. Medical Image Classification Using Double DNN

Image recognition in Deep Neural Networks is based on image classification, where the Neural Network is trained to classify an image into a list of predetermined piles or types, [28]. In its simplest form, it is used to determine if something is recognized or not. In our case, we tried to classify medical images and determine if there is cancer or not, thus we can simplify the outcome of the recognition as YES/NO (YES there is cancer or NO there is not). The Neural Network has to be trained so that it can be used for image classification. This process takes a list of input data, which are fed to the network and the outcome is compared to the expected outcome. The input data in our case were a pile of CT images that were fed to the network, so the input layer could have as many input nodes as the size of the array. This way, the input layer would have many nodes and the training would be slower and the network may overfit. Thus, we added additional layers (i.e., max-pooling algorithm) that downsized the input data. The output of the network can be a single node 0/1 or an array. The exit layer, in our case, outputs a single decimal value that is between 0.0 and 1.0 (0.0 (not cancer) or 1.0 (cancer)).

### 3.1. Data Preparation

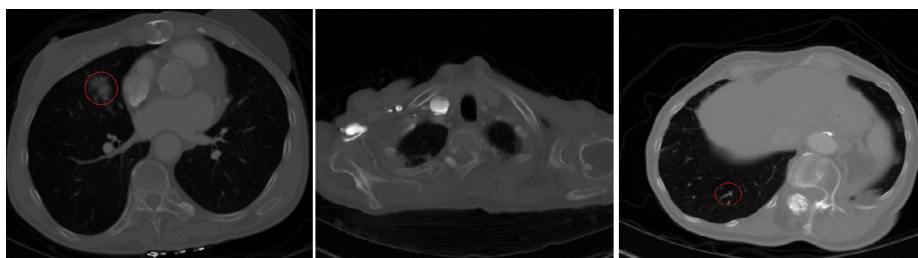
The data preparation is a crucial part in DNN training and testing. In our case, data preparation was done in several stages. The CT images were obtained from the Image & Data Archive of the University of South Carolina and the Laboratory of Neuro Imaging (LONI) database, ([ida.loni.usc.edu](http://ida.loni.usc.edu)). These images were analyzed and classified by medical personnel (as cancerous or not) by performing a biopsy of the lung cancer tissue to ensure high level of certainty about the labeling. The initial dataset contains CT scans of patients. When a patient does a CT scan, the scanner takes many images of the lung of the patient; each of these images is from a different part of the lung. These images are called slices (different angle images), which capture different parts or angles of the lung. Thus, one CT scan of one patient can produce many slices, and each of these slices is saved as an image. First, the initial dataset was divided into two piles; the first pile contained images from patients that were diagnosed with cancer, and the second pile contained images from patients without cancer. Thus, the two piles divided the images into cancerous or cancer-free. Next, the images of the two piles were further divided into groups, where each group contained images (slices) from the same part of the lung but from different patients.

The initial dataset had 95 patients, where each patient had gone through one CT scanning process. One such process produced 64 CT images (slices) of the chest of the patient. This means that the initial dataset had 6080 images that the DNN used for training and testing. One of the 64 slices (images) is shown on Figure 1. Next, these images were labeled by medical personnel as cancerous or cancer-free and the initial dataset was divided into the two piles. In our case, we had 73 patients who were diagnosed the possibility of cancer (Pile 1) and 22 who were without cancer (Pile 2).



**Figure 1.** A cancerous Computed Tomography (CT) image predetermined by medical personnel and confirmed by using biopsy of the tissue.

Next, we needed to further group the images in the two piles, where each group represented images from same slice (angle of the chest of the patient). This way, we created 64 groups within each pile, where each group in Pile 1 (cancer pile) had 73 CT images. Each of the 64 groups of Pile 2 (cancer-free pile) had 22 CT images. A sample of different angle (slice) CT images is shown on Figure 2. The middle image is cancer-free (belongs to Pile 2) and the other two are with marked location of the cancer (from Pile 1). Creating the piles of images was done so that we can create positives and negatives to train and test the network, but the groups in each pile were created so that Deep Neural Network would focus on recognizing same slice (angle) images. The groups were created using K-means algorithm to group the image into the appropriate slice group. The reason we used the K-means clustering is because we had CT images that were taken from 16-, 32-, 64-, 128-, 256- and 320-slice CT scanners. We used 64 groups because most of the images we had were 64-sliced. For images obtained from a scanning that made more or fewer than 64 slices, we used the K-means algorithm to put them into the correct group.



**Figure 2.** Different angles of CT lung cancer images.

The number of slices determines the distance between one scanned image of the body to the next. If there are more slices, the distance between each slice is smaller, but we get more information about the patient. Furthermore, one scanner from one manufacturer can output images from first to last (or vice versa) or save them in files in their own file format name, thus even if patients were scanned by a 64-slice scanner, if images were from a different manufacturer or version of the software, the images might not be in the same order. We could discard images that are not compliant to our scanner, but since CT images are hard to obtain, we did not discard images. The K-means algorithm we used is given in Equation (1).

$$J = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - C_j\|^2 \quad (1)$$

where “ $k$ ” is the number of groups (in our case 64), and we made about  $n = 100$  test cases where the image should be placed. More tests ( $n$ -parameter) lead to a more precise estimation (classification) of the image. In our case, by using basic trial-and-error method, we found that 100 cycles were the smallest sufficient number of tests to classify a CT image into the appropriate slice group.

The image that we tried to group is  $X_i$ , and we compared that image to pre-group one from each group,  $C_j$ . We picked a referent group of pre-classified images  $C_j$  and used them to determine the distance function (Euclidean distance)  $\|X_i - C_j\|$  and cluster the new image  $X_i$ . Since Euclidean distance is the shortest distance between two points, we calculated the smallest difference between the image we tried to group and the pre-group reference image. The distance between two images is the difference between them, i.e., how close they are to each other. We calculated the structural similarity index of two grayscale images (Python function `compare_ssim` from `skimage` Python package). This function returns the score of the comparison and the difference between the two images. The group with the smallest difference is where the image  $X_i$  was placed.

Additionally, the algorithm further divided the groups into training and testing sets. In our case, we used 10% of the images for testing and the other 90% for training. The data were loaded from a batch of images and the images were divided into training and testing sets (a subsets  $[X, Y]$ ). Each of these subsets was accompanied by a set of 0s and 1s indicating that the image was cancer-free or cancerous (this output was evaluated by medical personnel, in this case an oncologist). Finally, the data were shuffled, converted into binary class matrix and fed to the Neural Network.

### 3.2. Defining, Training and Testing the DNN

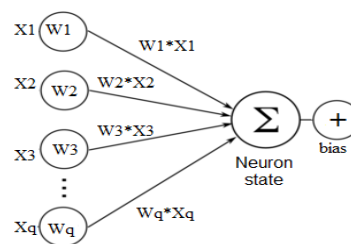
Once the images were ready and in binary matrix form, they were fed to the DNN and trained and tested. However, the network model had to be created so that it could be trained and tested. The creation of the network model means defining the parameters and layers of the DNN. After images were ready to be used for training (Section 3.1), we prepared the Neural Network with additional layers to create the Deep Neural Network. The inner layers are composed of one convolution layer, max pooling layer, followed by double convolution layers (two convolutions) and an additional max pooling. The first convolution layer does the initial segmentation of the images and the interconnection of the nodes. Next, we needed to reduce the size of the data with the max-pooling layer (to avoid over-fitting). The second and the third convolution was done so that we could make a more thorough



search of the problem (the cancer) and obtain more precise information of where the cancer might be. The connection between the convolution layer and the DNN is provided by Equation (2). Equation (2) shows how the state of one isolated neuron is calculated using convolution, if there were  $q$  input connections.

$$F^H = \sum_{i=1}^q W_i * x^i + b \quad (2)$$

We can see from Equation (2) that the state of the neuron with one convolution layer is  $F^H$ , where we have  $H$  kernel filtered images, which use filters  $W$  and a bias factor  $b$ . The bias factor  $b$  can have value 0 or 1, telling the network whether to include that neuron.  $x^i$  is the value of the input nodes of the previous layer to the  $i$ -th node of the current layer. Equation (2) is illustrated on Figure 3.

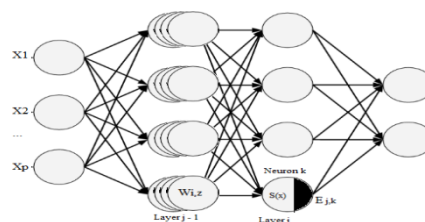


**Figure 3.** Calculation of the state of an isolated neuron.

The calculation of the correlation between the convolution and the DNN presented in Equation (2) is further explained in Equation (3). Equation (3) is Equation (2) with the addition that the neuron is now a part of a hidden convolution layer and the output energy of the neuron  $E_{j,k}$  is calculated as:

$$E_{j,k} = \sigma \left( b + \sum_{i=0}^q \sum_{z=0}^l w_{i,z} * X_{ii,zz} \right) \quad (3)$$

where the energy output  $E$  is calculated for neuron  $k$  in layer  $j$  of the DNN where we use  $\sigma$  (sigmoid function) to calculate the fire value. Again, we used the bias factor  $b$ , the input connection weights of each neuron in layer  $j-1$  denoted as “ $w$ ”, and we convoluted that value with the input from the nodes in the previous layer, denoted as “ $x$ ”. “ $q$ ” and “ $l$ ” represent the size of the input matrix of shared weights of  $W$ , and “ $ii$ ” and “ $jj$ ” are the indexes of the input activation at position  $(j + i, k + z)$ . The calculation of the energy output  $E_{j,k}$  is illustrated on Figure 4.

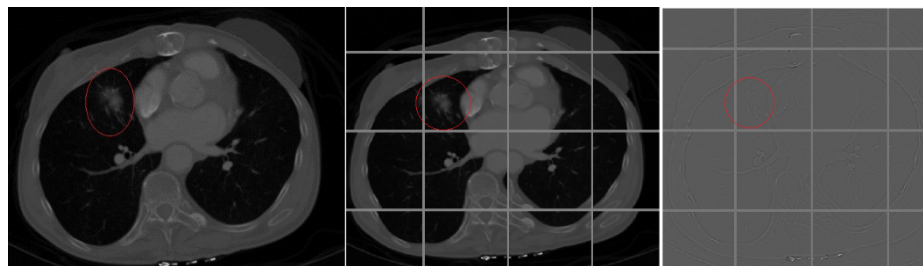


**Figure 4.** Calculation of the energy output of neuron  $k$  in layer  $j$  in a multi-layered Neural Network.

When the DNN is trained, the bias factor can be more precisely calculated and the correlations between the nodes adjusted. What the convolution does to an image is shown in Figure 5.

As shown in Figure 5, we further divided the image into smaller parts, where the parts overlap. This way, we could focus on (isolate) a certain part of the image and use that (smaller) image to search for a pattern. In our network, we defined the convolution parameters and the slicing window. Our convolution layer takes input of  $128 \times 128 \times 1$  (width  $\times$  height  $\times$  color). We used 1 for color (depth) since the image was grayscale. Furthermore, we used Rectified Linear Unit as an activation

function, which means that all negative values of activation are replaced with a value of 0. There is a tradeoff here as to how much overlapping there should be. If we increase the overlapping, we make more window images and thus more detailed search. However, by doing so, we slow down the process of learning and classifying, as the window-cutting requires more resources. Since the convolution results in more smaller images from the original one, by using max-pooling, we reduce the size of these images into chunks of data, where we get the most (maximum) of every image. This means that we searched for the cancer by upsizing the image in one layer and downsizing the results in the next by maximizing the bias (similarity) between adjacent kernels of the convolution. In the convolution function, we mainly used sharpening and edge detection filters. The filter in the convolution is a simple matrix that convolutes the image matrix and the result is another image whose edges are sharpened. The resulting image of the convolution filters is the third image in Figure 5.



**Figure 5.** Convolution of a CT scan image with kernel filters for edge sharpening filters.

Before we trained and tested the network, we had to define the learning rate and the dropout factor. In our case, the algorithm drops out elements that have below 50% success rate (by testing different models, we found that 50% dropout rate is most optimal for image classification). The calculation of the half square-error cost function is done by Equation (4). As we can see from Equation (4), we used half square-error cost function, that is, we back propagated the error to correct the previous layers and the convolution bias factor.

$$J(W; b; x; y) = \frac{1}{2} \|h_{w,b}(x) - y\|^2 \quad (4)$$

In Equation (4), as input, we take the connection weights of the network as  $W$ , the bias  $b$ , the input weights of the nodes  $x$  and the expected outcome  $y$ . The output of the network  $h_{w,b}(x)$  is calculated against the expected output  $y$  and one half of the error is propagated throughout the network.

Once the network was defined, the algorithm was then executed using the network parameters and the result of the training and testing is the DNN topology. The training process took the pre-prepared dataset presented in Section 3.1 and defined the  $(X, Y)$  as 90% (5472 images) of the dataset and testing  $(X_{\text{Test}}, Y_{\text{Test}})$  as the remaining 10% (608 images). Each epoch (train cycle), the algorithm passed all images once through the training process. Our training process had 100 epochs to train 95 CT images in one pile of 64 piles. After the training was finished, the network's topology was saved and could be used to classify CT images and determine the possibility of a cancer. In the process of training and testing, the network calculated the fit value of the topology by evaluating the number of properly classified images against the error.

In the training and testing process, we evaluate the system and its accuracy by making average of the epochs and how the algorithm classified the images. For defining the DNN and training and testing it, we used Tensorflow-GPU version 1.8 (Google, Montingeville, CA, USA) compiled for CUDA GPU version 7.1 (NVIDIA, Santa Clara, CA, USA) and Keras libraries version 2.1 in Python combined with native Python libraries to prepare the data. The algorithm was executed on a GPU-NVIDIA Corporation GM200 machine (NVIDIA, Santa Clara, CA, USA) equipped with about 1000 GPU GeForce cores. After the network was defined, trained, tested and cross-validated (this process took several hours), the topology was saved and used to classify new images outside the initial dataset.

The classification of new images took a few seconds, which means that medical personnel and patients would have initial diagnosis in just seconds after the CT scanning is finished.

We defined the cancerous images as positive images (4672 images), and the cancer-free images as negatives (1408 images), and we calculated the true positives (accurately classified positives), true negatives (accurately classified negatives), false positive and false negative (inaccurately classified positives and negatives). The averaged results of all 100 epochs are shown in Table 1.

**Table 1.** Averaged results of the classification of lung cancer images. CDNN: convolutional Deep Neural Network.

	Regular CDNN	Double CDNN
True Positive (TP)	4029	4653
True Negative (TN)	1303	1404
False Positive (FP)	643	97
False Negative (FN)	105	4

Using these parameters, we calculated the accuracy, sensitivity, specificity and positive prediction values of the two algorithms. The accuracy, shown in Equation (5), gives us the certainty of prediction or how accurate is the system. Furthermore, the sensitivity in Equation (6) gives us the measure of how the dataset is ready for classification or the measure of how accurate is the information it provides.

The specificity in Equation (7), on the other hand, gives us the ratio of how many of the cancerous images were classified as cancer-free, against other false-classified images of the dataset. We used an additional parameter called positive predictive value, shown in Equation (8), which indicates how much of the cancer has affected the patient (to determine the probability that the patient has cancer).

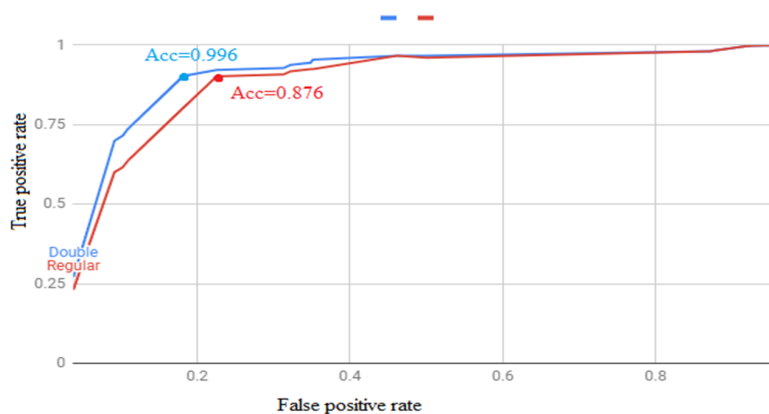
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (5)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (6)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} \quad (7)$$

$$\text{PositivePredictive} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (8)$$

We tested different thresholds for classification of the images and plotted the values of the results on a Receiver Operating Characteristic (ROC) curve to determine the best classification threshold. The ROC is presented on Figure 6.



**Figure 6.** Receiver Operating Characteristic (ROC) curves of different classification threshold with regular and double convolutional Deep Neural Network (CDNN).



From the analysis of the ROC curve, we found that 0.76 threshold gives the best accuracy of classification. The results are given in Table 2. In Table 2, we can see that our double CDNN algorithm has an almost 99.6% accuracy with a 0.76 threshold value, whereas the regular CDNN had the highest accuracy of 87% with a 0.70 threshold value. The sensitivity of the data was similar in both cases and this was expected since they used the same dataset. In addition, we can see in Table 2 that our double CDNN obtained higher results for prediction of cancer than the regular CDNN.

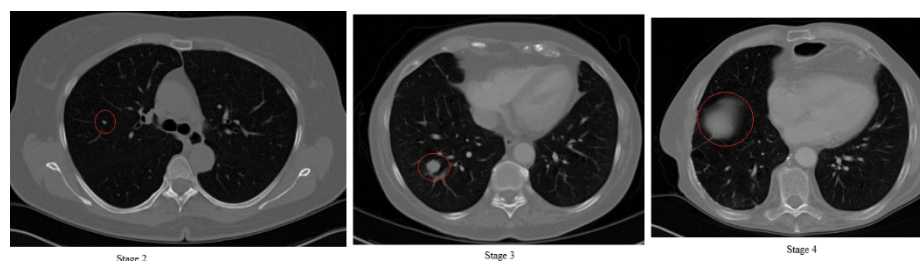
**Table 2.** Measurements of accuracy, sensitivity, specificity and positive predictive against regular and double CDNN.

	Regular DNN	Double CDNN
Accuracy	0.8769	0.99621
Sensitivity	0.97460	0.99912
Specificity	0.66957	0.98664
threshold classification	0.70	0.76

We discuss these topologies in Section 4 and the determination of the Txstage at which regular and double CDNN can detect the possibility of cancer.

#### 4. Tx Stages of Lung Cancer of Our Double CDNN

After defining, training and testing our network (Section 3.2), we used the topologies of the regular and double CDNN in experiments with an additional dataset from 35 patients diagnosed with lung cancer in stages 2, 3 and 4 (images obtained from the medical hospital in Tetovo, Macedonia). Since it is difficult to obtain images in stages 0 and 1, we found 35 patients whose possibility of cancer was diagnosed in stage 2 and recorded up until late stage 4. In Figure 7, we can see example CT images of stages 2, 3 and 4 of lung cancer.



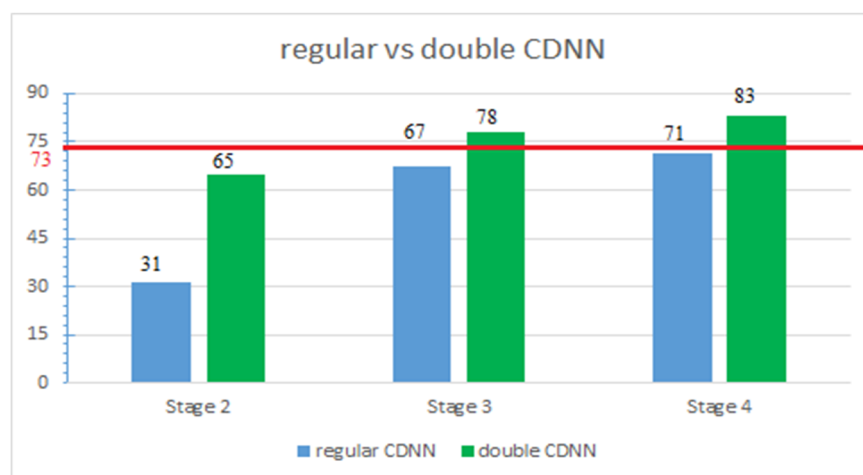
**Figure 7.** Stages 2, 3 and 4 of lung cancer.

From CT scan images, doctors can diagnose the stage only by using the size of the tumor (and in some cases the position of the tumor). Stage 2 (first image in Figure 7) shows the tumor in red circle on the left side, which in real size is around 4 cm. Stage 3 (second image in Figure 7) shows the cancer in red circle. In this stage, the tumor is larger than 4 cm and is in the middle of the lung and/or going towards the outer parts of the body. We can see in the second image in Figure 7 that the tumor is in late stage 3 since it leans towards the outer parts of the lungs. Stage 4 is in the third image in Figure 7 and we can see that the size of the tumor is covering large portions of the lung and is almost in the outer parts of the body and lung. This outer part of the lung is called area 1 and if the tumor is in this area (shown with red circle in the third image in Figure 7), it means that the cancer is terminal.

#### 5. Comparison of the Regular Against Double CDNN

We tested these images with standard Convolution Deep Neural Network, used by the authors of [28], against our double Convolution pre-clustered Deep Neural Network with edge sharpening filters. We used this test set of images to determine the threshold or Tx stage in which both networks can detect possibility of cancer. The networks output a decimal value from 0.0 to 1.0, where 1.0 is

cancer and 0.0 is cancer-free. We converted this value as a percentage of certainty and multiplied this by 100. The results of the two networks are shown in Figure 8. The drawback here is that we had to decide the minimal value of certainty we would accept as being satisfactory. To fairly compare both networks (regular and double CDNN), we took a mean of their best accuracy value. In Table 2, we can see that the best accuracy of the regular CDNN is 70%, and the best accuracy of the 76%, thus we used 73% as the minimal threshold value for certainty for cancer detection. Taking 73% as threshold for cancerous for both topologies and using this value as a threshold for cancerous, we can see in Figure 8 that our double CDNN detected cancer in stage 3, whereas the regular DNN from [28] did not detect cancer even in stage 4 (late stage). Taking the lower threshold value of 70% (Table 2), the regular CDNN detected possibility of cancer in late stage 4.



**Figure 8.** Results of classifying lung cancer images in stage 2, 3 and 4.

Our results were discussed and analyzed with medical personnel from the oncology department, of the hospital in Tetovo, Macedonia. The results were marked as satisfactory, since expert oncologists cannot determine possibility of cancer from a CT scan up to stage 2 or 3. Experts can have doubts of a possibility of cancer from stage 0, but will not schedule a biopsy of the tissue until late stage 2 or 3. The threshold is expected at this stage, since most of the cancerous images used (Section 3.1) for training and testing (Section 3.2) of the algorithms (both standard and double convolution DNN) were mostly from phase T3 or above.

## 6. Conclusions

The first novelty in our paper is using the K-means algorithm to pre-classify the images into piles of same slice images, where the DNN can focus on image classification of same slice images. The second novelty is the additional convolution layer with edge sharpening filters, to thoroughly search for cancer. Finally, the main novelty is testing our Deep Neural Network with lung cancer images from Tx stages 2, 3 and 4 and determining at which Tx stage the two algorithms can detect the possibility of cancer. The results were analyzed with medical personnel from the oncology department and were marked as satisfactory to determine cancer in T3 phase.

For future work, we plan on making a further analysis, where we will change the DNN to output 2 values (0 and 1) and determine which one has higher certainty of classification. This way, we can classify the image not just as being decimal value between 0.0 or 1.0, but also compare how much is 0 (not cancer) and how much is 1 (cancer). For additional future work, similar to Cruz-Roa and Arevalo Ovalle in [29], who used RGB (color) images to highlight the area of malignant cells, we plan on modifying the DNN to show us where (the location) on the CT image it has detected a cancer.

**Author Contributions:** G.J. and D.D. defined the problem to be detection of cancer in patients. Since G.J. obtained access to the image database of lung cancer, D.D. prepared the images to be fed to the algorithm. G.J. used the K-means algorithm to divide them into slice piles. D.D. adjusted the layers of the Deep Neural Network to reflect the algorithm. The additional tests with the stages II, III and IV was done by G.J. and D.D. The results from the tests, in the end, were analyzed by G.J. and medical personnel from the oncology department.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Vallone, S. LuCE Report on Lung Cancer-Challenges in Lung Cancer in Europe, Lung Cancer Europe. Available online: <http://www.lungcancereurope.com> (accessed on 18 June 2016).
2. Nguyen, K.; Fookes, C.; Sridharan, S. Improving Deep Convolutional Neural Networks with Unsupervised Feature Learning. In Proceedings of the ICIP, Quebec City, QC, Canada, 27–30 September 2015.
3. Guo, T.; Dong, J.; Li, H. Simple convolutional neural network on image classification. In Proceedings of the 2nd International Conference ICBDA, Beijing, China, 10–12 March 2017.
4. Ivanov, A.; Zhilenkov, A. The Prospects of Use of Deep Learning Neural Networks in Problems of Dynamic Images recognition. In Proceedings of the EIConRus, Moscow, Russia, 29 January–1 February 2018.
5. Huang, T.; Gao, F.; Wang, J. Combining Deep Convolutional Neural Network and SVM to SAR Image Target Recognition. In Proceedings of the International Conference iThings and IEEE GreenCom and IEEE CPSCO and SmartData, Exeter, UK, 21–23 June 2017.
6. Li, J.; Wang, C.; Wang, S.; Zhang, H.; Zhang, B. Classification of very high resolution SAR image based on convolutional neural network. In Proceedings of the International Workshop RSIP, Shanghai, China, 19–21 May 2017.
7. Sarraf, S.; Tofinghi, G. Deep learning-based pipeline to recognize Alzheimer’s disease using fMRI data. In Proceedings of the Future Technologies Conference, San Francisco, FL, USA, 6–7 December 2016.
8. Mesleh, A. Lung Cancer Detection Using Multi-Layer Neural Networks with Independent Component Analysis: A Comparative Study of Training Algorithms. *Jordan J. Biol. Sci.* **2017**, *10*, 239–249.
9. Kim, B.; Sung, Y.; Suk, H. Deep feature learning for pulmonary nodule classification in a lung CT. In Proceedings of the 2016 4th International Winter Conference on Brain-Computer Interface (BCI), Yongpyong, Korea, 22–24 February 2016.
10. Xie, Y.; Xia, Y.; Zhang, J. Knowledge-based Collaborative Deep Learning for Benign-Malignant Lung Nodule Classification on Chest CT. *IEEE Trans. Med. Imaging* **2018**. [[CrossRef](#)] [[PubMed](#)]
11. Jiang, H.; Ma, H.; Qian, W. An Automatic Detection System of Lung Nodule Based on Multigroup Patch-Based Deep Learning Network. *IEEE J. Biomed. Health Inform.* **2017**, *22*, 1227–1237. [[CrossRef](#)] [[PubMed](#)]
12. Nobrega, R.; Peixoto, S.; Silva, S. Lung Nodule Classification via Deep Transfer Learning in CT Lung Images. In Proceedings of the 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), Karlstad, Sweden, 18–21 June 2018.
13. Jiang, J.; Hu, Y.; Liu, C. Multiple Resolution Residually Connected Feature Streams for Automatic Lung Tumor Segmentation from CT Images. *IEEE Trans. Med. Imaging* **2018**, *38*, 134–144. [[CrossRef](#)] [[PubMed](#)]
14. Jin, T.; Cui, H.; Zeng, S. Learning Deep Spatial Lung Features by 3D Convolutional Neural Network for Early Cancer Detection. In Proceedings of the 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, NSW, Australia, 29 November–1 December 2017.
15. Fan, L.; Xia, Z.; Zhang, X.; Feng, X. Lung nodule detection based on 3D convolutional neural networks. In Proceedings of the 2017 International Conference on the Frontiers and Advances in Data Science (FADS), Xi’an, China, 23–25 October 2017.
16. Kanitkar, S.; Thombare, N.; Lokhande, S. Detection of lung cancer using marker-controlled watershed transform. In Proceedings of the 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015.
17. Miah, B.; Yousuf, M. Detection of lung cancer from CT image using image processing and neural network. In Proceedings of the 2015 International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, Bangladesh, 21–23 May 2015.

18. Koc, G.; Sarioglu, B. Statistical analysis of threshold algorithms in image processing based cancer cell detection. In Proceedings of the 2014 22nd Signal Processing and Communications Applications Conference, Trabzon, Turkey, 23–25 April 2014.
19. Taher, F.; Werghi, N.; Al-Ahmad, H. A thresholding approach for detection of sputum cell for lung cancer early diagnosis. In Proceedings of the IET Conference on Image Processing (IPR 2012), London, UK, 3–4 July 2012.
20. Cakar, E.; Turker, A.; Guleryuz, E.; Karaca, A. Detection of Candidate Nodules in Lung Tomography by Image Processing Techniques. In Proceedings of the 2017 21st National Biomedical Engineering Meeting (BIYOMUT), Istanbul, Turkey, 24 November–26 December 2017.
21. Swetha, T.; Bindu, C. Detection of Breast cancer with Hybrid image segmentation and Otsu's thresholding. In Proceedings of the 2015 International Conference on Computing and Network Communications (CoCoNet), Trivandrum, India, 16–19 December 2015.
22. Xue, J.; Titterington, M. t-Tests, F-Tests and Otsu's Methods for Image Thresholding. *IEEE Trans. Image Process.* **2011**, *20*, 2392–2396. [[PubMed](#)]
23. Stanitsas, P.; Cherian, A.; Truskinovsky, A. Active convolutional neural networks for cancerous tissue recognition. In Proceedings of the International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017.
24. Alakwaa, W.; Nassef, M.; Badr, A. Lung Cancer Detection and Classification with 3D Convolutional Neural Network (3D-CNN). *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*. [[CrossRef](#)]
25. Tafti, P.; Bashiri, F.; LaRose, E. Diagnostic Classification of Lung CT Images Using Deep 3D Multi-Scale Convolutional Neural Network. In Proceedings of the 2018 IEEE International Conference on Healthcare Informatics (ICHI), New York, NY, USA, 4–7 June 2018.
26. Kirienko, M.; Sollini, M.; Silverstri, G.; Moggetti, S. *Convolutional Neural Networks Detect Local Infiltration of Lung Cancer Primary Lesions on Baseline FDG-PET/CT*; MIDL: Amsterdam, The Netherlands, 2018.
27. Zong, Z.; Kim, Y. 3D fully convolutional networks for co-segmentation of tumors on PET-CT images. In Proceedings of the IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018.
28. Rossetto, A.; Zhou, W. Deep Learning for Categorization of Lung Cancer CT Images. In Proceedings of the IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies, Philadelphia, Pennsylvania, 17–19 July 2017.
29. Cruz-Roa, A.A.; Arevalo Ovalle, J.E.; Madabhushi, A.; González Osorio, F.A. A Deep Learning Architecture for Image Representation, Visual Interpretability and Automated Basal-Cell Carcinoma Cancer Detection. In Proceedings of the MICCAI, Nagoya, Japan, 22–26 September 2013.



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).