Lung Cancer Detection from Computed Tomography (CT) Scans using Convolutional Neural Network

M.Bikromjit Khumancha , Aarti Barai[y], C.B. Rama Rao[z]
Department of Electronics and Communication Engineering, National Institute of Technology Warangal
Email: bikramjitk8@gmail.com, [y]aarti.barai.10@gmail.com, [z]cbrr@nitw.ac.in

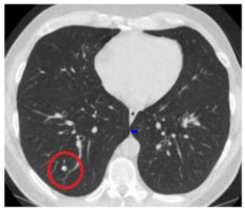
***Abstract*—With increasing patients of Lung Cancer every year, it is important to detect Lung Cancer so as to give proper medical treatments. Low dose CT Scan images are used for the detection of Lung Cancer. The first step is to detect the pulmonary nodules in lungs. LUNA16 data has 888 CT scans with annotated nodules in the CT scans. The annotation has coordinates of the lung nodules. A $32 \times 32 \times 32$ cube is made around the nodules with nodule as the centre. A 3D Convolutional Neural Network (CNN) is used to detect nodules using these cubes. For Lung Cancer detection, Data Science Bowl 2017 Kaggle competition data is used. It has 1595 CT scans. Lung nodules are predicted on this data using the nodule detector by running on the CT scans as grids. An ROI mask for lungs is applied to the CT scan using Image Processing. The predicted nodules coordinates are used to make cubes around nodules as the same size as before and a second 3D CNN is used to predict cancer using it. The model achieves precision and recall of 89.24% and 82.17% respectively.**

***Keywords*—Data augmentation, Convolutional Neural Network, Convolutional Layers, Max-Pooling Layers, Stochastic Gradient Descent, Learning Rate, Image Segmentation, Region Of Interest (ROIs)**

I. Introduction

Cancer is one of the leading reasons for death worldwide. Lung cancer results in most cancer deaths worldwide, accounting for 2.1 million new cases and 1.8 million deaths annually[1]. The death rate can be reduced if early diagnosis is done so that proper treatment can be carried out by the clinicians within the specified time. Cancer occurs when a group of cells undergo erratic growth uncontrollably and lose balance, forming malignant tumours which invade the neighbouring tissues. Stage of cancer indicates how broadly it has metastasized. During stage 1 and 2, cancers are localized to the lungs and in later stages, it spreads to other neighbouring organs. Present methods of diagnosis include biopsies and imaging like CT scans [2]. By early detection, lung cancer can then be identified at a time when it can be cured.

Traditionally, image processing algorithms have been used to extract features from images to distinguish between malignant and benign tumours. But distinguishing features between malignant and benign tumours is a complex task which also requires the development of hand designed features from the images [3,4]. A CNN has the ability to extract features by itself in an ordered way by use of multiple convolutional and max-pooling layers [5]. The traditional methods were used on 2D slices of CT Scan which has a drawback. The valuable spatial information or correlation between slices is lost if we use only a single slice for detection.



(a) Cancerous Nodule.

Fig. I: A slice of CT Scan containing cancerous nodule

We aim to detect the presence of early or non-early stage cancer in CT scan image of Lungs with the help of Image Processing and 3D CNN. The system will take the 3D CT Scans as input and give an output, whether the patient is cancerous or not. However, we cannot directly apply CNN on a CT Scan as it will be difficult to detect a small nodule in the large volume of 3D CT Scan. The solution is to reduce the search region by cropping out 3D cubes of small volume ,containing nodules from the CT Scan for training a nodule detector [6]. After the nodule detection is done, another 3D CNN can be trained for identification of cancerous and non-cancerous nodules. Figure 1 shows a slice of CT Scan containing a cancerous nodule.

II. Previous Work

Several systems have been proposed and still many of them have severe limitations. Artificial Neural Network based Classification and detection system of lung cancer [7,8], this system gives less accuracy. Automatic detection of small lung nodules on CT using a local density maximum algorithm [5] gives poor accuracy. Watershed gives segmentation results with relatively low computational cost but over-segmentation is the limitation [9]. Using Curvelet Transform to extract features and ANN have been used [10]. Recently Machine learning algorithms have been used widely. Lung Cancer Detection Using BPNN and SVM [11] have been used.

The available techniques of lung cancer detection use 2D images of the whole CT scan images and apply image processing on it to extract few calculated features. This limits the accuracy as it is missing the data of the whole three-dimensional structure of the lungs and the number of features used is limited. It is also difficult to detect small lung nodules in a full image of lungs. In our proposed structure, we divide the detection into two stages. First, we detect the nodules in a three dimensional cropped image of the scan and then predict cancerous nodules among those nodules. For this, we divide the image into multiple small images, so that the network achieves higher accuracy in finding the small nodules.

III. THEORY

Convolution is a mathematical operation of two functions that produces a new function. The output function gives light on the extent to which the two input function matches if their graphs are aligned with each other. In one dimension, convolution between two functions is defined as follows-

$$g(x) = f(x)h(x) = f(k)h(x-k)dk,$$

where f(x) and g(x) are two functions for which convolution is to be done. Computer applications have discrete data. So the equation of discrete convolution can be defined as follows-

$$g[n] = f[n]h[n] = k = -f[k]g[n-k]$$

In two dimension, convolution is defined as follows-

$$g(x,x) = f(x,y)h(x,y) = -f(s,t)h(x-s,y-t)dsdt$$

In machine learning algorithms, the inputs and kernels are generally multidimensional arrays of data. The kernel is adjusted by the algorithm in the training process. These kernels are also called tensors. In the case of three-dimensional images, the convolution can be explained as follows-

$$Y(i,j,k) = I(i,j,k)K(i,j,k) = mnpI(m,n,p)K(i-m,j-n,k-p),$$

where *I(i,j,k)* is the input image and K is the kernel.
The standard model of CNN consists of an input layer, alternating convolutional layers having max-pooling layers and non-linear layers or activation layers. The final layer consists of fully-connected layers. Every convolutional layer has a convolutional stage, activation layer (detection stage) and max-pooling (to reduce computation).

1.**Input Layer:** Input layer is the image fed into the network. It is a three-dimensional array of pixels.

2.**Convolutional Layer:** It is the most important part of CNN. Its job is to extract features from input. These layers have a series of kernels with which it has to perform convolution. The first layer extracts low - level features and as depth increases, the highest level of features are extracted.

3.**Activation Layer:** This layer introduces non-linearity into the model which facilitates learning complex data. The normally used functions for this model are

$$f(x) = tanh(x), f(x) = sigmoid(x) \ or \ ReLu(x) = Max(0,x)$$

$$ReLU(x) = Max(0,x)$$

ReLu steps up the speed of training the CNNs by keeping the gradient more or less constant at all network layers[8].

4**. Max-Pooling Layer:** It down-samples the input representation reducing the dimensionality and allowing assumptions to be made for the region on which max-pooling was done. Max-pooling is done by applying a max filter on the non-overlapping region of the initial region. The feature extraction is generally done using multiple similar steps with each step consisting of three cascading layers: Convolutional, Activation and Max-Pooling.

Suppose an image of size $MxMxN$ and we have p kernels with each of size $KxKxN$. One kernel convolves with one input feature map and similarly, p kernels produce p feature maps. These p feature maps are then applied to an activation function which adds non-linearity. Now, these feature maps are passed through a max-pool layer of kernel size of our choice.

In the last layer, we will have one output which will be used to determine how much the predicted and true labels deviate from each other. For these, we use loss functions like binary cross entropy, softmax, etc.

$$CrossEntropy = -t(log(f(s)) - (1-t)(log(1-f(s)))$$

where t is the actual value and f(s) is the predicted value.

The loss is then used for backpropagation, which looks for the minimum loss by adjusting the values of convolution kernels using gradient descent. The values in kernels that minimize the loss are taken as the solution of the network.

IV. DATASET

A. Lung Nodule Detection

The data used for nodule detection is from a competition called LUNA16 Grand challenge [13]. It has 888 CT scans. It is a subset of the LIDC/IDRI dataset from the Cancer Imaging Archive. The dataset has scans in which the slices are less than or equal to 2.5 mm. It contains annotations which were done by 4 experienced radiologists. All nodules which are greater than 3 mm and accepted by 3 out of 4 radiologists have been included while nodules smaller than that have been excluded. It also has nodules for false positive reduction. The annotation file is a CSV file with each finding per line. Each line has SeriesInstanceUID of the scan, the x,y and z coordinate of the nodule in world coordinates and the corresponding diameter in mm. The annotation has 408186 candidates.

B. Cancer Detection

The data used for Cancer Detection is from the Kaggle competition Data Science Bowl 2017 [14]. It has 1595 CT scans with an annotation file which has the patient ID and a value 1 or 0 according to whether the patient has cancer or not. The images are in DICOM format. The dataset has 1176 non-cancerous and 419 cancerous patients.

V. PROPOSED METHODOLOGY FOR NODULE DETECTION

A. Preprocessing

The first step of preprocessing is to make the CT scans homogenous. Every scan is rescaled so that each voxel represents a volume $1mm \ x \ 1mm \ x \ 1mm$. Then pixel values in each image are converted to Hounsfield units (HU), which is a measurement of radiodensity. Some CT scan scanners have cylindrical scanning bounds, but the output image is a square. Those pixels which fall outside this bound get the fixed value - 2000. These pixels are given a value 1. This is done to make all the unnecessary pixels black and to get the lung region properly. Now, the pixel values are scaled between 0 and 1. The next step is to make sure all the CT scans have the same orientation. The last step is to zero-centre the data by subtracting the mean of all the images [15].

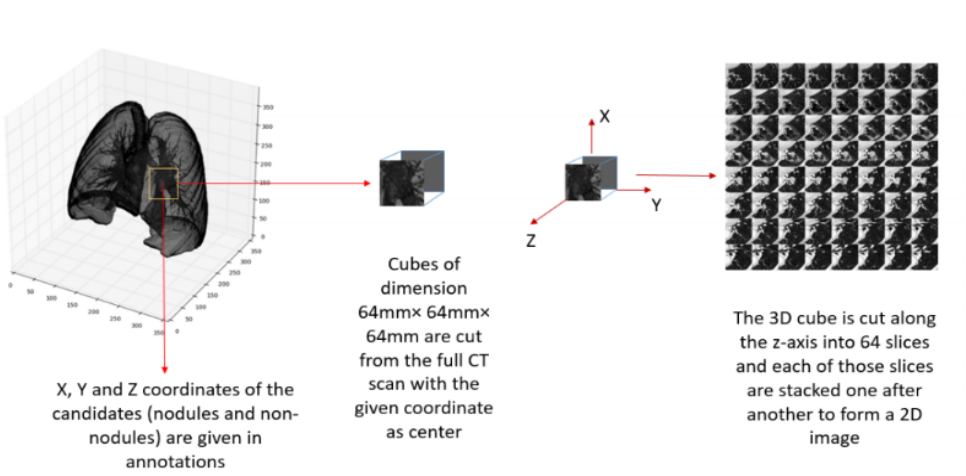
B. Creating Training Data for Nodule Detection

The annotations provided with the LUNA16 dataset contain 408186 candidates. Some candidates are also provided for False Positive reduction. The annotation has the coordinates of the candidate and diameter in mm. The number of annotated nodules in LUNA16 dataset is originally low (1186). So we used the LIDC/IDRI annotations to get nodules [16,17,18]. This increased the number of nodules to 5,000.

TABLE I : LUNA16 Label set

Description	Quantity
Nodule candidates	5,000
Non-nodules	400,000
Candidates for False Positive reduction	7,000

The CT scan images generally have sizes around $200mm \ x \ 200mm \ x \ 200mm$, while our nodules in early stages have sizes less than 10mm. So the amount of signal vs noise is around 1:1000,000. Due to this, a 3D CNN will not be able to learn from raw image data. This problem is solved by cropping small 3D cubes around the candidate coordinates. The size of the cube used is 32 32 32. Figure 2 illustrates the cropping of 3D cubes from the CT Scan. The low amount of nodules is handled by doing data augmentation (two loss-less data augmentation used-3D flipping and translation) [19].

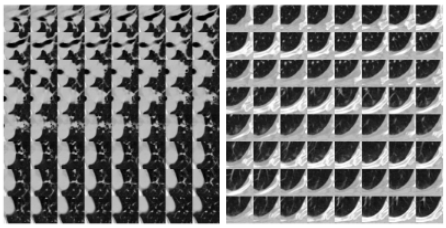


(a) Nodule.

Fig. 2: Illustration showing the cropping of 3D cubes of size $64mm\ x\ 64mm\ x\ 64mm$

After performing data augmentation, our dataset finally has 20,000 nodules,4000000 non-nodules and 7,000 candidates for false positive reduction. These are randomly shuffled and divided into training set, cross-validation set and test set. The train set, validation set and test set consist of 3,213,804 , 803,452 and 9,744 candidates respectively. The training is done on the raw image cubes of the lung CT scan.

A $64mm\ x\ 64mm\ x\ 64mm$ cube is shown in Figure 1. Each slice of the cube along the z-axis are stacked together and represented in a 8 by 8 grid image. The thickness of each slice is 1 mm. Figure 3:a is the image of a nodule. Figure 3:b is the image of a non-nodule. Each of these cubes is further cropped to $32mm\ x\ 32mm\ x\ 32mm$ before feeding to CNN.



(a) Nodule. (b) Non-nodule.

Fig 3: $64mm\ x\ 64mm\ x\ 64mm$ cubes cropped out from the lung region with the candidate as the centre. The slices along z-axis are represented as a grid image. The white portion corresponds to lung walls.

C. Training the 3D CNN for Nodule Detection

A 3D CNN is used for nodule detection. The CNN has convolutional layers with ReLU activation function for all except the last one. The last one has a sigmoid activation function which will give the value of whether a candidate is a nodule or not. The model uses Convolutional layers with padding to keep the output size the same as the input size. Max Pooling layers are used to down-sample the input and reduce its dimension[20].

As we go deeper into the layers, the input size decreases and the number of filters used for each layer increases. The minibatch size used is 32. Since the data used for training is huge, a generator is used to feed samples of the size of minibatch into the network without loading the whole dataset into the memory. The optimizer used is SGD (Stochastic Gradient Descent). Dropouts of 0.2,0.3 and 0.4 are used after second, third and fourth Max Pooling layer respectively, to prevent overfitting the model [21]. The Learning Rate initialized is 0.001, which is decreased to 0.0001 after fifth epoch using a learning rate scheduler. This decreasing of learning rate is done to converge our model properly to a local minima as a higher learning rate might make the Gradient Descent skip the minima. The loss function used is Binary Cross-entropy and the metrics used for validation are Binary Cross-entropy and Binary Accuracy. The network is run for 12 epochs with the weights saved after each iteration in HDF5 format. The weights giving the best validation accuracy is finally chosen. The model is trained in Keras with Tensor-flow backend in Google Collaboratory cloud.

A visualisation of the CNN architecture is shown in Figure 4 and its detailed description is given in Table II.

TABLE II: Architecture of the 3D CNN used for nodule detection.

Layer	Parameters	Activation	Output
Input			$32 \times 32 \times 32, 1$
Average Pool	$2 \times 1 \times 1$		$16 \times 32 \times 32, 1$
3D Convolution	$3 \times 3 \times 3$	ReLU	$16 \times 32 \times 32,64$
Max Pooling	$1 \times 2 \times 2$		$16 \times 16 \times 16,64$
3D Convolution	$3 \times 3 \times 3$	ReLU	$16 \times 16 \times 16,128$
Max Pooling	$2 \times 2 \times 2$		$8 \times 8 \times 8,128$
3D Convolution	$3 \times 3 \times 3$	ReLU	$8 \times 8 \times 8,256$
3D Convolution	$3 \times 3 \times 3$	ReLU	$8 \times 8 \times 8,256$
Max Pooling	$3 \times 3 \times 3$		$4 \times 4 \times 4,256$
3D Convolution	$3 \times 3 \times 3$	ReLU	$4 \times 4 \times 4,512$
3D Convolution	$3 \times 3 \times 3$	ReLU	$4 \times 4 \times 4,512$
Max Pooling	$2 \times 2 \times 2$		$2 \times 2 \times 2,512$
3D Convolution	$2 \times 2 \times 2$	ReLU	$1 \times 1 \times 1,64$
3D Convolution	$2 \times 2 \times 2$	Sigmoid	$1 \times 1 \times 1,1$

VI. Results Of Nodule Detector

The trained model of nodule detector is tasted on 9744 candidates. The results obtained are shown in the confusion matrix shown in Table III.

The precision, which tells us what proportions of positive identifications was actually correct , is 89.46% for the model.

The recall, which tells us which tells us what proportions of the actual positives was identified correctly, is 85.99%.

From the confusion matrix, 9233 non-nodules were correctly predicted to be non-nodule out of 9298 nodules. 65 non-nodules were wrongly predicted. 399 nodules were correctly predicted out of a total 446 nodules. 47 nodules were predicted incorrectly to be a non-nodule.

From these results, we can interpret that LUNA16 dataset is prone to False Positive prediction. Since we are now able to detect the small nodules in a large 3D CT Scan , we can crop these small nodules from the scans and form a dataset for the next stage of cancer detection. It will make easier the detection as we don't have to search the whole image for a cancer nodule. This will improve performance as well as accuracy.

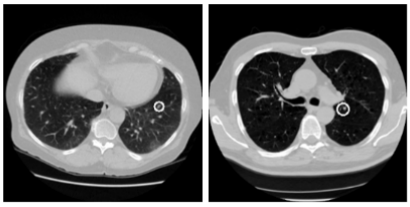
TABLE III: Confusion Matrix of Lung Nodule Detection

		Prediction outcome n	Prediction outcome p	total
actual value	n′	TN=9233	FP=65	N′ = 9298
	p′	FN=47	TP=399	P′ = 446
	total	P=9280	N=464	

VII. Proposed Methodology For Cancer Detection

A. Predicting Nodules in Kaggle Dataset

The nodule detector CNN model obtained before is now used to detect nodules on the Kaggle dataset and after that, the predicted nodules will be used for training a model to predict whether it is cancerous or not. Since our nodule detector predicted nodules in cubes of size $32mm\ x\ 32mm\ x\ 32mm$, we run the nodule detector through the whole 3D CT scan assuming it to be a grid made of cubes with strides of 12mm.
The 12mm stride ensures that the whole nodule is covered in any one of the cubes that go through the CT scan. If the probability of nodule is greater than a threshold(0.6), it is classified as a nodule, otherwise not. The predicted nodule coordinates (centre of the cube in which the nodule is found) is saved for each CT scan in a CSV file. Thus for each patient, we have coordinates of the nodules. Now, the next task is to predict cancer using it. Figure 5 shows the predicted nodules by our trained nodule detector on 2D slices of CT Scan. The slice having the predicted z-coordinate is selected and a white circle encircling the nodule is drawn using the x and y coordinates.



(a) p1. (b) p2.

Fig. 5: Nodules predicted by the nodule detector on 2D slices of CT scan (Encircled by a white circle).

B. Segmentation of CT Scan Images

Image segmentation, which aims at automated extraction of object boundary features, plays a fundamental role in understanding image content for searching and mining in medical image archives. A challenging problem is to segment regions with boundary insufficiencies, i.e., missing edges and/or lack of texture contrast between regions of interest (ROIs) and background. The segmentation of lungs is an important step because the region of interests lies inside the lungs. Our main task is to create a mask to separate the lung region from the other areas of the image. The segmentation used has the following steps-

1. Binarization: The first step of segmentation is binarization. Binarization is the process of converting a pixel image to a binary image. First, the image is converted to grayscale. An important task of this step is to select the threshold for binarization. It is found that Radiodensity(HU) for lung tissue is -500. So, a value of 604(-400 HU) is used as a threshold to separate lung First item tissue from others [15]. The Radiodensity(HU) of different materials are given in Table IV.

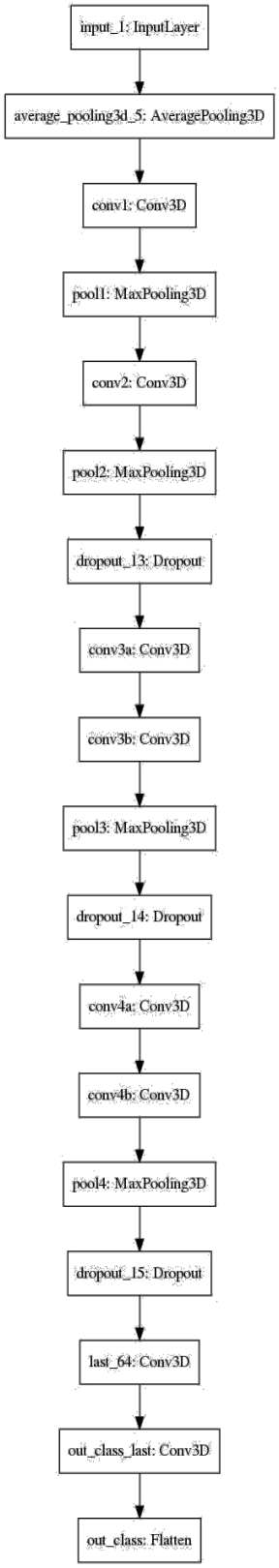


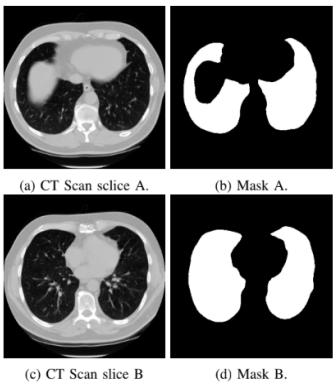
Fig. 4: Architecture of the 3D CNN used for nodule detection.

TABLE IV: Radiodensity(in HU) of various substances in a CT Scan[22]

Substance	Radiodensity(HU)
Air	-1000
Lung Tissue	-500
Water and Blood	0
Bone	700

2. Clear Border: The next step is to remove the blobs connected to the boundary of the lungs. This is done using clear border. Clear border suppresses structures that are lighter than their surroundings and that are connected to the image border
3. Labelling the Image: The next step is to label the image. A connected component in a binary image is a set of pixels that form a connected group. All the different connected components are assigned different labels. After labelling we keep the labels with the two largest areas.
4. Erosion: The next step is to perform erosion. Erosion is one of the popular techniques of morphological image processing. It is typically applied on binary images. The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels shrink in size, and holes within those areas become larger. This step is important because it separates the lung nodules from the blood vessels.
5. Closure Operation: The next operation is a Closing operation which is a dilation followed by an erosion. The basic effect of dilation on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (i.e. white pixels, typically). Closing fills holes in the regions while keeping the initial region sizes. This step is done to keep the blobs connected to the walls of the lungs.
6. Filling Holes: The next step is to fill the holes inside the binary mask of lungs.

The Figure 6 shows original CT scan images and their corresponding masks obtained after segmentation



(a) CT Scan sclice A. (b) Mask A.

(c) CT Scan slice B (d) Mask B.

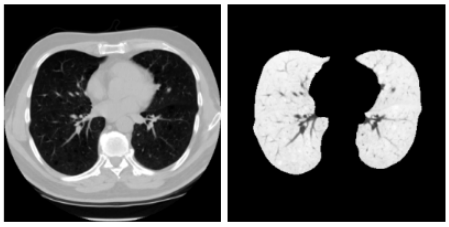
Fig. 6: Two slices of CT Scan and their corresponding masks.

C. Applying Mask on the Original CT Scan Image

The mask obtained in the previous step is multiplied with original CT Image for all the slices of the CT scan. Figure 7 shows the original image and image after applying the mask on the slice.

D. Making Training Data for Cancer Detection

The coordinates that were saved after nodule detection in Kaggle Dataset are taken as centre and training cubes of size $32mm\ x\ 32mm\ x\ 32mm$ are cropped from the images obtained after applying the masks. Figure 8 shows two such training cubes of size $64mm\ x\ 64mm\ x\ 64mm$ with slices along z-axis represented on a grid on images.



(a) An original slice of CT Scan (b) Image after applying mask

Fig. 7: Application of mask on lung CT Scan

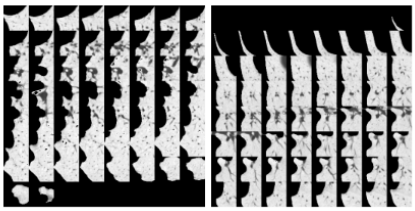


Fig. 8: Two $64mm\ x\ 64mm\ x\ 64mm$ cubes cropped out from the segmented image with the predicted coordinate as centre. The slices along z-axis are represented as a grid image. The white portion corresponds to lung walls.

Since the nodule detector has a high probability of False Positive prediction, the predicted nodules for each patient is arranged in descending order according to predicted probability by the nodule detector and the top four nodules with the highest probability for each CT Scan are taken. The training, validation and test set consist of 3,569, 608 and 608 nodules respectively.

E. Training a 3D CNN for Cancer Detection

The model for the detection of cancer is another 3D CNN. The model used is not very deep and has less number of filters for each convolutional layer to prevent overfitting due to the low amount of training data. The model uses ReLU activation function for all the convolutional layers except the last one. The last convolutional layer uses a Sigmoid activation function as it is used for making cancer predictions. Dropouts of 0.2 and 0.2 are used after the first and second Max Pooling layers respectively, to prevent overfitting the model [21]. The loss function used is Binary Cross-entropy and the metric used for evaluation is Binary Accuracy. The optimizer used is an Adam Optimizer. The Learning Rate is initialized with 0.001 and a time-based decay is used to reduce the Learning Rate after each epoch. This helps the model converge properly to a local minima. The size of the minibatch used is 16. A Generator is used to feed the data of minibatch size to the model without loading the whole training and validation dataset into the memory. The model is trained for 12 epochs. The architecture of the model is shown in Figure 7 and the detailed description is given in Table V.

TABLE V: Architecture of the 3D CNN used for cancer detection.

Layer	Parameters	Activation	Output
Input			$32 \times 32 \times 32,\ 1$
Average Pool	$2 \times 1 \times 1$		$16 \times 32 \times 32,\ 1$
3D Convolution	$3 \times 3 \times 3$	ReLU	$16 \times 32 \times 32,32$
Max Pooling	$1 \times 2 \times 2$		$16 \times 16 \times 16,32$
3D Convolution	$3 \times 3 \times 3$	ReLU	$16 \times 16 \times 16,32$
Max Pooling	$2 \times 2 \times 2$		$8 \times 8 \times 8,32$
3D Convolution	$3 \times 3 \times 3$	ReLU	$8 \times 8 \times 8,32$
Max Pooling	$2 \times 2 \times 2$		$4 \times 4 \times 4,32$
3D Convolution	$3 \times 3 \times 3$	ReLU	$2 \times 2 \times 2,32$
3D Convolution	$2 \times 2 \times 2$	Sigmoid	$1 \times 1 \times 1,1$

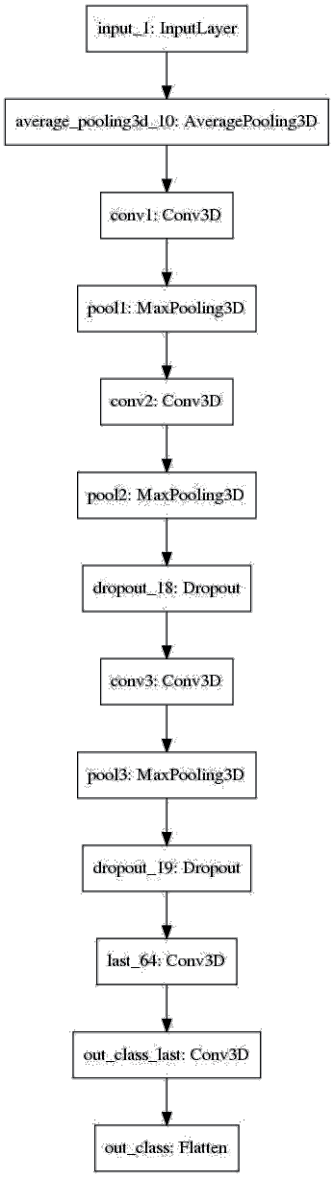


Fig. 9: Architecture of the 3D CNN for cancer detection

VIII. Results Of Cancer Detection

The model is tested on a test set containing 608 nodules. It is found that the number of observations that are sampled in each iteration, the size of the so-called minibatch, effects the performance of the model. Different minibatch sizes 8,16,32,64 were tried, out of which minibatch size 16 performed best. Initially, training with 30 epochs resulted in validation loss increasing after iterations around 10 indicating overfitting. So, after testing with different total number of epochs around 10, it is found that 12 as number of epochs works the best.

Different filter sizes are tried and it is found that a $2 \; x \; 2 \; x \; 2$ for Max Pooling and $3 \; x \; 3 \; x \; 3$ for Convolutional layer works the best. Initializing with a very low learning rate along with decreasing it at each step and a low number of filters for each layer also helped reduce overfitting. Our results suggest that the architecture of CNN that we have used is one of the optimal architecture for this task.

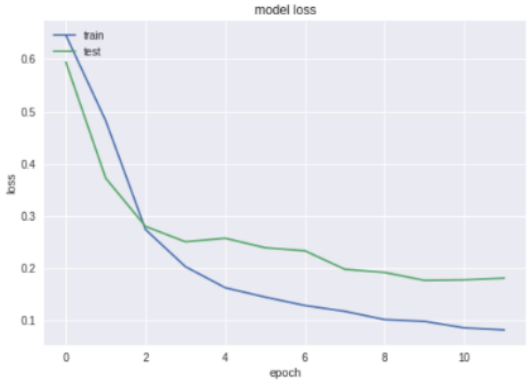


Fig. 10: Plot of loss function versus epoch for cancer detection

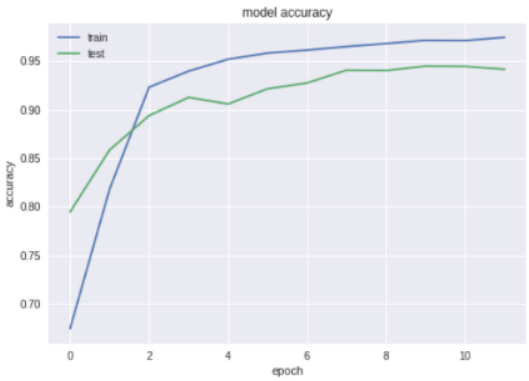


Fig. 11: Plot of accuracy versus epoch for cancer detection

The plot of loss function versus epoch and accuracy versus epoch for training and validation set is given in Figure 10 and Figure 11 respectively. From the loss function graph, it can be concluded that the model does not underfit as the loss is decreasing at each epoch. The gradual decrease of loss at each epoch shows that proper selection of learning rate and model architecture prevented overfitting. The confusion matrix is shown in Table VI.

The performance of the system gives satisfactory results and the precision and recall of the model are 89.24% and 82.17% respectively. On calculating the accuracy based on total true count to overall test data count, it leads to 90.78%.

IX. Conclusion

The 3D structure of nodules is used to extract features by 3D CNN to predict lung cancer. By using CNNs, the tedious task of manually extracting features can be eliminated. The obtained precision and recall obtained for the model are decent considering that less labelled data than most state-of-the-art CAD systems are used. Comparison with other Lung Detection Method is made in Table VII.

TABLE VI: Confusion Matrix of Lung Cancer Detection

		Prediction outcome: n	Prediction outcome: p	total
actual value	n′	TN=386	FP=20	$N' = 406$
	p′	FN=36	TP=166	$P' = 202$
total		P=422	N=186	

TABLE VII: Comparison To Other Lung Detection Existing Method

Lung Cancer Detection System	Accuracy
Only Image Processing Techniques	80%
Image Segmentation based on Gray Coefficient Mass Estimation with Gabor filter	83%
Linear CNN	66.5%
Proposed Method	**90.78%**

Further experimentations can be done by using widely used CNN architectures such as VCG [23] , ResNet [24] and GooLeNet [25]. Also, the model is saved at its best validation accuracy. Other metrics ,such as F1 can also be used. Other experimentations include making the network deeper and more extensive hyperparameter tuning. Finally, many models with different architectures and input sizes can be trained and an ensemble of these classifiers can be used to improve the cancer prediction.

X. REFERENCES

1. World Health Organization. International Agency for Research on Cancer. GLOBOCAN 2018: Estimated Cancer Incidence, Mortality and Prevalence Worldwide *http://gco.iarc.fr/today/data/factsheets/cancers/15-Lung-fact-sheet.pdf* ,2018.
2. Tests and diagnosis, *http://www.mayoclinic.org/diseases-conditions/lung-cancer/basics/tests-diagnosis/con-20025531*.
3. B. v. Ginneken, A. A. A. Setio, C. Jacobs, and F. Ciompi, *Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans*, 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), pp. 286-289 2015
4. J. Kuruvilla and K. Gunavathi, *Lung cancer classification using neu-ral networks for CT images,Computer Methods and Programs in Biomedicine*, vol. 113, no. 1, pp. 202-209, 2014.
5. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, *Gradient-based learning applied to document recognition, Proceedings* of the IEEE, vol. 86, no. 11, pp. 22782324, 1998.
6. J. d. Wit 2nd place solution for the 2017 *national datascience bowl*.
7. G. Bhat, V. G. Biradar , H. S. Nalini, "*Artificial Neural Network based Cancer Cell Classification (ANN - C3),*" Computer Engineering and Intelligent Systems, vol. 3, no. 2, 2012.
8. A. Pathan, B. K. Saptalkar, "*Detection and Classification of Lung Cancer Using Artificial Neural Network*," International Journal on Advanced Computer Engineering and Communication Technology, vol. I Issue. I, pp . 62-17, 2012.
9. Omid Talakoub,Javad Alirezaie,Paul Babyn,"*Lung Segmentation in Pulmonary CT Images Using Wavelet Transform*", ICASSP,IEEE,vol.1, pp. 1-453,2007.
10. B. Gupta and S. Tiwari, " *Lung Cancer Detection using Curvelet Transform and Neural Network*," International Journal Of Computer Science , vol. 86, no. I, pp. 15-17,Jan. 2014
11. G. Kaur and H. Singh, "*Lung Cancer Detection Using BPNN and SVM*," International Journal of Latest Scientific Research and Technology, vol. 1, no. 2, Jul. 2014, pp. 95-98, 2014.
12. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "*ImageNet Classification with Deep Convolutional Neural Networks*," In proceedings of Neural Networks (NIPS), Nevada, USA, pp. 1106–1114, 2012
13. LUNA16, Lung nodule analysis 2016. *https://luna16.grand-challenge.org/,* 2016.
14. Kaggle, Data Science Bowl 2017." *https:// www.kaggle.com/c/data- science-bowl-2017*".
15. A. Chon, N. Balachandar, and P. Lu , Data Science Bowl 2017." *Deep Convolutional Neural Networks for Lung Cancer Detection", 2017.*
16. S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, B. Zhao, D. R. Aberle, C. I. Henschke, E. A. Hoff-man et al. he *lung image database consortium (lidc) and image database resource initiative (idri): a completed reference database of lung nodules on ct scans*, Medical physics, vol. 38, no. 2, pp. 915931, 2011.
17. K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle et al. *The cancer imaging archive (tcia): maintaining and operating a public information repository*, 2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI), 2015.
18. Data from lidc-idri,2015. *http://dx.doi.org/10.7937/K9/TCIA.2015.LO9QL9SX.*
19. J. Lemley,S. Bazrafkan and P. Corcoran. *Smart Augmentation Learning an Optimal Data Augmentation Strategy*, arXiv:1703.08383v1 [cs.AI] 24 Mar, 2017.
20. A. Krizhevsky, I. Sutskever, and G. E. Hinton. *imageNet Classification with Deep Convolutional Neural Networks*
21. N. Srivastava, G. Hinton ,A. Krizhevsky ,I. Sutskever ,and R. Salakhutdi-nov Dropout: *A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research* 15 (2014) 1929-1958
22. H. Lepor Dropout: *A Simple Way to Prevent Neural Networks from Overfitting, Prostatic Diseases*, vol. 2000. W.B. Saunders Company, 2000.
23. K. Simonyan, A. Zisserman *Very Deep Convolutional Networks for Large-Scale Image Recognition*, Computer Science, 2014.
24. K. He, X. Zhang, S. Ren, Jian Sun *Deep Residual Learning for Image Recognition*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, 2016.
25. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich "*Going Deeper With Convolutions*", The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, 2015.