Lung Cancer detection using 3D Convolutional Neural Networks

Adarsh Pradhan[1], Bhaskarjyoti Sarma[2] and Bhiman Kr Dey[3]

[1]*Assistant Professor*
Department of Computer Science and Engineering
Girijananda Chowdhury Institute of Management and Technology, Azara, Guwahati - 781017, Assam, India
adarsh@gimt-guwahati.ac.in

[2,3,4]*Department of Computer Ssience and Engineering*
Girijananda Chowdhury Institute of Management and Technology,
Azara, Guwahati - 781017, Assam, India
{bhaskarjyotisarma85 & bhimandey}@gmail.com

***Abstract* - A large number of cancer deaths in the world is due to lung cancer, which is caused due to unbalanced cell growth. In this paper, we used 3D Convolutional Neural Network (CNN) for identification of lung cancer from the Computed Tomography (CT) scans of the patient, since CNN makes it easier to obtain the important information from the images. Here we use the SPIE-AAPM Lung CT Challenge dataset and employ different morphological preprocessing techniques like conversion to Hounsfield Unit, removing the air region and filling the lung area to obtain the lung nodule mask. We utilize our 3D CNN model for lung cancer detection and obtain a very good evaluation of the model. We divide our preprocessed dataset into 60%, 20% and 20% for training, validation and testing respectively, and obtain training accuracy of 83.33%, testing accuracy of 100% and precision, recall, kappa-Score, and F-score of 1.**

Index Terms –lung cancer, convolutional neural network, computed tomography, Hounsfield Unit.

I. INTRODUCTION

Lung cancer is a virulent lung tumor in which uncontrollable and unbalanced growth of cells ultimately leads to invasion of the neighbouring cells resulting in a cancer [1]. Lung cancer can be seen on computed tomography (CT) scans which is created by combining a series of X-ray images taken from different angles around the body. Most of the cancerous deaths in the world are due to lung cancer with 1.8 million deaths annually [2].

Deep Learning falls under machine learning which helps you to abstracts a model from the dataset by implementing deep neural networks. Deep neural networks can be used which would be an effective way for finding the cell growth region from the CT scans using image pattern recognition and image classification. According to [3] Convolutional Deep Neural Networks have been able to perform better than the traditional Deep Neural Networks because of weight sharing. Since in CNN the weights are shared and the pooling layer stores only the important information about the image, this makes it very versatile in scanning the features and characteristics of the image. CNN also makes use of memory more efficiently. According to [4], CNN has the ability to learn different features using different filters rather than learning only those features that are being hand-engineered. These filters learn the pattern of the features in the image for more accurate prediction. It allows us to extract multiple features at every convolution layer [5].

For our project we use the SPIE-AAPM Lung CT Challenge dataset. In the preprocessing part, we illustrate how to separate the lung nodules from the CT scan images. In the beginning we show how to extract the lung region from the body and the air. First, we use the radiodensity value to differentiate the body part. Then we use the method of pixel labeling techniques, explained in detail in the preprocessing step, to remove the air region. Next, we generate two different masks, one with fill lung area operation and another without fill lung area operation, subtract them, and obtain a new mask with only nodules. Finally, we multiply the original image with the nodule mask extracted so far, and retain only the nodules in the original image.

After performing all these extensive procedures, next we crop 100 slices from the DICOM image file, rescale and resize each slice to a size of 100 x 100. Then we construct a 3D CNN model and feed 60% of the preprocess data to train it. On evaluating the model's performance, we obtain testing accuracy of 100% and precision, recall, kappa and F-score value of 1.

II. LITERATURE REVIEW

Convolutional neural network (ConvNet/CNN), Deep Learning or other machine learning algorithms have been used by various researchers to perform experiments on different types of lung cancer detection. Taruna Aggarwal et al [6][7] classified the lung nodules from CT scanned images using Linear Discriminant Analysis (LDA). While pre-processing, they used median filter to eliminate the irregularities from the input images and obtain the smoothing effect. Optimal threshold technique is used for lung image segmentation and GLCM (Gray Level Co-Occurrence Matrix) for calculating the statistical features. Finally, the success rate was found to be 84%. M. Bikromjit Khumancha et al [8] used CNN to detect lung cancer using CT scans. The dataset is used from the Data Science Bowl 2017 Kaggle competition [17]. In the first step, they detected the annotated nodules. A cube of size 32 X 32 X 32 is made around the nodules, with the nodule in the center. An ROI (Region of Interest) mask is applied for lungs. Then cubes are made around the predicted nodules and the prediction is done using a second 3D CNN. The precision achieved in the model is found to be 89.24%.

P. Mohamed Shakeel et al [9] used the Cancer Imaging Archive (CIA) dataset for their work. The noise is removed

using weighted mean histogram equalization approach and segmentation is done using Improved Profuse Clustering Technique (IPCT). For prediction of cancer they use deep learning instantaneously trained neural networks. The system ensures 98.42% accuracy. Rahul Paul et al [10] used pre-trained CNN and applied three techniques, namely warping, cropping and sliding window methods to take out the deep features from the CT images. By combining traditional features along with deep features and applying a decision tree classifier with a leave one out cross validation and symmetric uncertainty feature ranking method, their best accuracy reached the level of 82.50%.

Prajwal et al [11] used a CanNet architecture, wherein two convolution layers are placed after the input layer and a pooling layer is followed by a dropout layer and a fully connected layer. The maximum testing accuracy using CanNet architecture is found to be 76.00%. Goran Jakimovski et al [12] used double CNN for lung cancer stage detection. The images are classified into piles of same slice images using the K-means algorithm. An in-depth search is done using a double CDNN followed by max pooling. After training the CDNN, images from different Tx stages were given input for testing the Deep Neural Network to check the stage at which the algorithm detects the cancer. After being analyzed by medical personnel, the outcoming results were satisfactory. Jinsa Kuruvilla et al [13] worked on lung cancer classification using neural networks. As a feature for classification, the statistical features are used. As compared to other parameters, skewness gave the maximum classification accuracy. After using the training function the classification accuracy was found to be 91.11%. Two more training functions were proposed with the best accuracy of 93.30%. T. W. Way et al [14][15] used diagnosis of pulmonary nodules by 3D active contours. They used parameters such as morphological and texture features followed by Linear Discriminant classifier with the results giving the area under ROC curve (Az) as 0.85. Y. Khwata et al [16] used curvature index as the parameter followed by Linear Discriminant Classifier for lung nodule classification. The best accuracy achieved was Az=0.97.

III. Methods

A. *Convolutional Neural Networks*

Currently, convolutional neural network (ConvNet/CNN) is widely used in visual recognition system. A CNN is designed in such a way that it adaptively learns the key features of different patterns through the application of relevant filters. As we go deeper into the CNN layers, the model extracts the higher-level features of the input.

CNN consists of many layers. The convolution layer does feature extraction from the images using different filters. The convolution filter moves over the input in two directions, that is x and y, and performs convolve operation and extracts the features from the image which then gives output (weighted sum), which is the feature space, in a low dimensional 2D matrix. Then the activation function is used. ReLU (Rectified Linear Unit) is one such activation function. It is linear for all values greater than a certain threshold value and zero for all the values less than the threshold value. After this, the max pooling layer is used where the maximum pixel value from the window area is calculated. The output from the last pooling layer is flattened and then fed into a fully connected layer. Another activation function, which is softmax, is used at the last layer of the fully connected layer. The output from the softmax activation function gives us the probability distributions. It maps all the extracted features from the image and maps into final output [18].

B. *3D ConvNet*

3D CNN is used when we need to consider volumetric context. In 3D convolutional neural network, everything is similar to a simple convolutional neural network except that in 3D CNN a 3 dimensional filter moves throughout the dataset in 3 directions (x, y and z - axis). In 3D convolution, a 3D kernel is convolved to a cube by adding multiple contiguous frames one after another [19]. The filters extracts the features important for more precise prediction from the dataset, and gives output in another 3 dimensional space.

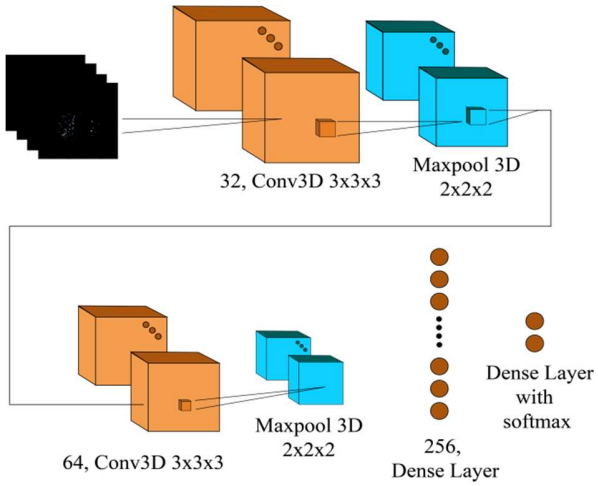

Fig. 1. 3D CNN architecture.

C. *Image Acquisition*

The dataset used for training our CNN is the SPIE-AAPM Lung CT Challenge [20][21][22][23]. The dataset contained studies of 70 patients divided into training and testing. From the training dataset, we have taken 60%, 20% and 20% for training, validation and testing respectively.

D. *Pre-processing*

The dataset used for Cancer detection consists of CT scans, which are in the Digital Imaging and Communications in Medicine (DICOM) format. Each patient has more than 200 to 400 DICOM files, constituting a CT scan.

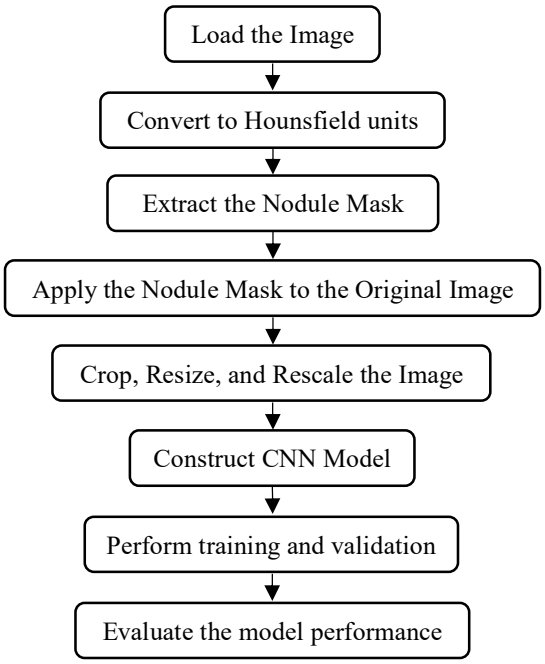

Fig. 2. Proposed method for lung cancer nodule detection.

We have used various metadata of the DICOM files such as slice thickness, pixel spacing, rescale slope, rescale intercept to preprocess the data. The first step in the preprocessing section is to build a Lung Nodule mask and segment the lung nodules from the CT scans which are described from step 1 to 8. From step 9 onwards, we perform the following operations- we crop 100 slices as per the nodule center image location mentioned in the CSV, rescale the image in a uniform spacing of (1, 1, 1) and resize each slice in a size of 100 x 100.

1) Conversion to Hounsfield Unit (HU): We extract the pixel array from the DICOM files of the patient and sort them according to the Instance Number mentioned in the metadata of the DICOM files. Then we convert them to Hounsfield Unit (HU) with the help of Rescale Slope and Rescale Intercept values obtained from the DICOM's metadata (equation 1).

$$HU = (P * S) + I \tag{1}$$

where,

P ~ Pixel Value
S ~ Rescale Slope
I ~ Rescale Intercept
HU ~ Hounsfield Units

2) Threshold the image: The CT scans have different attenuations of Hounsfield Units (HU), a measure of radiodensity, for various substances. The lung portion has a specific radiodensity measure of -500 HU. We have found that the threshold of -420 HU fits perfectly for the lung segmentation in our case. Fig. 3 is the original image whereas Fig. 4 is the mask in which 1 is assigned to those pixels that have radiodensity less than or equal to -420 HU and 2's otherwise (equation 2).

$$Mask = \begin{cases} 2, & if\ HU > -420 \\ 1, & otherwise \end{cases} \tag{2}$$

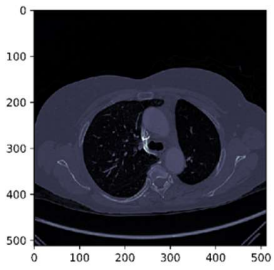

Fig. 3. Original Image.

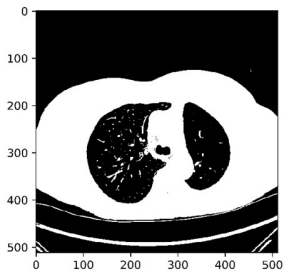

Fig. 4. Threshold Image Slice.

3) Removing the air pixel locations above the patient tray: The adjacent pixels which have the same pixel values are called as connected region. We have labeled each connected region (except for the regions whose pixel values are 0) with a different value. Then we have assigned 2 to those pixels whose label values are equal to the label value of the top left corner (i.e., in the coordinates (0,0,0)) of the mask. The tray where the patient lie during the CT scan, separates the air above the body, and thus the pixel representing the air above the tray gets a different label value, whereas the pixel representing the air below the tray gets a different label value. So, after the above operation, the pixel representing the air above the patient is set to the value 2 whereas the one below the patient remains at 1 as shown in Fig. 5.

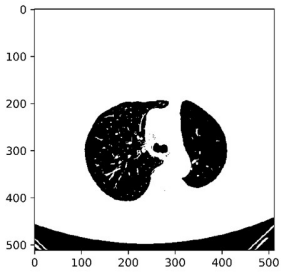

Fig. 5. Removed Upper Air.

4) Fill the lung area: Here we first make a copy of mask, then take each slice from the secondary copied mask and convert their pixel values into binary, i.e., 0's and 1's, by subtracting 1 from each pixel value. Now we again perform the labeling process that we have mentioned earlier. Then we count the number of connected pixels having the same label values and find those having the maximum count, which in this case will be the white region surrounding the lungs as shown in Fig. 5. Now, excluding those pixels having the maximum similar label count, we set 1 to all the pixel values of the original mask generated from the previous step. Thus, in the process, we have assigned the value 1 to all the nodule pixels

in the mask, which have 2's assigned previously. At the end of this process the lung area will be filled with the same pixel value, i.e., 1 in this case, as seen in Fig. 6.

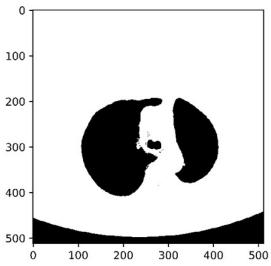

Fig. 6. Filled Lung Slice.

5) Removing air pockets and air below the tray: We now subtract 1 from every pixel value, thus converting the mask into binary. The binary mask is then inverted. This inverted binary mask is labeled again and then we again count the number of connected pixels having the same label values and find those having the maximum count, which in this case will be the lung region. Now, excluding those pixels having the maximum similar label count, we set 0 pixel value to the remaining pixels. This operation helps in removing the air pockets and the air below the tray, shown in Fig 7. If we do not perform the fill lung operation and directly perform the air removal operation as mentioned above we get a mask which is as shown in Fig 8.

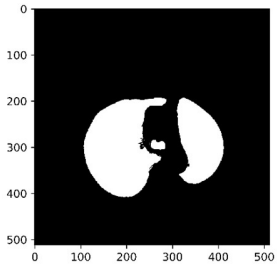

Fig. 7. Removed lower air with lung fill operation.

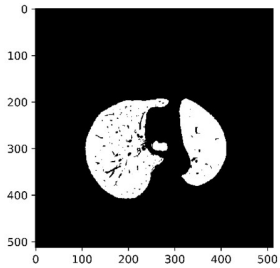

Fig. 8. Removed lower air without lung fill operation.

6) Obtaining Nodule Mask: Now we have generated two different masks, one with fill lung area operation and another without fill lung area operation as mentioned above. We, then, subtract the filled lung mask with the other mask, resulting in a new mask with only nodules as shown in Fig. 9.

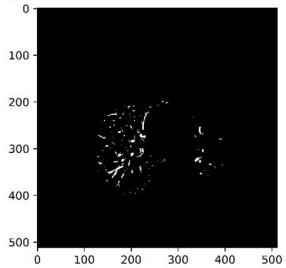

Fig. 9. A nodule mask slice.

7) Applying Nodule Mask to the Original Image: After the extraction of the nodules by performing the above operations, we multiply the original image (Fig. 3) with the nodule mask extracted so far (Fig. 9). This operation will retain only the nodules in the original image. A slice from the resultant image is shown in Fig. 10.

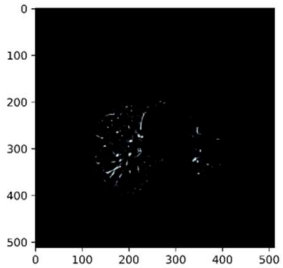

Fig. 10. A slice of nodule image.

8) Cropping the Nodule Image: There are different numbers of slices, ranging from 200 – 400, for different patients. Therefore, to fit them into the CNN model, we have to first make the slice count uniform for all the patients. Thus, to make the slice count uniform, we have cropped 100 slices from the DICOM image file, using the nodule image centre location mentioned in the annotation of the dataset. We have taken 50 slices to the left of the nodule image centre location and 50 slices to the right of the nodule image centre location, making a total count of 100 slices.

9) Rescaling the Nodule Image: Every slice in the nodule image is rescaled to a uniform spacing of (1, 1, 1) using the slice thickness and the pixel spacing values available in the metadata of the DICOM files by Spline Interpolation method. Therefore, we bring similarity in the resolution of all the nodule images and removing diversity of scanner resolution.

10) Resizing the Nodule Image: After performing all the above operations, we finally have the nodule images with a uniform size of 100 x 512 x 512. But, training with such a large-sized image would require very high computational power. So, to reduce the complexity, we resize each of the slices in the nodule image from 512 x 512 to 100 x 100 using bilinear interpolation. Thus, in this way, we have converted the nodule image into a size of 100 x 100 x 100.

E. Experiment

1) CNN Architecture: We created a 3D CNN model with two convolutional layers and two fully connected layers as shown in Fig 11. In each convolutional layer, 32 filters are used of size 3 x 3 x 3 which is followed by Relu activation function and Max pooling layer. In the first fully connected layer, 256 nodes are used and in the last layer one node is included.

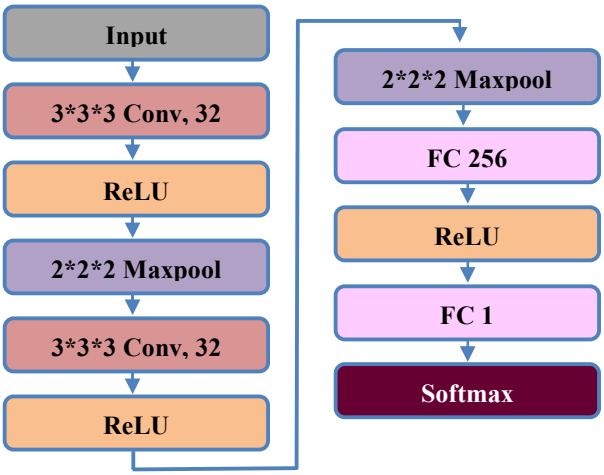

Fig. 11. CNN architecture.

2) Training Algorithm: For training the proposed CNN model, we have applied a widely used optimizer known as Adam optimizer, designed for training an extensive range of deep learning architectures. It is also known as the Adaptive Moment Estimation Algorithm, which is a combination of two optimizers, namely, RMSprop and gradient descent with momentum. In brief, the optimizer calculates an exponentially weighted average of the past gradients Vdw and Vdb, as shown in equations 3 and 4. The update rule for Vdw and Vdb is similar to that of gradient descent with momentum. On the other hand, the rule of updating Sdw and Sdb is like RMSprop (equation 3 and 4).

$$V_{dw} = ß_1 \times V_{dw} + (1 - ß_1) \times dW \quad (3)$$

$$V_{db} = ß_1 \times V_{db} + (1 - ß_1) \times db \quad (4)$$

$$S_{dw} = ß_2 \times S_{dw} + (1 - ß_2) \times dW^2 \quad (5)$$

$$S_{dw} = ß_2 \times S_{dw} + (1 - ß_2) \times db^2 \quad (6)$$

$$V_{dw}^{corrected} = V_{dw} / (1 - ß_1) \quad (7)$$

$$V_{dw}^{corrected} = V_{db} / (1 - ß_1) \quad (8)$$

$$S_{dw}^{corrected} = S_{dw} / (1 - ß_2) \quad (9)$$

$$S_{db}^{corrected} = S_{db} / (1 - ß_2) \quad (10)$$

Further $V_{dw}^{corrected}$, $V_{db}^{corrected}$, $S_{dw}^{corrected}$, and $S_{db}^{corrected}$ are estimated using equations 7,8,9, and 10. Finally, the parameters, i.e., the weights and the biases are updated using $V_{dw}^{corrected}$, $V_{db}^{corrected}$, $S_{dw}^{corrected}$, and $S_{db}^{corrected}$ as shown in equations 11 and 12

where,

$ß_1$, $ß_2$ ~ Hyperparameters

ε ~ Epsilon

$$Weight = Weight - learning\ rate \times V_{dw}^{corrected} / \sqrt{(S_{dw}^{corrected} + \varepsilon)} \quad (11)$$

$$Bias = Bias - learning\ rate \times V_{db}^{corrected} / \sqrt{(S_{db}^{corrected} + \varepsilon)} \quad (12)$$

3) Hyperparameters: Hyperparameters play an essential role in regulating the performance of a model. Proper tuning of the hyperparameters can contribute enormously to optimize a model. In our model, we have tuned some hyperparameters such as learning rate, number of epochs, etc, as shown in Table 1. We have set the parameters ß1, ß2, and ε used in the Adam optimizer algorithm (equations 3 to 10) to their default value of 0.9, 0.999, and 10-8, respectively. Epsilon (ε), is a minimal value used to prevent the update rule for weights and the bias from dividing by 0 (equations 11 & 12). Also, the learning rate that is used in the optimizer for training the model is 0.001. Lastly, we have accomplished the training of the model by using 30 number of epochs with a batch size of 2 per epoch.

TABLE 1
HYPERPARAMETERS VALUES

Hyperparameters	Values
ß1	0.9
ß2	0.999
Epsilon (ε)	10^{-8}
Batch Size	2
Learning Rate	0.001
Number of Epochs	30

IV. RESULT

We have used various python libraries such as 'pydicom' (library to work with DICOM Files), 'scikit-image' (an image processing library), and 'OpenCV' (an open-source, real-time computer vision library) to preprocess the dataset before fitting into the model. We have then split the preprocessed dataset into 60%, 20%, and 20% as training, validation, and testing respectively. Using the TensorFlow Library and the Keras API, we have trained the model by using the training dataset. We have trained the model for 30 epochs in a system having Intel i7-4790k 4th Generation Processor, 16 GB of RAM, and a platform of Windows 10 architecture.

Metrics Evaluation: After completion of the training process, we have found that the model yields 83.33% as the training accuracy. We further assess the model performance by examining various metrics such as Testing accuracy, Precision, Recall, Kappa-Score, and F-score.

Accuracy is a measure of how many instances that are correctly classified out of the total number of instances. The

proposed model provides us an excellent testing accuracy of 100%. The reason behind obtaining 100% accuracy might be due to the small dataset having a smaller number of instances. Working with a larger dataset might affect the accuracy of the model.

Precision shows how many of the total selected instances are correctly classified, whereas recall shows how many correctly classified instances are selected. The computed precision and recall of our model are depicted in Table 2.

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{\mathrm{TP} + \mathrm{FN}} \tag{14}$$

TABLE 2 HYPERPARAMETERS VALUES

Classes	Precision	Recall
Benign	1	1
Malignant	1	1

Kappa score (equation 15), is the measure of agreement between two raters. In our case, predicted values and the truth values are the two raters. Using Observed Accuracy and Predicted Accuracy, we have computed the Kappa score of the model and obtained an attenuation of 1.

$$Kappa\ Score(K) = \frac{p_o - p_e}{1 - p_e} \tag{15}$$

The harmonic mean of Precision and Recall is known as F-score or F-measure. We obtained a F-score of 1 using equation 16, which signifies perfect Precision and Recall of the model.

$$F - Score = 2 \times \frac{Precision\ \times Recall}{Precision + Recall} \tag{16}$$

V. CONCLUSION

Here we have attempted to detect lung cancer using 3D CNN on CT scan images. We used the SPIE-AAPM Lung CT Challenge dataset. By utilizing the various metadata of the DICOM files such as slice thickness, pixel spacing, rescale slope and rescale intercept, we preprocessed the data and segmented the lung nodules from the CT scans. Then we cropped 100 slices from the DICOM image file, rescaled it and resized the slices to a size of (100 x 100). After training our 3D CNN model we examine the model using various metrics such as testing accuracy, precision, recall, Kappa-Score, and F-score. Our model gives an excellent testing accuracy of 100% and precision, recall, kappa and F-score value of 1. The SPIE-AAPM Lung CT Challenge dataset is a very small dataset of about 12 GB. The encouraging results obtained here is very motivating for us to try the 3D CNN based model for lung cancer detection on larger datasets.

ACKNOWLEDGMENT

This work is supported by Assam Science and Technical University, under TEQIP III program of Ministry of Human Resource Development, India, funded by World Bank.

REFERENCES

[1] "Lung Cancer 101," Lung Cancer 101 | Lungcancer.org. [Online]. Available: https://www.lungcancer.org/find_information/publications/163-lung_cancer_101/265-what_is_lung_cancer. [Accessed: 14-Mar-2020].

[2] World Health Organization. International Agency for Research on Cancer Incidence, Mortality and Prevalence Worldwide http://gco.iarc.fr/today/data/factsheets/cancers/15-Lung-fact-sheet.pdf , 2018

[3] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[5] P. Rao, N. A. Pereira and R. Srinivasan, "Convolutional neural networks for lung cancer screening in computed tomography (CT) scans," 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, 2016, pp. 489-493.

[6] Aggarwal T, Furqan A, Kalra K. Feature extraction and LDA based classification of lung nodules in chest CT scan images. 2015 Int Conf Adv Comput Commun Informatics, ICACCI 2015; 2015;1189–93.

[7] T. Aggarwal, A. Furqan and K. Kalra, "Feature extraction and LDA based classification of lung nodules in chest CT scan images," 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, 2015, pp. 1189-1193.

[8] M. B. Khumancha, A. Barai and C. B. R. Rao, "Lung Cancer Detection from Computed Tomography (CT) Scans using Convolutional Neural Network," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 2019, pp. 1-7.

[9] P. Shakeel, M. A. Burhanuddin, and M. Desa, "Lung Cancer Detection from CT Image Using Improved Profuse Clustering and Deep Learning Instantaneously Trained Neural Networks," Measurement, vol. 145, May 2019, doi: 10.1016/j.measurement.2019.05.027.

[10] R. Paul, S. H. Hawkins, L. O. Hall, D. B. Goldgof and R. J. Gillies, "Combining deep neural network and traditional image features to improve survival prediction accuracy for lung cancer patients from diagnostic CT," 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, 2016, pp. 002570-002575.

[11] P. Rao, N. A. Pereira and R. Srinivasan, "Convolutional neural networks for lung cancer screening in computed tomography (CT) scans," 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, 2016, pp. 489-493.

[12] G Jakimovski and D. Davcev, "Using Double Convolution Neural Network for Lung Cancer Stage Detection," Appl. Sci., vol. 9, p. 427, Jan. 2019, doi: 10.3390/app9030427.

[13] J. Kuruvilla and K. Gunavathi, "Lung cancer classification using neural networks for CT images," Comput. Methods Programs Biomed., vol. 113, Oct. 2013, doi: 10.1016/j.cmpb.2013.10.011

[14] T. W. Way et al., "Computer-aided diagnosis of pulmonary nodules on CT scans: segmentation and classification using 3D active contours," Med. Phys., vol. 33, no. 7, pp. 2323–2337, Jul. 2006, doi: 10.1118/1.2207129.

[15] T. W. Way et al., “Effect of CT scanning parameters on volumetric measurements of pulmonary nodules by 3D active contour segmentation: a phantom study,” Phys. Med. Biol., vol. 53, no. 5, pp. 1295–1312, Mar. 2008, doi: 10.1088/0031-9155/53/5/009.

[16] Y. Kawata et al., "Computerized analysis of 3-D pulmonary nodule images in surrounding and internal structure feature spaces," Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205), Thessaloniki, Greece, 2001, pp. 889-892 vol.2.

[17] Kaggle, Data Science Bowl 2017.” https://www.kaggle.com/c/data-science-bowl-2017”

[18] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: an overview and application in radiology,” Insights Imaging, vol. 9, no. 4, pp. 611–629, Aug. 2018, doi: 10.1007/s13244-018-0639-9.

[19] S. Ji, W. Xu, M. Yang and K. Yu, "3D Convolutional Neural Networks for Human Action Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 1, pp. 221-231, Jan. 2013.

[20] S. G. Armato 3rd et al., “LUNGx Challenge for computerized lung nodule classification,” J. Med. Imaging Bellingham Wash, vol. 3, no. 4, pp. 044506–044506, Oct. 2016, doi: 10.1117/1.JMI.3.4.044506.

[21] S. G. Armato 3rd et al., “LUNGx Challenge for computerized lung nodule classification: reflections and lessons learned,” J. Med. Imaging Bellingham Wash, vol. 2, no. 2, pp. 020103–020103, Apr. 2015, doi: 10.1117/1.JMI.2.2.020103.

[22] S. G. Armato 3rd et al., “LUNGx Challenge for computerized lung nodule classification: reflections and lessons learned,” J. Med. Imaging Bellingham Wash, vol. 2, no. 2, pp. 020103–020103, Apr. 2015, doi: 10.1117/1.JMI.2.2.020103.

[23] K. Clark et al., “The Cancer Imaging Archive (TCIA): maintaining and operating a public information repository,” J. Digit. Imaging, vol. 26, no. 6, pp. 1045–1057, Dec. 2013, doi: 10.1007/s10278-013-9622-7.