



# A Fast Automatic Juxta-pleural Lung Nodule Detection Framework Using Convolutional Neural Networks and Vote Algorithm

Jiaxing Tan<sup>1</sup>, Yumei Huo<sup>1(✉)</sup>, Zhengrong Liang<sup>2</sup>, and Lihong Li<sup>1</sup>

<sup>1</sup> City University of New York, New York City, NY, USA  
jtan@gradcenter.cuny.edu, {Yumei.Huo,Lihong.Li}@csi.cuny.edu

<sup>2</sup> Stony Brook University, Stony Brook, NY, USA  
Jerome.Liang@sunysb.edu

**Abstract.** Lung Nodule Detection from CT scans is a crucial task for the early detection of lung cancer with high difficulty performing an automatic detection. In this paper, we propose a fast automatic voting based framework using Convolutional Neural Network to detect juxta-pleural nodules, which are pulmonary (lung) nodules attached to the chest wall and hard to detect even by human experts. The detection result for each region in the CT scan is voted by the detection results of the extracted candidates from the region, which we formulate as a generative model. We perform two sets of experiments: one is to validate our framework, and the other is to compare different convolution neural network settings under our framework. The result shows our framework is competent to detect juxta-pleural lung nodules especially when only a weak classifier trained on noisy data is available. Meanwhile, we overcome the problem of determining the proper input size for nodules with high variance in diameters.

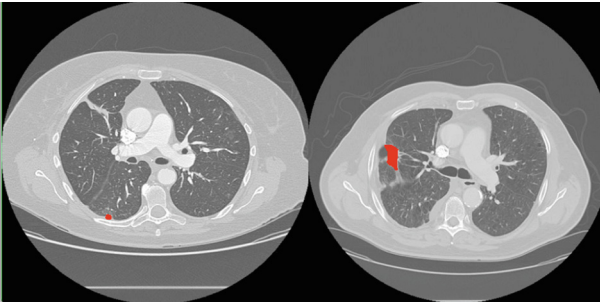
**Keywords:** Lung cancer · Juxta-pleural nodule detection  
Deep learning · Weakly labeled

## 1 Introduction

Lung cancer is the leading cause of cancer-related deaths in the United States [1]. Automated lung cancer detection is of great importance. In general, based on their locations, lung nodules can be classified into two types. One is juxta-pleural, which is typically attached to the lung wall; and the other one is the isolated type within the lung area. Compared with isolated nodules, juxta-pleural nodules are more difficult to detect because their intensities and texture are similar to those of the chest wall and they usually have relatively small radius. As a result, the traditional methods such as region growing [2] and active contour model [3],

usually fail in the classification of juxta-pleural nodules. Some examples of juxta-pleural nodules are shown in Fig. 1, where the red part is the labeled nodule by the radiologist.

Recently, Convolutional Neural Network (ConvNet) has achieved a great success in computer vision, especially in image classification with a fast updating accuracy score on the ImageNet Challenge since 2012 [4,5]. In contrast to the traditional feature-engineering, where machines use human hand-crafted features to learn, ConvNet is designed to have the machine learn features from data itself without human involvement. This makes the machine learning task more efficient as less pre-processing is needed. The success of ConvNet has cast some light on the area of bio-medical. Only in the area of bio-medical image scan analysis, ConvNet has already been applied to solve the problems such as organ segmentation [6,7], and lung nodule diagnosis [8]. Recently, Suzuki et al. [9] applied ConvNet for lung nodule detection and compared several different ConvNet designs.



**Fig. 1.** Examples of juxta-pleural nodule from our dataset. The red color indicating the location and shape of nodules (Color figure online)

Traditionally, nodule detection pipeline requires a series of preprocessing, such as lung segmentation, vessel elimination, suspect candidates extraction and classification. On the other side, for a ConvNet based solution, it is tricky to find a proper input patch size to contain the whole nodules as it has high variance in diameter. Small nodules might be easily overlooked if the patch size is too small.

In this paper, we design a fast automatic voting based framework using Convolutional Neural Network to detect juxta-pleural nodules from raw CT scan. For each CT slice, we first divide the CT image into regions, where each region could be viewed as a bag of candidates. Then instead of throwing the whole region into ConvNet, we extract several candidates from each region and apply a voting algorithm to decide whether a nodule exists in that region or not. In addition, we compare our ConvNet with two ConvNet structures which have the highest AUC from [9] in terms of the performance on our juxta-pleural dataset. We perform two sets of experiments: one is to validate our framework and the other is to compare different ConvNet designs under our framework. Our experimental results show that the framework is efficient and our ConvNet structure

outperforms the ones from [9] especially when only weakly labeled data with noise is available for training. Our voting algorithm could improve the original model by a large margin.

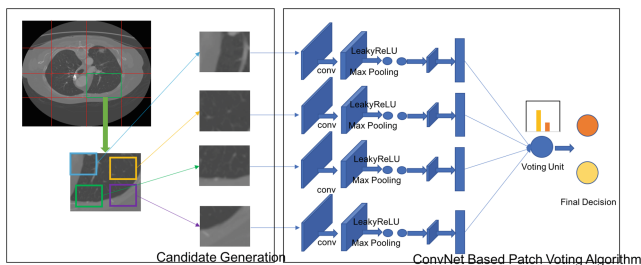
There are three major strengths of our propose framework. First, our model does not need any pre-processings on the raw data. Secondly, with the design of patch-based voting framework, we eliminate the problem of window size selection as well as enhance model performance. At last, our voting framework could significantly improve original ConvNet model’s performance and could be generalized to different kinds of ConvNets.

## 2 Methodology

We designed a bag-of-voting-candidates (BOVC) model to perform nodule detection. For a CT scan  $R$ , we assume it contains  $H$  regions, denoted as  $R = R_1, R_2, \dots, R_H$ , where each region has size  $M \times N$  and independent and identically distributed (i.i.d). We view each region  $R_i$  as a bag of  $K$  voting candidates  $R_i = C_{i,1}, C_{i,2}, \dots, C_{i,K}$ , where  $C_{i,j} = \langle x_{i,j}, y_{i,j} \rangle$  is the  $j$ th candidate of region  $i$  containing a data patch  $x_{i,j}$  with its corresponding label  $y_{i,j}$ . We further assume that each candidate is independently generated by a hidden variable  $\phi_i$ , indicating the class distribution of region  $i$ . The total probability of this generative model could be written as:

$$P(R; \phi) = \prod_{i=1}^H P(R_i | \phi_i) P(\phi_i) = \prod_{i=1}^H \prod_{j=1}^K P(C_{i,j} | \phi_i) P(\phi_i) \quad (1)$$

The major workflow of our framework is shown in Fig. 2. There are two steps, the first step is to generate candidate given a certain region, the second step is to generate detection result with our ConvNet based patch voting algorithm. We will introduce both steps accordingly in the rest of this section.



**Fig. 2.** Workflow of our framework, the first step is candidate extraction and the second step is our ConvNet based voting algorithm.

## 2.1 Candidate Extraction

We formulate our candidate generating algorithm as follow: For each region, take region  $i$  as an example, we extract  $k_1$  candidates with size  $T \times T$  ( $T < \min(M, N)$ ) from region  $i$ , where  $k_1$  is set to make sure there are the least overlap to cover the whole region. Then we design to randomly extract  $k_2$  more candidates to get more random votes. At last, we have  $K$  candidates in total. Then we perform translation on each candidate, which can further enrich dataset as well as provide more possible views of the candidate, as a result, several image patches (which will also be candidates) can be generated from a single original candidate.

## 2.2 Parameter Estimation with ConvNet

We design our parameter estimation algorithm with ConvNets. The two-step algorithm is described as follow:

**Step1:** We update the ConvNet parameter  $\theta$  with regard to the following equation:

$$\theta = \arg \max_{\theta} P(R|\phi; \theta) = \arg \max_{\theta} P(R|\theta) = \arg \max_{\theta} \prod_{x_{i,j} \in R} P(x_{i,j}, y_i | \theta) \quad (2)$$

**Step 2:** Given  $\theta$ ,  $\phi_i$  is estimated by:

$$\phi_{ij} = \frac{N_{ij}}{N_i}, \quad (3)$$

where  $N_{ij}$  is the count of candidates in region  $i$  predicted with label  $j$  by ConvNet and  $N_i$  is the total number of candidates in region  $i$ .

For our task, there are only two classes (nodule or non-nodule contained), denoted as positive and negative. Our vote algorithm generates the final decision for region  $i$  based on  $\phi_i$  with the following rule:

$$Result(Region_i) = \begin{cases} Positive \ Ratio > \ Threshold \\ Negative \ Ratio < \ Threshold \end{cases} \quad (4)$$

The threshold in the equation is a predefined empirical value, which is the 15 percentile of the validation dataset and the ratio of region  $i$  is calculated as:

$$Ratio_i = \frac{\phi_{i,Positive}}{\phi_{i,Positive} + \phi_{i,Negative}} \quad (5)$$

**ConvNet Design.** The design of the ConvNet contains input layer, convolution layer, subsampling layer and fully connected layer. We design a ConvNet of input size  $64 \times 64$  with two convolutional layers, each followed by a max-pooling layer. For the convolutional layer, we choose Leaky Rectified Linear Unit (LeakyReLU) [10] as the activation function:

$$\text{LeakyReLU}(x, \alpha) = \max(x, 0) + \alpha \times \min(x, 0),$$

where  $\alpha$  is a small user pre-defined non-zero gradient negative slope that is set to negative values.

In Table 1, we detailed our ConvNet design with two ConvNet designs with the highest AUC in [9] as a comparison, which is denoted as sh-CNN and rd-CNN. The last column shows how many different kernels are used for a convolution layer or how many neurons are used in a fully connected layer. We are not specifying the input layer in the table since all ConvNets take input size of  $64 \times 64$ .

**Table 1.** Table of Different ConvNet Designs, here we compare three different designs, our proposed network, the one with the best performance from [9] and a very shallow model as comparisons.

Layer	Type	Kernel No.	Layer	Type	Kernel No.
(1) Our Model			(3) rd-CNN		
1	Convolution	$7 \times 7$ 32	1	Convolution	$11 \times 11$ 32
2	Max-pooling	$2 \times 2$ -	2	Max-pooling	$2 \times 2$ -
3	Convolution	$5 \times 5$ 64	3	Convolution	$5 \times 5$ 32
4	Max-pooling	$2 \times 2$ -	4	Max-pooling	$2 \times 2$ -
5	Fully connected	- 128	5	Convolution	$3 \times 3$ 32
6	Fully connected	- 2	6	Max-pooling	$2 \times 2$ -
(2) sh-CNN			7	Convolution	$3 \times 3$ 64
1	Convolution	$11 \times 11$ 32	8	Max-pooling	$2 \times 2$ -
2	Max-pooling	$2 \times 2$ -	9	Fully connected	- 128
3	Fully connected	- 128	10	Fully connected	- 2
4	Fully connected	- 2			

A softmax fully-connected layer is used as the last layer to generate the probability distribution over two classes (Nodule and Non-Nodule). For the training of our ConvNet, cross-entropy Loss is used to minimize the difference between detected class and real class from groundtruth with L2 Norm regularization added. Note that in order to reuse training data, we design weight sharing among the separate ConvNet paths to make better use of the training data.

### 3 Experiments

The goal of the experiments is two-fold. One is to validate our framework, and the other is to compare different ConvNet designs under our framework.

### 3.1 Dataset

The original RAW CT data is acquired from the largest public database founded by the Lung Image Database Consortium and Image Database Resource Initiative (LIDC-IDRI). Each CT slice has a size of  $512 \times 512$ . Our radiologist labels the position of the nodule for each given CT slice.

We select 90 patients from our dataset with at least one juxta-pleural nodules included and has more than 12 slices containing nodules on average. By randomly sampling patches around nodule areas and non-nodule areas respectively, we obtain the positive samples and negative samples. And the rotation is applied as a translation method to each sample. In each patient’s CT scan, the ratio of the nodule to non-nodule areas is now not balanced. However, to train a binary classifier, we have to use a balanced dataset, which means that there should be equal numbers of positive and negative samples. With regard to this point, we perform the last step to balance our training dataset.

### 3.2 Experiment Design

In our experiments, we choose the region size to be  $128 \times 128$  and use rotation as the translation method. We extract  $k = 9$  candidates from each region, which are patches located at 4 corners, 4 middle part on each edge, and 1 in the center. Apparently, there are some overlaps among the candidates. However, our candidate extraction method is more efficient than using a sliding window to cover the whole region. Each candidate will then be rotated 3 times.

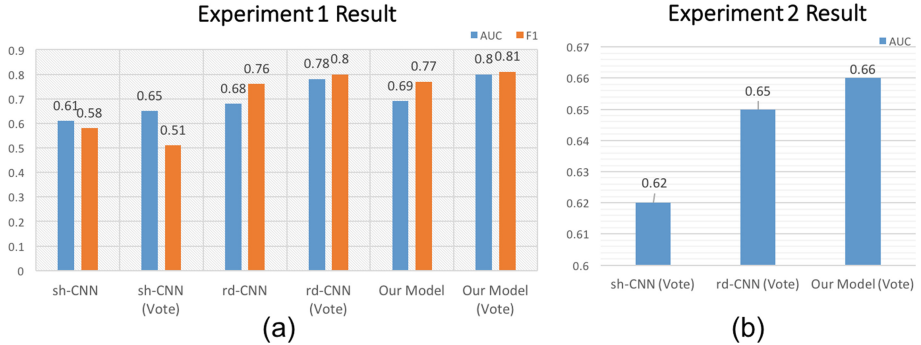
We use Theano to implement ConvNet designs. For training, we apply Adam [11] optimizer with a batch size of 40. The learning rate is set to  $10^{-4}$ , momentum to 0.9, and weight decay to 0.0005. The network is initialized with a Gaussian Distribution.

We designed two experiments to validate our framework. In the first experiment, we compare three models with and without voting algorithm to validate our framework on a test dataset containing 20 patients with a balanced number of positive and negative regions. The model without voting will consider each patch as a distinct input. And the voting result is achieved by our proposed model. We use AUC and F1 score as evaluation metrics for experiment 1. However, in practice, nodule detection is performed on highly unbalanced data. As a result, we designed second experiment to test our framework performance on 10 patients’ CT slices containing nodules without balancing the dataset and use AUC for evaluation.

### 3.3 Experiment Result

The result of experiment 1 is shown in Fig. 3(a). It shows that vote algorithm has improved AUC for all ConvNet. And our model has a better performance than both rd-CNN and sh-CNN in both scenarios when the voting algorithm is used and not used respectively. Different from [9], sh-CNN has the worst performance on our dataset in both scenarios. Some typical highly confusing patches are

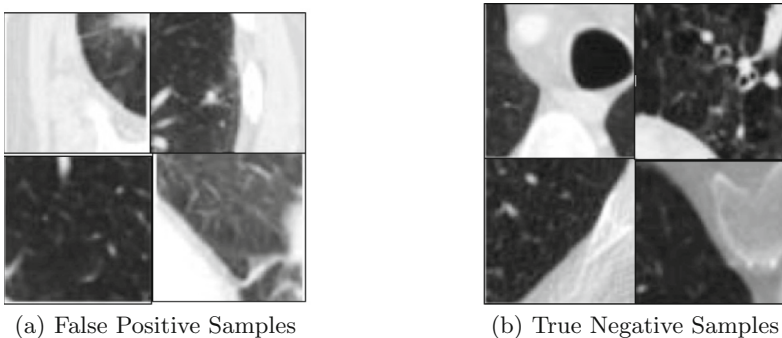
shown in figure Fig. 4. We can see that most False Positive ones are caused by including the chest wall or some other tissues into the sampled patches, which increase the noise of data samples. On the other hand, the small size (radius) of a nodule is a major reason for True Negative ones.



**Fig. 3.** Experiment result, (a) Performance comparison results in experiment 1 and (b) Performance comparison results in experiment 2.

In experiment 2, as the real detection has unbalanced data for two classes (*positive : neagtive*  $> 1 : 20$ ), the performance is lower than that of experiment 1. The results are shown in Fig. 3(b). We can see that our model works the best, which is slightly better than rd-CNN, while sh-CNN has the lowest AUC.

In conclusion, our experiments showed that our automatic detection framework, which is based on ConvNet, can detect juxta-pleural lung nodule from CT scan of a patient efficiently. Especially when the effect of a detection classifier is limited by noisy training data, our vote algorithm could be used to enhance its performance.



**Fig. 4.** Typical highly confusing samples in experiment 1

## 4 Conclusion and Future Work

In this paper, we propose a framework to detect juxta-pleural nodule from CT scans based on ConvNet using vote algorithm. We compared different ConvNet structures in our framework and examined the effectiveness of our framework on LIDC-IDRI juxta-pleural lung nodule datasets. Experiments show that our framework is competent at detecting juxta-pleural nodules. On the other side, our experiments show that the incorrectly classified data samples are those containing the chestwall or some noisy, so some preprocessing methods could be used to filter out those “bad” samples. A possible extension could be training different ConvNet with image patch randomly split into different groups. For our future work, besides the above mentioned continuing research, we will also try other ConvNet structures to enhance the accuracy of the classifier and design an efficient framework to locate the nodules from CT scans.

## References

1. American Cancer Society: Key statistics for lung cancer (2016)
2. Adams, R., Bischof, L.: Seeded region growing. *IEEE Trans. Pattern Anal. Mach. Intell.* **16**(6), 641–647 (1994)
3. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: active contour models. *Int. J. Comput. Vis.* **1**(4), 321–331 (1988)
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105 (2012)
5. Szegedy, C., et al.: Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9 (2015)
6. Li, R., Zeng, T., Peng, H., Ji, S.: Deep learning segmentation of optical microscopy images improves 3-D neuron reconstruction. *IEEE Trans. Med. Imaging* **36**(7), 1533–1541 (2017)
7. Cha, K.H., et al.: Urinary bladder segmentation in ct urography using deep-learning convolutional neural network and level sets. *Med. Phys.* **43**(4), 1882–1896 (2016)
8. Kumar, D., Wong, A., Clausi, D.A.: Lung nodule classification using deep features in ct images. In: *2015 12th Conference on Computer and Robot Vision (CRV)*, pp. 133–138. IEEE (2015)
9. Tajbakhsh, N., Suzuki, K.: Comparing two classes of end-to-end machine-learning models in lung nodule detection and classification: MTANNS vs. CNNS. *Pattern Recognit.* **63**, 476–486 (2017)
10. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)