Binary Classification for Lung Nodule Based on Channel Attention Mechanism

Khai Dinh Lai[1,2,3], Thai Hoang Le[1,2(✉)], and Thuy Thanh Nguyen[4]

[1] Faculty of Information Technology, University of Science, Ho Chi Minh City 70000, Vietnam
laidinhkhai@sgu.edu.vn, lhthai@fit.hcmus.edu.vn
[2] Vietnam National University, Ho Chi Minh City 70000, Vietnam
[3] Faculty of Information Technology, Saigon University, Ho Chi Minh City 70000, Vietnam
[4] VNU University of Technology, Ha Noi City, Vietnam
nguyenthanhthuy@vnu.edu.vn

Abstract. In order to effectively handle the problem of tumor detection on the LUNA16 dataset, we present a new methodology for data augmentation to address the issue of imbalance between the number of positive and negative candidates in this study. Furthermore, a new deep learning model - ASS (a model that combines Convnet sub-attention with Softmax loss) is also proposed and evaluated on patches with different sizes of the LUNA16. Data enrichment techniques are implemented in two ways: off-line augmentation increases the number of images based on the image under consideration, and on-line augmentation increases the number of images by rotating the image at four angles (0°, 90°, 180°, and 270°). We build candidate boxes of various sizes based on the coordinates of each candidate, and these candidate boxes are used to demonstrate the usefulness of the suggested ASS model. The results of cross-testing (with four cases: case 1, ASS trained and tested on a dataset of size 50 × 50; case 2, using ASS trained on a dataset of size 50 × 50 to test a dataset of size 100 × 100; case 3, ASS trained and tested on a dataset of size 100 × 100 and case 4, using ASS trained on a dataset of size 100 × 100 to test a dataset of size 50 × 50) show that the proposed ASS model is feasible.

Keywords: Attention convolutional network · Nodules detection · SE block

1 Introduction

Lung cancer is a form of cancer that occurs in the lungs, often known as a malignant tumor of the respiratory tract [1]. The condition develops when a malignant tumor starts in the lung and rapidly grows in size, causing invasion and pressing on surrounding organs. Furthermore, benign lung tumors can still occur in some circumstances. Benign tumors vary fundamentally from malignant tumors (cancerous cells). However, proper scientific diagnostic methods from doctors and medical professionals are still required to precisely detect the nature of the tumor. Lung cancer progresses via four stages, the first of which is the detection of lung lesions via general examination or CT scan. The earliest symptoms of lung cancer, however, are quite similar to those of common

P. Cong Vinh and A. Rakib (Eds.): ICCASA 2021, LNICST 409, pp. 205–218, 2021.
https://doi.org/10.1007/978-3-030-93179-7_16

pneumonia and flu, and the features on small CT scans are not obvious, which can easily confuse clinicians [2]. Over the last two decades, computer-aided diagnosis (CAD) systems have been proposed as a viable solution [3]. These systems have strongly developed high performance as well as diagnostic accuracy. Especially, groundbreaking researches when there is the birth of deep learning networks, typically convolutional deep learning networks (CNNs) [4]. The convolutional deep learning network has handled image processing and classification challenges with excellent results, such as medical image analysis and emotional face classification [5]. CAD's primary goals are candidate screening and false positive reduction. These two tasks correspond to the two challenges set by LUNA to research groups in order to develop a full lung nodule detection system based on data provided by LIDC/IDRI [6–8]. Problem 1: Pulmonary nodule detection is a candidate detection stage, which is designed to simplify lung imaging. That is, ignoring the unimportant components such as the lung wall, keeping only the element with great potential is called the nodule candidate. Problem 2: false positive reduction. This aims to remove false positive candidates from the candidate group collected during the candidate detection step. Therefore, the false-positive reduction process needs to be equipped with a classifier with strong discriminant ability to separate nodule and non-nodule objects. In this study, we focus on the second problem of reducing false positives, or in other words, the binary classification problem: nodule class and non-nodule class. Many researches have carried out this work with good outcomes; we would like to refer only a few works that use the CNN methodology on 2D or 3D models.

2D Models

The DIAG CONVNET model, developed by Setio and colleagues [9], is a multi - view convolutional network that incorporates three node detectors for small, medium, and large notes (depending on diameter). For each option, the scientists collected nine 65×65 patches of size 50×50 pixels from distinct viewpoints corresponding to distinct planes of symmetry in the cuboid. These images are processed using a stream of 2D convolutional networks comprised of three integration layers and a max-pooling layer. The sizes of the convolutional layers are as follows: 24 kernels of size 5×5; 32 kernels of size 3×3; and 48 kernels of size 3×3.

3D Models

Dou and colleagues [10] use the multi-context 3D convolutional network concept called CUMedVis as the foundation to build 3 deep learning network architectures (Archi-1, Archi-2, Archi-3) to solve challenges due to differences in node size, shape and geometrical features. Each network uses a different input image to incorporate different levels of contextual information around lung nodules. In the final stage, these three architectures are merged with a weighted linear combination to obtain the final classification result for the candidates.

Ding [11] proposed a pulmonary lymph node detection system based on CNNs. Their system involves applying region-based CNNs to detect nodules on image slices and using 3-D CNNs to reduce the false positive rate. The method was evaluated on the LUNA16 dataset and achieved a high sensitivity (94.4%). The authors have taken advantage of VGG 16 and used the proposed network to achieve high performance, but the cost of building the proposed network by region is not small.

Rushil Anirudh et al. [12] present a tumor search strategy based on Convolution neural networks (CNN), with pre-labeled dataset, using 5 Convolution and ReLU convolution layers, 2 max-pooling and 2 softmax layers, 2 fully connected (FC) layers for classification of 1 × 1 kernels. The author offers a tumor notation model in which the doctor just needs to make notes about the nodule's center and size. Then, they can use CNN to pinpoint the tumor's location with high accuracy, even if the labels are inaccurate.

We found that most of the research groups performed the combination of multiple models on original size CT images. The results are appreciated but the training process is quite expensive. In addition, the authors have not exploited the specificity of CT images to have appropriate weights. Therefore, we approach the method of dividing the image into small patches around the candidates, and at the same time apply the Attention mechanism to keep the important features and ignore the less important features.

Our main contributions to this study can be summarized as follows:

(1) Data preparation: (a) Propose technique for creating the candidate box on CT images; (b) Propose a data enrichment technique for CT images containing tumors in the training data set with the aim of solving the problem of imbalanced data.
(2) Propose an image classification system, with or without a nodule, that is integrated with a deep learning network model and an attention mechanism.

2 Background

2.1 Attention Mechanism with SE Block

Jie Hu proposes the Squeeze-and-Excitation block (SE Block) to improve the quality of representations of complex neural networks [13]. This SE block consists of two main steps. In the "Squeeze" step, the author uses the global mean to aggregate feature maps of the size H × W to create a channel descriptor. In the "excitation" step, the author uses fully-connected layers corresponding to the output neurons after the "squeeze" step to give weights to each channel. This weight will be used for feature maps to select the important feature that is also the final result of the SE block (Fig. 1).

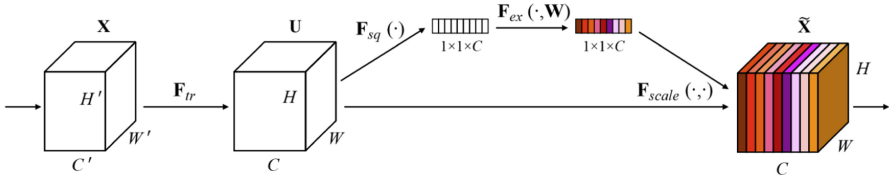


Fig. 1. Squeeze-and-Excitation (SE Block)

The implementation of the SE block is detailed below. We assume that the input image X is of size H′ × W′ × C′, where H′ and W′ are the height and width of the

image, respectively, and C′ is the number of channels. F_{tr} is the convolutional multiplication X with the filters V = [v_1, v_2, …, v_C] where v_c is the parameter value corresponding to the Cth kernel to generate the feature map $U \in \mathbb{R}^{H \times W \times C}$. So for C filter we have $U = [u_1, u_2, \ldots, u_c]$ với

$$u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x^s \tag{1}$$

In the "squeeze" step, we aggregate feature maps across their spatial dimensions H × W to create a channel descriptor using global average pooling. With the C feature map, we have a vector $z \in \mathbb{R}^C$, with the cth component of z determined using a formula:

$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j) \tag{2}$$

In the "excitation" step, two fully-connected layers are used to get vector s via the formula:

$$s = F_{ex}(z, W) = \sigma(W_2 RELU(W_1 z)) \tag{3}$$

where $W_1 \in \mathbb{R}^{\frac{c}{r} \times C}$ và $W_2 \in \mathbb{R}^{C \times \frac{c}{r}}$ is the weights of the fully-connected layers, The parameter r is defined as the "reduction ratio." The smaller the intermediate representations are, the higher the r-value. σ is the sigmoid function to map the component values of s to the range [0, 1]. In this case, s is a gate with an Attention mechanism, Specifically, each component s_i corresponds to a channel U_i. If the value s_i is close to 1 then the U_i corresponding channel is significant and vice versa. In the last step, the output of the SE block is performed by multiplied the feature map U by the vector s.

$$\hat{X} = s \times U \tag{4}$$

By this multiplier, less important channels will be ignored, while other critical channels will be retained.

2.2 Revisiting ASS Model

a. Attention Sub-Convnet

The squeeze-and-excitation block acts as a plugin that may be added to any CNN network architecture. In our research, we just apply the Attention mechanism to build us a distinct subnet namely Attention sub-Convnet [14]. Specifically, an Attention sub-convnet is made up of many components. The first is that the convolutional layer accepts an image of size H × W with initial channels C′. Then, use convolutional multiplication with a Kernel of size K × K to get feature maps of size H × W with C channels. Second, the average pooling layer averages the values of the channels' components on each feature map of size H × W and form a vector z of length C. Third,

the fully-connected layer on the z vector produces vector s with C components. Next, the sigmoid function is used to normalize the components of the vector s to the corresponding values of the value domain [0; 1]. Finally, the Attention sub-convnet network performs vector multiplication s with feature maps of size H × W × C where C is the number of channels to get the output of feature maps with selected channels (Table 1).

Table 1. An overview of the Attention sub-Convnet architecture included: layers, input size, output size and parameters

Layer	Input size	Output size	Parameter
Input	H × W × C′		
Convolution	H × W × C′	H × W × C	Kernel K × K
Average Pooling	H × W × C	1 × 1 × C	Kernel H × W
Fully connected	1 × 1 × C	1 × 1 × C	
Sigmoid	1 × 1 × C	1 × 1 × C	
Multiply	H × W × C, 1 × 1 × C	H × W × C	
Output		H × W × C	

b. ASS Model

In [14], we integrates Attention sub-convnet networks into what we call ASS. ASS combines 3 Attention sub-convnet networks interspersed with max pooling layers to reduce feature map size. At the flatten layer we get a vector of size 10816. This vector is passed through two fully-connected layers with output sizes of 512 and 2 respectively. Finally, to determine if a candidate is a nodule or not, we apply the softmax-loss function for binary classification.

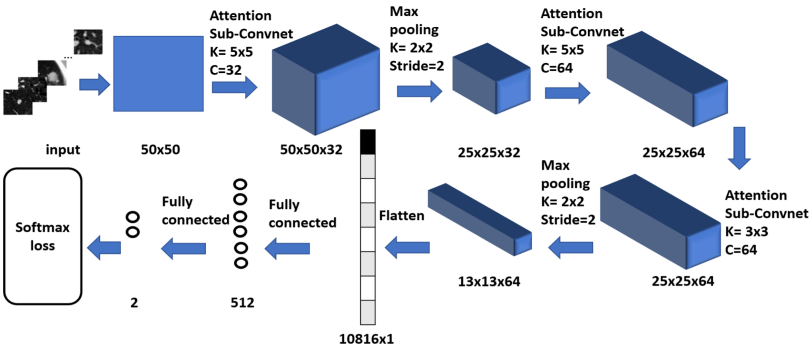


Fig. 2. Architecture of ASS model in [14]

3 Proposal Nodule Detection System

We propose a lung nodule classification system that integrates the ASS model shown in Fig. 3.

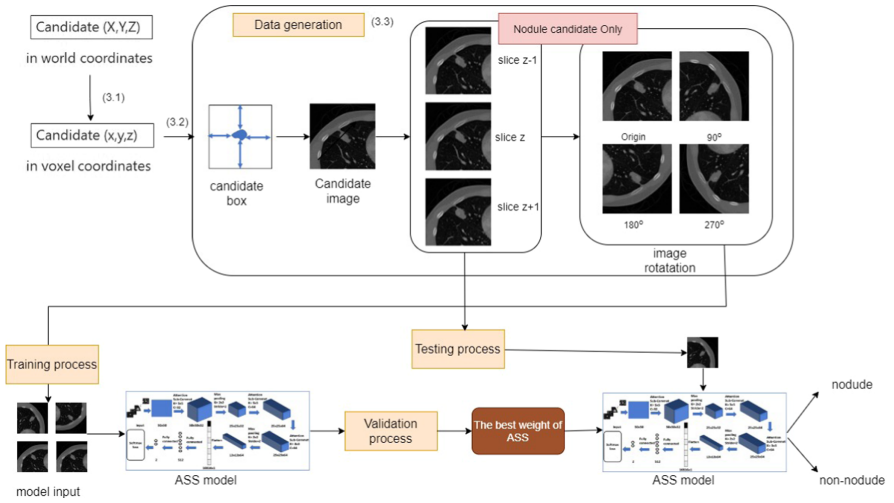


Fig. 3. Proposal nodule detection system with (3.1, 3.2, 3.3) will be explained in next section.

3.1 Convert World Coordinates into Voxel Coordinates

For the first challenge, LUNA encourages research groups to build systems for identifying lung nodules, and the results of this challenge are collected by LUNA in an excel file candidate_v2. In this file, each nodule candidate is placed on a row with data about the nodule's X, Y, and Z coordinates in the real world coordinate system and labeled data as 0 or 1 corresponding to the tumor or non-tumor classification. Therefore, in step 1, we convert the candidate coordinates in the real world coordinate system to the voxel coordinate system. (see in Fig. 4).

```
def worldToVoxelCoord(worldCoord, origin, spacing):
    stretchedVoxelCoord = np.absolute(worldCoord - origin)
    voxelCoord = stretchedVoxelCoord / spacing
    return voxelCoord
```

Fig. 4. Pseudocode for converting world coordinate system to voxel coordinate system

While the worldCoord parameter is an array of candidate values received from the file candidate_v2, origin parameter is the origin representing the position of the first voxel (0, 0, 0) in the anatomical coordinate system, and spacing is the distance that defines the distance between the voxels along each axis (units in mm). These parameters are read from the mhd file in the LUNA16 database.

3.2 Candidate Box

Once the candidates appear in the voxel coordinate system, we perform candidate framing by considering the candidate coordinates to be the center of a square with sides of the desired size. From this box, we get an image containing the candidate whose size is equal to the box size with the formula (x ± size/2, y ± size/2, z_0). For example, a candidate with coordinates (x_0, y_0, z_0) and the box is 100 × 100 pixels, we determine the height and width of the candidate box by using the formula (x ± 50, y ± 50, z_0). Figure 2 shows pseudocode that captures the frame surrounding a candidate. From the candidate envelope, we crop the image to the box size and get the image containing the candidate (Figs. 5 and 6).

```
def crop_patch(img_file, out_file, X, Y, size=100):
    # read image from file
    img = cv2.imread(img_file)
    H, W = img.shape[:2] # get image height and width

    # get coordinates for cropping patch
    i_min, i_max = max(Y-size, 0), min(Y+size+1, H)
    j_min, j_max = max(X-size, 0), min(X+size+1, W)
    patch = img[i_min:i_max, j_min:j_max]

    # resize patch to expected size
    patch = cv2.resize(patch,(size, size))

    # save patch to output file
    cv2.imwrite(out_file, patch)
```

Fig. 5. Pseudocode for creating a candidate box.

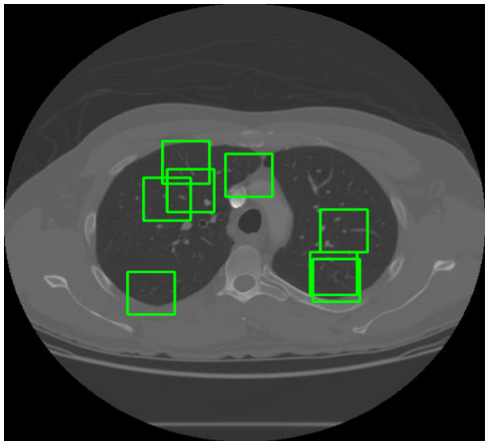


Fig. 6. A slice with the candidate boxes

3.3 Data Augmentation

The number of positive candidates in the LUNA16 database is quite modest in comparison to the number of negative candidates. Therefore, in order to reduce the impact of data imbalance during the learning process, we augment the data in two ways based on the positive candidates.

Off-line Data Augmentation
As shown in Sect. 3.2, from a candidate box we take an image sized according to the size of the box. The LUNA16 dataset is another format derived from LIDC_IDRI collections, which have all cases with slice thickness <2 mm [6]. This ensures that the appearance of the nodule with a diameter >3 mm is represented by three contiguous slices [15]. As a result, we can take two more patches, one each from the front and rear slices that contain the candidate under consideration and are labeled as positive candidates. Thus, for a positive candidate whose coordinates (x_0, y_0, z_0), a candidate box is determined by the formula ($x \pm size/2$, $y \pm size/2$, z_0), the formula ($x \pm size/2$, $y \pm size/2$, z_0 -1); ($x \pm size/2$, $y \pm size/2$, $z_0 + 1$) finds the two more candidate boxes to make 2 patches with positive labels (Fig. 7).

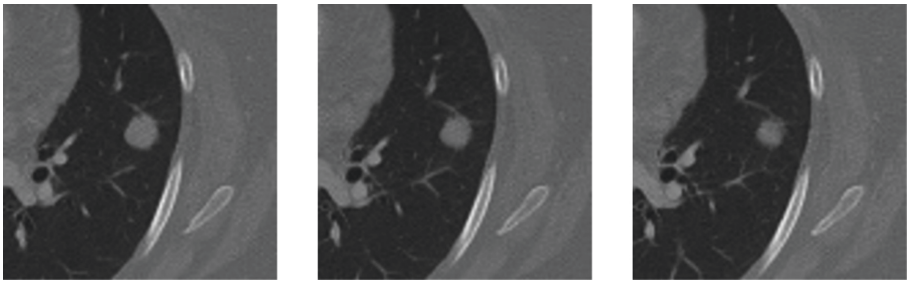


Fig. 7. Three patches in three consecutive slices with minor differences

On-line Data Augmentation
Based on the analysis of how the doctor places the CT image when observing, we discover that the doctor can orient the image horizontally or vertically and that even if the image is accidentally inverted, the doctor can still determine whether the image has a nodule or not. Aside from placing the image in a standard view, the doctor can also place it in the 90°, 180° or 270° right rotation. Therefore, we take advantage of this feature to enrich the data during training. When an image is supplied, we select one of three the right rotations 90°, 180°, 270° at random, or we preserve the original form (0° rotation). We guarantee that from an initial image, we will generate three additional images with three separate rotations at each learning time without saving any images on our device (Fig. 8).

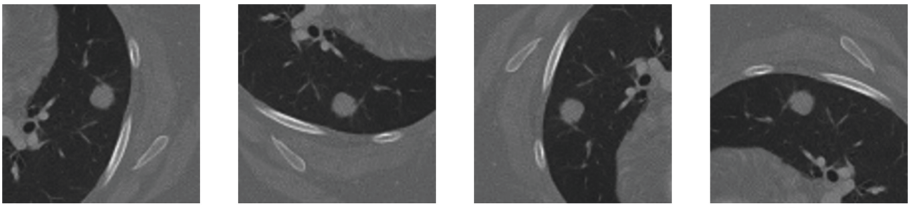


Fig. 8. The origin image and its three different rotation

4 Experimental Results

4.1 Datasets

From the dataset LUNA16 and the file candidate v2, which contains candidate coordinates and class labels. We utilize an algorithm to define the candidate box and off-line data augmentation to produce two databases with varying image sizes, which we call data50 and data100 matching image data with dimensions of 50 × 50 pixels and 100 × 100 pixels. These two sets of images were created in collaboration with a number of picked candidates differed only in the size of their candidate box. These two sets have the same number of images. The ratio of positive to negative groups is 20:80 (Table 2).

Table 2. The number of candidates were classed as positive or negative.

Dataset type	Positive samples	Negative samples
Train	1200	4800
Val	400	1600
Test	400	1600

4.2 ASS_w50 and ASS_w100

We undertake sequential training on two data sets, data50 and data100, using the ASS model provided in Sect. 2.2, and get two different sets of weights. As a result, we termed the model ASS_w50 after the ASS model that was trained on data50. The ASS model trained on data100 is represented by the ASS_w100 model and we obtain the weight ASS_w100.

4.3 Environment for Experiments

Models are built in the same computing environment. Setting up a set of weights based on a Gaussian distribution with a mean of zero, a standard deviation of 0.02, and a bias of zero. The gradient-based optimization technique of stochastic objective functions, Adam was selected based on adaptive lower-order moments estimations. We have a learning rate of 1e−4 and a weight decay rate of 1e−6. We run roughly 50 epochs for each scenario, with a batch size of 8 images. For training process, we used an early stopping strategy. This not only saves us time training after the performance converged, but also helps us prevent overfitting.

4.4 Evaluation

We used the Holdout method to evaluate models. To get objective results, we split the data into three independent directories with ratio between training set, validating set and testing set, respectively 60%, 20% and 20%.

When we performed the test step, we identified that the outcomes were divided into four groups in the confusion matrix. They were, in order, true positive (TP), false positive (FP), true negative (TN), and false negative (FN). Precision, recall, specificity, and AUC were the measures we used in our comparative analysis [16].

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \tag{5}$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \tag{6}$$

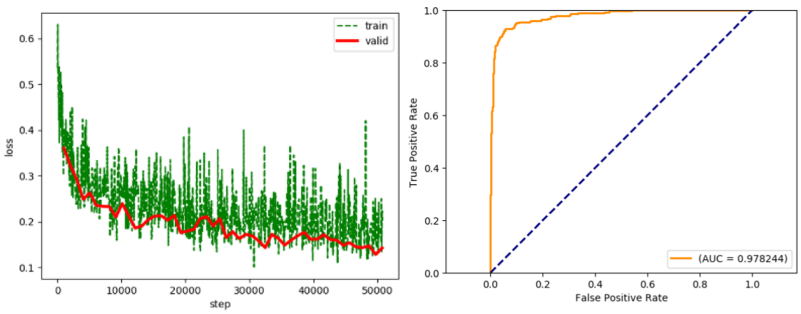
The specificity is the measurement which is distinctive in that it represents the percent of correctly defined negatives. Properly defined as the proportion of segmented slices free of nodule cancer.

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

5 Experimental Results

Case 1: ASS_w50 performs training, evaluating and testing on data50.

In this case, we use ASS model to train and validate on data50 to get w50 weight set (Fig. 9 and Table 3).



(a) (b)

Fig. 9. ASS_w50 performs training, evaluating and testing on data50. (a) the value of loss parameter during training and validating. (b) the ROC curve of case 1.

Table 3. Displays results for Precision, Recall, Specificity of ASS_50 on data50

Model	Precision	Recall	Specificity
ASS_50/Data50	0.917	0.840	0.984

Case 2: ASS_w50 performs a cross-testing on data100.

In this scenario, we employ ASS trained on data50 to cross-test against data100 (Fig. 10 and Table 4).

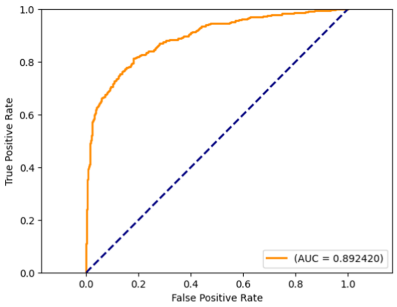


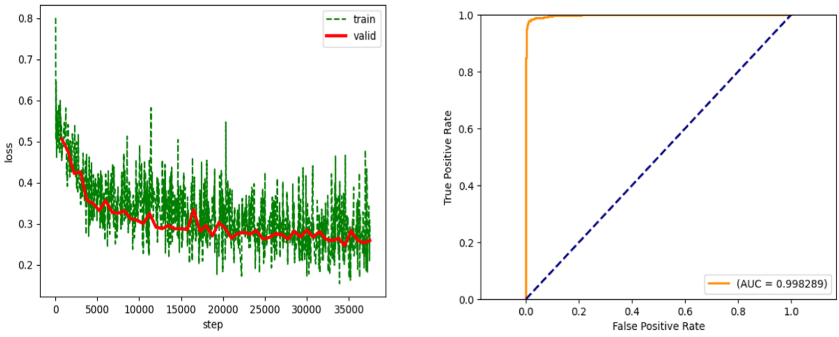
Fig. 10. The diagrams show the ROC curve of case 2.

Table 4. Displays results for Precision, Recall, Specificity of ASS_50 on data100

Model/Data	Precision	Recall	Specificity
ASS_50/Data100	0.853061	0.522500	0.977500

Case 3: ASS_w100 performs training, evaluating and testing on data100.

The same as in case 1, but we use ASS model to train and validate on data100 to get ASS_w100 weight set (Fig. 11 and Table 5).



(a) (b)

Fig. 11. ASS_w100 performs training, evaluating and testing on data100. (a) the value of loss parameter during training and validating. (b) the ROC curve of case 4.

Table 5. Displays results for Precision, Recall, Specificity of ASS_100 on data100

Model	Precision	Recall	Specificity
ASS_100/Data100	0.962500	0.996250	0.973451

Case 4: ASS_w100 performs a cross-testing on data50.

In this time, we train the ASS on data100 and use this model with its weight which is obtained to conduct to test on data50 (Fig. 12 and Table 6).

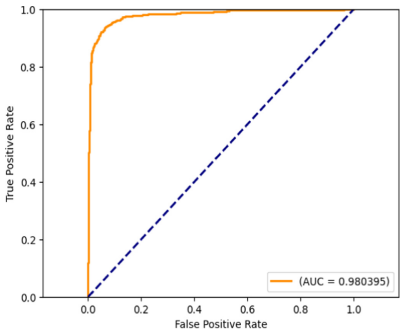


Fig. 12. The diagrams show the ROC curve of case 4.

Table 6. Displays results for Precision, Recall, Specificity of ASS_100 on data50

Model/Data	Precision	Recall	Specificity
ASS_100/Data50	0.929539	0.857500	0.983750

We make some observations, with two cases having similarity in the size of the data during training, evaluating and testing (case 1 and 3), the model achieves positive results, such as using ASS for training and testing on data50. When using a set of weights trained from a set of 50 × 50 size, testing data with size of 100 × 100 gives lower results than the rest of the cases. However, the specificity result of case 2 is still above 90%. In the opposite case, ASS with w100 weight set to test on data50 produces acceptable results. Consequently, it demonstrates that the proposed ASS model is effective when tested on various LUNA16 data sets.

6 Conclusions

In this paper, we offer a new data augmentation strategy to deal with the problem of imbalanced data in the training process. This method is based on the characteristic of lung nodules screening with computed tomography and the doctor's visual angle while observing the CT scan. Furthermore, we present a method for generating candidate

boxes based on candidate coordinates in order to generate patches of varying sizes (50×50, 100×100). The techniques of augmenting and building candidate box allow us to assess the usefulness and feasibility of the ASS model presented in [14] on datasets of various sizes. The data enrichment strategy is not restricted in this study; it may also be used to produce rich data sources in imbalanced data problems.

Acknowledgements. This research is funded by University of Science, VNU-HCM under grant number CNTT 2021-09.

References

1. Bray, F., Ferlay, J., Soerjomataram, I., Siegel, R.L., Torre, L.A., Jemal, A.: Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer J. Clin. **68**(6), 394–424 (2018)
2. Grippi, M.A., et al.: Fishman's Pulmonary Diseases and Disorders, 5th edn. McGraw-Hill, New York (2015)
3. Ignatious, S., Joseph, R.: Computer aided lung cancer detection system. In: Global Conference on Communication Technologies (GCCT), Thuckalay, pp. 555–558 (2015)
4. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
5. Vo, D.M., Le, T.H.: Deep generic features and SVM for facial expression recognition. In: 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), Danang, pp. 80–84 (2016). https://doi.org/10.1109/NICS.2016.7725672
6. Armato III, S.G., et al.: Data from LIDC-IDRI (2015). https://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX
7. LIDC-IDRI. https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI
8. LuNa. https://luna16.grand-challenge.org/
9. Setio, A.A.A., et al.: Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. IEEE Trans. Med. Imaging **35**(5), 1160–1169 (2016)
10. Dou, Q., Chen, H., Yu, L., Qin, J., Heng, P.A.: Multilevel contextual 3-D CNNs for false positive reduction in pulmonary nodule detection. IEEE Trans. Bio-medical Eng. **64**(7), 1558–1567 (2017). https://doi.org/10.1109/TBME.2016.2613502. https://ieeexplore.ieee.org/document/7576695. Print ISSN 0018-9294. Electronic ISSN 1558-2531
11. Ding, J., Li, A., Hu, Z., Wang, L.: Accurate pulmonary nodule detection in computed tomography images using deep convolutional neural networks. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10435, pp. 559–567. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66179-7_64
12. Anirudh, R., Thiagarajan, J.J., Bremer, T., Kim, H.: Lung nodule detection using 3D convolutional neural networks trained on weakly labeled data. In: SPIE Medical Imaging (2016). https://doi.org/10.1117/12.2214876
13. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-excitation networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7132–7141 (2018)
14. Lai, K.D., Cao, T.M., Thai, N.H., Le, T.H.: The combination of attention sub-convnet and triplet loss for pulmonary nodule detection in CT images. In: Miraz, M.H., Excell, P.S., Ware, A., Soomro, S., Ali, M. (eds.) iCETiC 2020. LNICSSITE, vol. 332, pp. 227–238. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60036-5_16

15. Brown, M.S., Lo, P., Goldin, J.G., et al.: Correction to: toward clinically usable CAD for lung cancer screening with computed tomography. Eur. Radiol. **30**, 1822 (2020). https://doi.org/10.1007/s00330-019-06512-1
16. Branco, P., Torgo, L., Ribeiro, R.: A survey of predictive modeling on imbalanced domains. ACM Comput. Surv. **49**(2), Article no. 31 (2016)