

# Enhancing Pulmonary Nodule Classification performance with TriCaps-RL: A Capsule Network Reinforcement Learning Approach

Guang-Yuan Zheng

College of Information Technology  
Shanghai Jian Qiao University, Shanghai 201306, China  
School of Mathematics and Computer Science  
Yanan University, Shaanxi 716000, China  
3257662@qq.com

Fu-Quan Zhang

School of Computer and Data Science, Minjiang University  
No.200 Xiyuangong Road, Fuzhou University Town, Fuzhou 350108, China

Digital Media Art, Key Laboratory of Sichuan Province  
Sichuan Conservatory of Music, Chengdu 610021, China

Fuzhou Technology Innovation Center of intelligent Manufacturing information System  
Minjiang University, Fuzhou 350108, China

Engineering Research Center for ICH Digitalization and Multi-source Information Fusion  
(Fujian Polytechnic Normal University)  
Fujian Province University, Fuzhou 350300, China  
8528750@qq.com

Chen Cheng

College of Business  
Shanghai Jian Qiao University, Shanghai 201306, China  
13050@gench.edu.cn

Hong-Tao Shan

College of Electronic and Electrical Engineering  
Shanghai University of Engineering Science, Shanghai 201620, China  
shanhongtao@sues.edu.cn

Nouman Qadeer Soomro

Software Department  
Mehran University of Engineering & Technology, Sindh 76062, Pakistan  
noumansoomro@gmail.com

Yi Liu\*

School of Mathematics and Computer Science  
Yanan University, Shaanxi 716000, China  
ly@yau.edu.cn

\*Corresponding author: Yi Liu

Received October 23, 2023, revised January 27, 2024, accepted April 6, 2024.

**ABSTRACT.** *Distinguishing the manifestations of pulmonary nodules poses a significant challenge in the medical field, demanding the expertise of experienced radiologists. This complexity results in the high cost and inadequacy of sample labeling. In response, this study introduces a reinforcement learning approach, denoted as TriCaps-RL, for the classification of pulmonary nodules based on CT signs. This approach initially employs the Q-value loss to train a single CapsNet. When the performance of the CapsNet plateaus and further enhancement becomes challenging, the loss function is progressively replaced by a triplet metric to augment performance. Consequently, a more nuanced differentiation among various types of pulmonary nodules is achieved.*

*The proposed method can improve its performance through interactions with radiologists during their CT image reading process, thereby mitigating the shortage of radiologists and addressing the time-consuming issue of training sample labeling in medical image research. This paper delves into the fine-grained classification of pulmonary nodules using deep reinforcement learning based on CT imaging signs. In comparison to previous studies that primarily concentrate on the benign/malignant classification of lung nodules, this fine-grained classification of nodules founded on CT imaging signs proves to be more valuable for medical practitioners in making accurate diagnostic decisions.*

**Keywords:** CT image, lung cancer, pulmonary nodule, CT imaging sign, reinforcement learning

---

**1. Introduction.** In recent years, despite the considerable achievements of artificial intelligence algorithms in numerous research domains [1], a significant scope for expansion still exists in the field of medical image analysis [2, 3]. The classification of pulmonary nodules based on CT images serves as an illustrative example. The visual manifestations of pulmonary nodules are complex and indistinguishable, which makes the labelling process of training samples very difficult and requires experienced senior radiologists. However, a significant scarcity of skilled radiologists exists on a global scale [4]. These reasons lead to expensive and inadequate of labeled training samples.

Reinforcement learning (RL) is a paradigm of machine learning. Its performance can be improved by a correct answer or reward from the environment indicating its good decision. Its parameters are constantly modified to improve the performance by comparing the difference between the periodic achievements and the expectations. In recent years, researchers have paid more and more attention to reinforcement learning. In the field of game and robot control, reinforcement learning has been very successful.

CapsNet is a convincing capsule network proposed by Sabour et al. [5], which uses capsules to store local features of images and their transformed information, such as position and attitude [6]. CapsNet transmits and compresses information through a routing protocol, which can save the information of the relationships between various components of high-level features [7], can achieving knowledge replication across spaces. The features learned by the CapsNet are transformation-equivalent, and robust to affine transformation. CapsNet considers the relative spatial relationship of objects in the image, which can better summarize the perceived content. Therefore, it can better complete the network training with less annotated data, higher recognition accuracy and stronger generalization ability.

Triplet network [8] learns network parameters through feature representation distance metric, which is known as triplet loss function. The purpose of triplet loss metric learning is to reduce or limit the representation distance between samples in one class and increase the distance between different classes through training. The triplet loss metric is suitable for the situations with many categories in the training data set and few samples in each category.

In this paper, a method of lung nodules CT sign classification based on triplet CapsNet is proposed, which is denoted TriCaps-RL (TriCapsNet Reinforcement Learning). This method combines Capsule Networks (CapsNets) with reinforcement learning techniques, aiming to achieve precise nodule classification and improve interaction between radiologists and the automated diagnosis system. The TriCaps-RL method not only holds great potential for enhancing the accuracy of nodule classification but also can offer opportunities for real-time collaboration with medical experts during the diagnosis process. TriCapsNet improves performance by means of reinforcement learning, striving for increased similarity among samples within the same categories and greater dissimilarity between different categories. The agent of TriCaps-Net receives states (image blocks) and feedback corresponding actions (label predictions) according to its learnt strategy, in reinforcement learning process. The method calculates the gap between the rewards and the expected values and updates the agent's parameters according to the gap. In this way, the performance of the method is gradually improved. The framework of the TriCaps-RL method is shown in Figure 1. Firstly, TriCaps-RL uses a single CapsNet as agent for preliminary learning, and then conducts further optimization using triplet CapsNets metric strategy. Through triplet metric learning, the distances between the feature representations of the anchor and the negative samples become larger, and the distances between the anchor and the positive samples become smaller. Finally, the feature representations will reach an appropriate relative distance corresponding to the categories. Thus, different categories of samples can be differentiated more subtly. Now, let's delve into the learning process of the TriCaps-RL method.

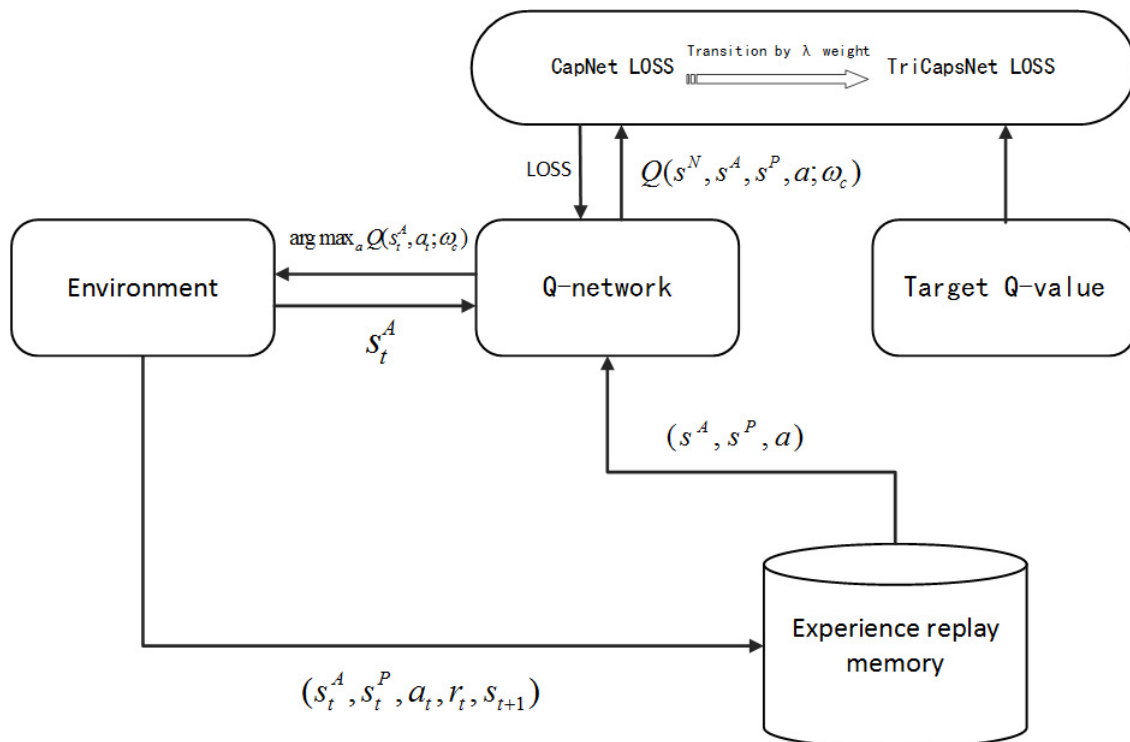


FIGURE 1. Framework of the proposed TriCaps-RL method.

TriCaps-RL can improve its performance by comparing its actions with the diagnosis result of radiologists, which can address the problem of highly expensive labor and time costs for training sample labelling in the field of medical image research.

**2. Related works.** The capsule network uses a new architecture that mimics the human visual system and incorporates CNN's lack of information about relative position, angle and other factors, so that it need less training data and achieve enhanced generalization capability from different perspective [9]. Some scholars have tried CapsNet in the classification of pulmonary nodules.

Mobiny and Nguyen [10] tried to use CapsNet instead of Convolutional Neural Network(CNN) to diagnose pulmonary nodules. The experimental results showed that CapsNet is better than CNN when the number of samples was very large. This advantage was even more obvious when the train samples were small. To improve the efficiency of computation, the authors also changed the original image reconstruction method of CapsNet to reduce the reconstruction error of the network and improve the accuracy of classification. Jiménez-Sánchez et al. [11] conducted an experiment to evaluation the performance of CapsNet and CNN on a medical image data set. The results show that the performance of CapsNet was the same or better than CNN, even with less data for training. Moreover, CapsNet was more robust to the unbalanced distribution of classes. Iesmanta and Alzbutas [12] proposed a deep learning method with convolutional CapsNet to classify images of four kinds of breast tissue biopsies stained with hematoxylin and eosin. In this method, he used a 5-layer convolutional neural network to extract the basic information of the images, then the final images were classified using CapsNet, and good results were obtained. Silva et al. [13] used a set of 3D geometric features combined with a reinforcement learning method to classify benign and malignant pulmonary nodules. The authors selected 36 nodules, including 29 benign and 7 malignant tumor samples, and then experimented with manually extracted sphericity index, convexity index, curvature index, and surface features of the candidate regions. The accuracy of the algorithm was 81%. Ali et al. [14] proposed a lung nodule detection algorithm based on deep reinforcement learning inspired by the AlphaGo method. The agent used a CNN network. The algorithm took the original CT images as input, regarded it as the state set, and output the category attribution of the existence nodule. The authors trained their models on LIDC-IDRI (Lung Image Database Consortium and Image Database Resource Initiative) [15] database and tested them. The overall accuracy was 64.4% (sensitivity 58.9%, specificity 55.3%, PPV 54.2%, and NPV 60%). Reinforcement learning can rely on the constructed reward feedback mechanism for performance improvement without labeling data, so it is applied in many fields [16]. There are also some attempts in the research field of pulmonary nodule diagnosis CAD.

Bhandary et al. [17] put forward a deep learning (DL) framework named Modified AlexNet (MAN) to examine pneumonia and cancer in the lung. Chaunzwa et al. [18] trained and validated convolutional neural networks on a dataset comprising adenocarcinoma (ADC) and Squamous Cell Carcinoma (SCC) nodule CT images of 311 patients. Khan et al. [19] proposed a novel design of contrast stretching based classical features fusion process for localizing the of lungs cancer classification. Wang et al. [20] suggested a novel residual neural network to identify several type of lung cancer via CT images. None of these works were classified by signs.

**3. TriCaps-RL method.** TriCaps-RL can learn from the radiologist's CT image diagnosing process. Its classification performance can be improved by simple interactions with the physician. In this way, the current difficulty of training samples insufficiency in medical image analysis can be alleviated. On the one hand, for a new CT image to be analyzed, the algorithm selects a corresponding action (sign diagnose) and displays it back to the image; on the other hand, the radiologist gives the correct labels to the lesions which are wrong labelled by the algorithm. Afterward, the algorithm compares

the agent's classification outcomes with the radiologist's labeling results and returns a reward for each action based on the diagnostic consistency between the two. By comparing the expected values and the method's rewards, the agent improves its classification performance by modifying its parameters step by step.

The input of TriCaps-RL includes three image blocks in the second learning stage, which are anchor image ( $A$ ), positive sample image ( $P$ ), and negative sample image ( $N$ ). The triplet loss function can be described by Euclidean distance function:

$$L(A, P, N) = \max(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha, 0) \quad (1)$$

Where  $\alpha$  is the distance between positive and negative sample pairs and  $f(\cdot)$  is the feature expression function.

The triplet loss metric can capture the relative similarity between samples, making the distance between the feature representations of the anchor and the positive images as small as possible and the distance between the anchor and the negative sample images as large as possible, through continuous learning. Finally, the gap between the two representations satisfies at least one minimum interval, an  $\alpha$ . This procedure can be interpreted in Figure 2. At the beginning, although the anchor sample and the positive sample belong to the same category, not the same category as the negative one, the anchor image may be closer to the negative sample in feature representation, as shown in Figure 2(a). After training is completed, the distance between the anchor and positive representation becomes closer than that between anchor and negative sample, as shown in Figure 2(b). In TriCaps-RL,

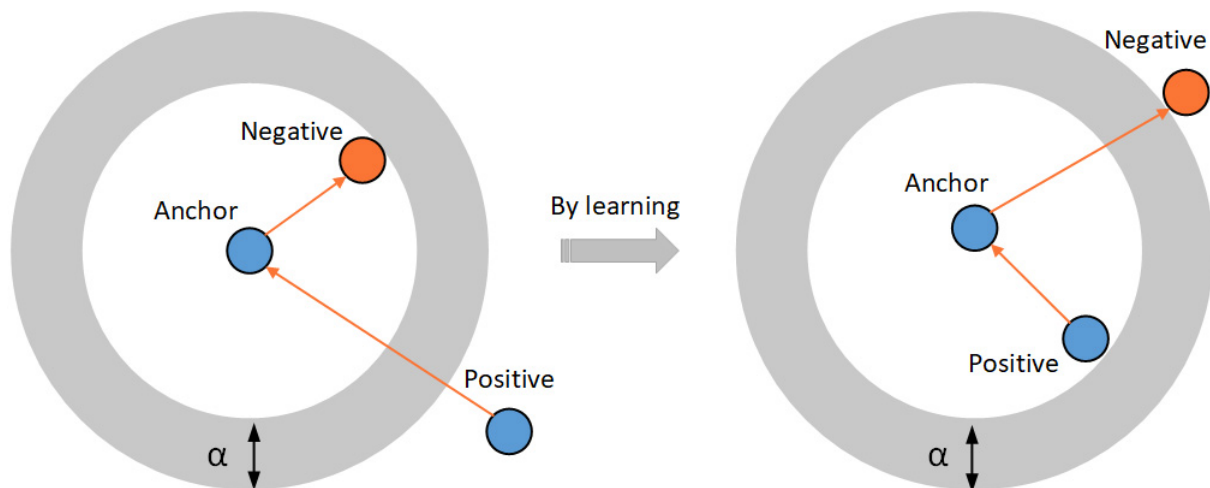


FIGURE 2. Schematic diagram of learning objectives of triplet loss function.

the structure of the agent based on the triplet CapsNet metric is shown in Figure 3. These three capsule subnetworks in Figure 3 share the same structure. For any input image patch, each CapsNet outputs a recognition confidence vector. CapsNet tries to remember each detail of an input image, which will result in a huge computation. In this algorithm, the input image is convolved three times before the capsules are used to encapsulate the image's features, to ignore the general features in low-level and reduce the amount of computation in the training process. Figure 4 shows the structure of a CapsNet in Figure 3. The ultimate learning objective of this algorithm is for the TriCapsNet agent to accurately determine whether a given positive or negative sample, in terms of its feature representation distance metric, belongs to the category of the anchor for classification, as shown in Figure 2.

In the process of reinforcement learning, an image sample  $(s_i, y_i)$  in the training sample set  $D = (s_i, y_i)_{i=1}^B$ . The feature embedding of  $s_i$  is expressed as  $\text{CapsNet}(s_i, \omega_c) \in R$ , where

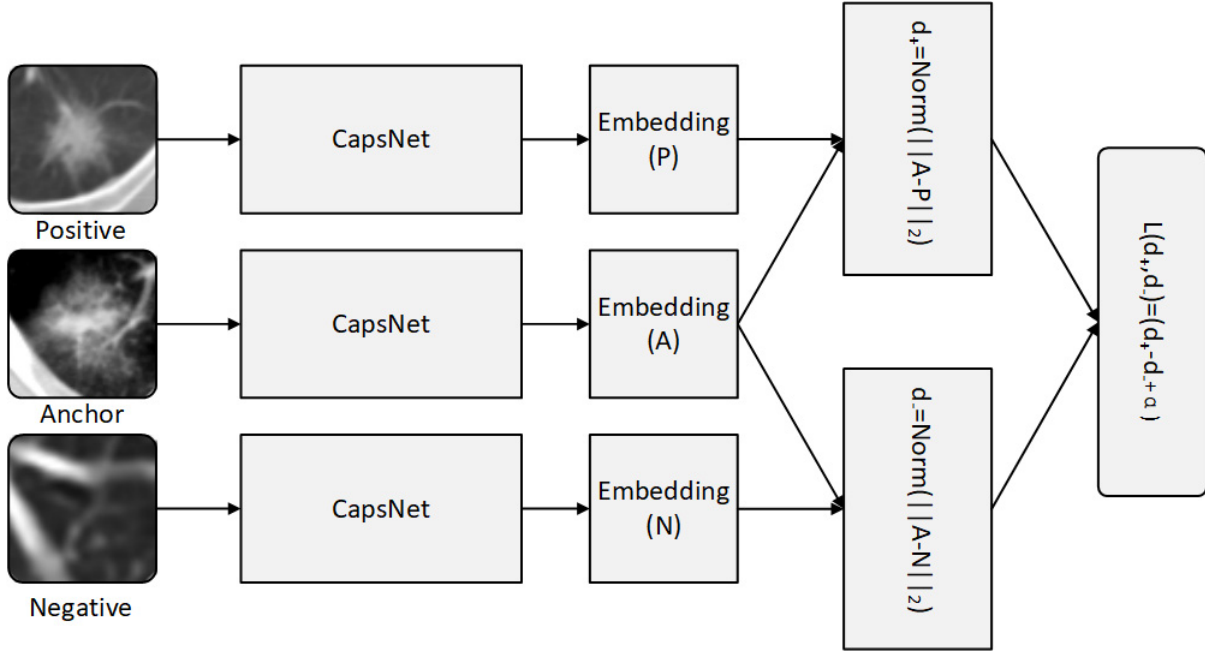


FIGURE 3. Structure of triplet CapsNet.

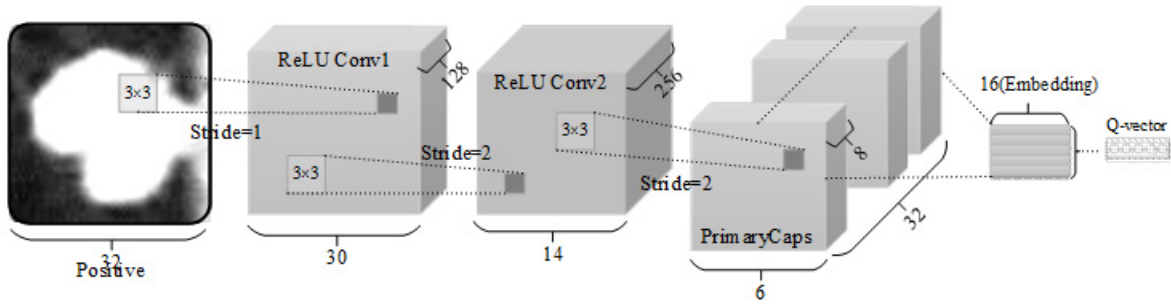


FIGURE 4. The network structure of single CapsNet.

$\omega_c$  is the parameter of Capsnet. In this problem, the feature expression of  $s_i$  is also the  $Q$  value of state  $s_i$ , and the label  $y_i$  is as the corresponding action of the maximum reward of  $s_i$ .

In the learning stage, three states should be selected each time to form a triplet  $T_i(s_i^A, s_i^P, s_i^N)$  and sent to the triplet capsule network. The elements selected to make up  $T_i$ , whose category labels should meet the condition of  $y_i^A = y_i^P \neq y_i^N$ . The utility of the loss function of the triplet CapsNet is to pull samples of the same category to the points near the manifold surface during learning and push the samples belonging to different categories away from each other as much as possible. Then the loss function (1) can be expressed as Formula (2):

$$L_j(d_+, d_-) = \|d_+ - d_- + \alpha\|_2^2 \tag{2}$$

Based on Formula (2), further regularization is carried out as Formula (3) and (4):

$$d_+ = \frac{e^{\|CapsNet(s_i^A) - CapsNet(s_i^P)\|_2}}{e^{\|CapsNet(s_i^A) - CapsNet(s_i^P)\|_2} + e^{\|CapsNet(s_i^A) - CapsNet(s_i^N)\|_2}} \tag{3}$$

$$d_- = \frac{e^{\|CapsNet(s_i^A) - CapsNet(s_i^N)\|_2}}{e^{\|CapsNet(s_i^A) - CapsNet(s_i^p)\|_2} + e^{\|CapsNet(s_i^A) - CapsNet(s_i^N)\|_2}} \quad (4)$$

The loss function adopted by TriCaps-RL in the initial stage of reinforcement learning is as Formula (5):

$$L(\omega) = E[(r_{t+1} + \gamma * \max Q(s_{t+1}, \alpha_{t+1}, \omega_-)^2] \quad (5)$$

where  $\omega$  is the parameter of the capsule neural network to be optimized,  $r(t+1) + \gamma * \max Q(s(t+1), a(t+1), \omega_-)$  is the target value. The function of the target  $Q$  value is as shown in Equation (6):

$$Q_{target} = r_{t+1} + \gamma \max_{a_{r+1}} \tilde{Q}(s_{t+1}, a_{t+1}, \omega_-) \quad (6)$$

In the equation,  $\gamma$  is a discount factor. The network is used to calculate the value of the objective function, which is dynamic and changes with the update of the  $Q$ -value prediction network.

In the initial stage of the learning, the update of the networks' weight is calculated by the loss function (as shown in Equation (5)) of the subnetwork (A) (shown in Figure 3). In the second learning stage, the triplet network metric function is used to further improve the accuracy of action strategy.

Therefore, in the whole process, the TriCaps-RL method can be utilized for reinforcement learning using the following loss function:

$$\begin{aligned} L_{compose} &= \frac{1}{B} \sum_{j=1}^B ((1 - \lambda)L_j(\omega) + \lambda_j(d_+, d_-)) \\ &= \frac{1}{B} \sum_{j=1}^B ((1 - \lambda) \|Q_{target} - \sum_{i=1}^M R(\max_a Q_c(s_{i,j}^A, a_{i,j} : \omega_c))\|^2 \\ &\quad + \lambda \|d_+ - d_- + \alpha\|^2) \end{aligned} \quad (7)$$

Where  $B$  is the number of elements in each small batch randomly selected during experience playback,  $R(\cdot)$  is the reward function and  $\lambda$  is the weight adjustment factor, which increases with the number of learning iterations.

Because of the introduction of the  $\alpha$  boundary, the tri-CapsNet agent can perform classification tasks more finely. However, it is difficult to train a triplet network from a completely new initial state. Moreover, it is not easy to converge. In order to overcome this difficulty, the reinforcement learning is conducted using a single CapsNet in the initial phase, and adopt an adjustment factor  $\lambda$  to complete the transition procedure, as is shown in (8):

$$\lambda = \frac{1}{1 + e^{-(0.005 \times (n - EPISODES))}} \quad (8)$$

The *EPISODES* in Equation (8) is a preset value obtained by experiments. The purpose of this equation is that: when the number of iterations  $n$  is less than the *EPISODES*, the algorithm uses the loss equation of subnetwork (A) in Figure 3 to improve the classification performance, which can converge at a fast speed. When the number of iterations  $n$  increases close to *EPISODES*, the accuracy of the algorithm will reach a bottleneck value and it cannot be significantly improved. Thus, the loss function is gradually transited to

the triplet loss after this anchor point, as shown in Equation (7). The triplet loss function can boost the classification accuracy again from the performance plateau.

When the TriCaps-RL method is used for classification, the multiclass classification task is transformed into two classification problems by feature similarity measurement. By this means, different types of samples can be distinguished more finely. Suppose  $s^A$  is a sample to be classified, and  $s^A$  belongs to one of the  $C$  possible categories. From the labelled sample set,  $N^P$  positive individuals ( $s^P$ ) are randomly selected from each category, for a total of  $N^P \times C$ . Each selected  $s^P$  is integrated with a sample  $s^A$  to compose a sample pair  $(s^A, s_j^i)$ ,  $i \in [0, C - 1]$ , which are fed to the triplet CapsNets respectively. Then, the tri-CapsNet agent outputs the predicted category for  $s^A$ , that  $s^A$  has the largest average feature similarity with the samples selected from it. The formula is as follows:

$$y^* = \operatorname{armin}_{y^i} \frac{1}{N^P} \sum_{j=1}^{N^P} \| \operatorname{CapsNet}(s^A) - \operatorname{CapsNet}(s_j^i) \| \quad i \in [0, C - 1] \quad (9)$$

Where  $y^*$  is the final output category,  $s^A$  is the sample to be tested,  $s_j^i$  is the  $j$ th sample from the  $i$ th base class in the test set,  $j \in [1, N^P]$ .

**4. Algorithmic procedures.** Based on the above steps, the reinforcement learning process of the TriCaps-RL method can be divided into three stages: 1) In stage A, the agent receives a state (image block) from the environment, and selects the best action (identify the category of image block) according to the learned strategy to execute, then the algorithm calculates its  $Q$  value, stores the current state and action as a set along with the selected reference samples temporarily, and receives the next state. This process is learned iteratively until the end of an episode. Here, "an episode" means that the agent performs corresponding actions to all states obtained from the CT image by the sliding window method. 2) In stage B, the algorithm calculates the corresponding rewards according to the predefined rules. The reward in the temporary element group are updated and stored in the memory pool. 3) In stage C, the experience playback stage is performed to update the parameters of the  $Q$  network, and a new episode start.

The specific process of TriCaps-RL algorithm is as follows:

---

Algorithm: TriCaps-RL

**Input:** A CT slice image

**Output:** Trained TriCaps-RL

---

- (1) Initialize the storage amount of replay memory pool  $D$  to  $N$
- (2) Initialize the temporary memory pool  $D_t$  with a capacity of  $M$
- (3) Initialize the reward value  $r$
- (4) Initialize  $\alpha$
- (5) Initialize the parameter  $EPISODES$  of  $\lambda$
- (6) The parameter of the random initialization action value function  $Q_c$  is  $\omega_c$
- (7) Iteration to meet the termination condition
- (8) Initialize state  $s$
- (9) Iterates - until all image blocks on a CT are taken
- (10) Randomly select an action  $a_t$  with probability  $\varepsilon$
- (11) Randomly select the reference base class state  $S_t^P, s_i^P \in S_t^P, i \in [1, M]$
- (12)  $r \leftarrow 0$
- (13) Store  $\langle s_t^A, s_t^P, a_t, r_t, s_{t+1} \rangle$  to  $D_t$
- (14)  $s_t \leftarrow s_{t+1}$
- (15) Iteration to traversal memory pool  $D$



- (16) Take  $\langle s_t^A, s_t^P, a_t, r_t, s_{t+1} \rangle$  from memory pool  $D$
- (17) Environment returns  $r_t$  by comparing with the label marked by the doctor
- (18) Store  $\langle s_t^A, s_t^P, a_t, r_t, s_{t+1} \rangle$  in memory pool  $D$
- (19) If the block on a CT ends, then  $y_j \leftarrow r_j$ , otherwise  $y_j \leftarrow Q_{target}$
- (20) Randomly select a small batch of  $B$  sequences from memory pool  $D$ ,  $\langle s_j^A, s_j^P, a_t, r_t, s_{t+1} \rangle$
- (21) For each  $\langle s_t^A, s_t^P, a_t, r_t, s_{t+1} \rangle$ , randomly select annotation samples of different classes from the existing sample library  $s_t^P$
- (22) Recalculate  $\lambda$  according to the number of iterations
- (23) Use the loss function  $\frac{1}{B} \sum_{j=1}^B ((1 - \lambda)L_j(\omega) + \lambda_j(d_+, d_-)) = \frac{1}{B} \sum_{j=1}^B ((1 - \lambda) \|Q_{target} - \sum_{i=1}^M R(\max_a Q_c(s_{i,j}^A, a_{i,j} : \omega_c))\|^2 + \lambda \|d_+ - d_- + \alpha\|^2)$  to train the network  $Q_c$
- (24) Copy  $\omega_c$  of  $Q_c$  network to the other two networks

In TriCaps-RL algorithm above,  $Q_c$  and  $\omega_c$  mean the CapsNet's  $Q$  value function and its network parameters, respectively, the values of the other two networks' parameters, in the agent are the same as  $\omega_c$  in the agent.

**5. Applying TriCaps-RL to pulmonary nodules classification.** A  $512 \times 512$ -pixel CT image to be diagnosed is used as an example to illustrate the TriCaps-RL based signs classification process. Firstly, different image blocks are acquired one by one from this CT image using a sliding window of  $K \times K$  ( $K < 512$ ) pixels. The coordinate of the pixel  $(x_i, y_i)$  on the upper-left corner of each image block are recorded simultaneously. All the extracted image blocks from the CT image make up a state set  $S$  ( $s_t \in S$ ). Here,  $S$  serves as the environment  $E$  in which the agent interacts with. For any state  $s_t$  in  $E$ , the agent of the algorithm will select an action  $a_t$  from the action space  $A$  according to its learned strategy  $\pi$  to execute. In this problem, the action  $a_t$  means to identify the class  $c_j$  of the image block  $s_t$ . The blocks obtained from the CT image are performed actions by the agent one by one, and the class labels identified by the actions will be displayed back to the CT image according to the upper-left coordinate of the image block. By comparing the original class label (If this algorithm is applied to a diagnosis process, the diagnosis results of the radiologist are used as original class labels) of a  $s_t$  (image block) with the identified result from the agent, the algorithm will produce a corresponding reward. Depending on the rewards, the loss of Formula (6) will be performed, and the iterative learning is carried out to improve the classification performance of TriCaps-RL.

For the problem to be analyzed in this paper, the diagnosis environment of the algorithm is composed by the image blocks of a CT image and its labels, which is different from the exist reinforcement learning application paradigms. In the paper by Mnih et al. [21], different actions of the agent caused changes of the environment, and produced different subsequent states. In this problem, the size of the sliding window and the sliding step are invariant, so the number of the image blocks acquired from each CT is determined. Therefore, the number of states in each episode is also determined during the reinforcement learning process. The target  $Q$ -value is the sum of all the rewards, and it is also a fixed number in this problem. Assuming that the number of states obtained from a CT slice is  $M$ , and the reward for each correct action is  $r$ , the expected value of the objective function  $Q_{target}$  is:

$$Q_{target} = M \times r \quad (10)$$

TABLE 1. Signs selection scheme for experimentation

	Characteristic visibility grade of selected samples	Characteristic visibility grade of excluded samples
subtle	-	-
Internal Structure	-	-
Calcification sign	= 4	<> 4
Sphericity	-	-
Margin	-	-
Lobulation sign	>= 4	< 4
Spiculation sign	>= 4	< 4
GGO sign	<= 2	> 2
Malignancy	>= 3	< 3

For a state, assuming that its class label identified by the agent is  $C_A$ , and annotated by the radiologists is  $C_R$ , then:

$$r = \begin{cases} r, & \text{if } |C_A - C_R| = 0, \\ 0, & \text{if } |C_A - C_R| \neq 0 \end{cases} \quad (11)$$

The loss function is:

$$L_j(\omega_{cj}) = E_{s,a \sim \text{TriCapsnets}(\cdot)} [(Q_{target} - \sum_{i=1}^M R(Q_c(s_i, a_i : \omega_{cj})))^2] \quad (12)$$

In Equation (12),  $a \sim \text{TriCapsnets}(\cdot)$  means the action  $a$  with the maximum  $Q$  value chosen by the agent for the new state under its learned strategy.  $Q_{target}$  represents the value of the objective function of the  $j$ -th iteration. In this problem, it is a constant calculated by Equation (10).

Having established the TriCaps-RL framework, we can now turn our attention to the experimental results.

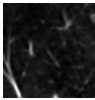
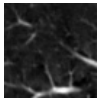
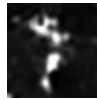



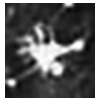
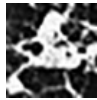
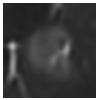
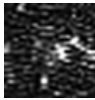
## 6. Experiment and result analysis.

**6.1. Database.** In the LIDC-IDRI database, the margin of each nodule was delineated in detail by several radiologists, and the nodule was evaluated as benign or malignant in grade. The LIDC-IDRI database contains 1018 cases and 20000 CT slice images. Each image is 512 pixels in length and width. In this study, the four kinds of positive labeled data in LIDC-IDRI are used as a baseline for the environment to evaluate the reward to the agent's actions in the TriCaps-RL algorithm.

The greater the degree of Lobulation sign, Spiculation sign, or GGO sign, the greater the possibility of malignant nodules [22]. The majority of Calcification sign nodules are benign, while the most of non-central calcification sign nodules are malignant [22]. According to the nodules' visual characteristics and malignant degree ranked by radiologists, nodules are usually malignant when the degree of its calcification sign is grade 4 (non-central appearance), the Lobulation sign or Spiculation sign is above grade 4, and the texture sign is grade 1 and 2. Therefore, in this paper, the nodules with calculation = 4, location  $\iota=4$ , dispersion  $\iota=4$ , texture  $\jmath=2$ , malignancy  $\iota=3$  were selected as positive experimental samples according to the scheme shown in Table 1.

In addition to these four types of common signs related to cancer, normal tissue image blocks were also selected as negative samples for comparison purposes. Five types of nodule sign images were available in total: non-central calcification, lobulation, spiculation, and GGO. Some instances of selected samples are shown in Table 2. In the table above, Sbd, Isd, Cald, Shd, Mgd, Lbd, Spd, and Nond represent the degrees of subtlety, internal structure, calcification, spherical, margin, lobulation, spiculation, and non-solid texture, respectively.

TABLE 2. Normal tissue, Non-central Calcification, Lobulation, Spiculation and GGO signs samples

	Negative		Non-Calcification		Lobulation		Spiculation		GGO	
										
Sbd	-	-	4.67	5.00	5.00	4.00	4.00	5.00	3.67	3.00
Isd	-	-	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Cald	-	-	<b>4.00</b>	<b>4.00</b>	6.00	6.00	6.00	6.00	6.00	6.00
Shd	-	-	3.33	2.67	4.25	4.00	5.00	5.00	4.33	4.00
Mgd	-	-	3.67	4.67	4.50	4.00	4.00	1.00	2.67	1.00
Lbd	-	-	1.67	2.00	<b>4.25</b>	<b>4.00</b>	4.00	1.00	1.67	1.00
Spd	-	-	1.33	1.33	2.25	4.00	<b>4.00</b>	<b>5.00</b>	1.67	2.00
Nond	-	-	5.00	4.67	4.75	4.00	5.00	4.00	<b>1.33</b>	<b>1.00</b>

The positive sample images cropping scheme is as follows: the union of the areas marked by different experts on a lesion is considered a whole region, and the center point of this region is used as the center for the image block to be cropped. The size of the cropped blocks is  $32 \times 32$  pixels. In addition, for experimental comparison, negative samples of  $32 \times 32$  pixels are also randomly selected from the normal lung parenchymal region, as shown in Table 2.

**6.2. Experimental setup.** To facilitate the comparison of experimental results, nodule characteristics provided by radiologists were graded according to its appearance. Table 2 illustrates the selected sample selection rules in this paper. The sign image blocks were increased through rotation and flip operations. A total of 12,000 CT images were generated, and 5 cross-validation experiments were conducted.

To normalize the intensity of the images and reduce the influence of artifacts on CT images, the  $z$ -value of each image is calculated by subtracting the average pixel intensity  $\mu$  of all CT images from each image  $X$  and then dividing it by the standard deviation  $\sigma$  of the pixel intensity of all images.

$$z = \frac{X - \mu}{\sigma} \quad (13)$$

Sliding window acquisition operations are performed for each CT image with a  $32 \times 32$ -pixel window and step interval of 11 pixels. Accordingly, 2116 image pictures ( $46 \times 46$ ) can be acquired from each CT. In this experiment, each correct action or classification yields a reward with a value of  $r = 1$ , so the  $Q_{target}$  for each episode is 2116. For each state generated from a CT image using a sliding window, the agent has 5 actions to choose from: non-central calcification, lobulation, spiculation, GGO sign, or negative. After performing an action, the algorithm will return corresponding reward values based on the baseline sample (doctor’s diagnosis).

Figure 5 is a paradigm of how the algorithm evaluates the reward of an agent’s action, where the LIDC-IDRI database simulates the environment. Figure 5(a) shows the diagnostic results of TriCaps-RL and the classification result are sent to the CT image. Figure 5(b) exhibits the comparison between the original annotation of the image blocks and the diagnostic results of TriCaps-RL. The position of block 3 in image (b) indicates that a diagnostic result of TriCaps-RL coincides with the doctor’s annotation area.

In Figure 5, the area outside the green square represents normal tissue. The original category labels of the image blocks from which we extracted are all negative.

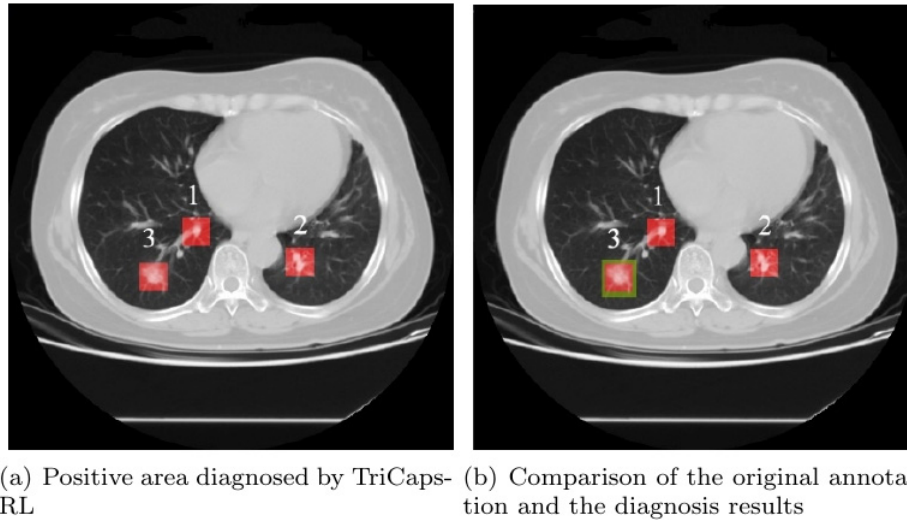


FIGURE 5. Paradigm of TriCaps-RL's diagnostic results on LIDC-IDRI.

The algorithm computes the reward for each image block by comparing the discrepancy between the baseline and the method's predicted output. If the two are the same, the reward is 1; otherwise, the reward is 0. As shown in Figure 5(b), for block 3, the diagnostic result of TriCaps-RL is consistent with the baseline, so the environment rewards 1. The rewards can also be evaluated by comparing with the radiologist's diagnosis, when learning through interaction online with them. In Figure 5(b), the agent considers region 1 and 2 as positive nodules, but it is not consistent with the baseline labels, so the algorithm returns reward 0.

When comparing the original annotation of the samples with the classification results of TriCaps-RL, the reward algorithm returns a value according to the following principles:

- 1) If 2/3 of an image block overlaps with the lesion annotated by the radiologist, its label given by the radiologist is used as the reference label;
- 2) If the overlap is less than 2/3, the algorithm's classification result is considered correct if the classification result of the algorithm and the radiologist's annotation are all positive. Else if the algorithm identifies this image block as negative, it is also considered to be correct. For the other situations, the results are incorrect.

When training TriCaps-RL, the maximum value ( $N$ ) of the experience replay memory pool ( $D$ ) is set to 3K. 50 small batch samples are extracted from  $D$  at a time. For  $\epsilon$  greedy strategy,  $\epsilon$  reduces linearly from 1 to 0.1 in 3000 episodes. Empirically, for *EPISODES* in Equation (8), it is set to 22000. In the sign recognition phase, the size of category space  $C$  is 5.  $N^p = 10$  individual samples are randomly selected from each category at a time, for a total of 50 samples (with  $N^p \times C = 50$ ), and then 50 sample pairs were composed for validation.

**6.3. Learning process.** Figure 6 shows the increasing curve of TriCaps-RL's validation accuracy during the learning process.

As can be seen from Figure 6, the accuracy of TriCaps-RL varies greatly until 2600 episodes. The reason for this phenomenon is that the data in the experience memory pool is not completely updated, so the change of the loss is not stable. After 2600 episodes, the accuracy tends to rise steadily with the accumulation of experience. After 14000 iterations of training, it reaches a plateau and becomes stable. When the value of episodes exceeds 22000, the accuracy achieves a higher plateau again. This is why the

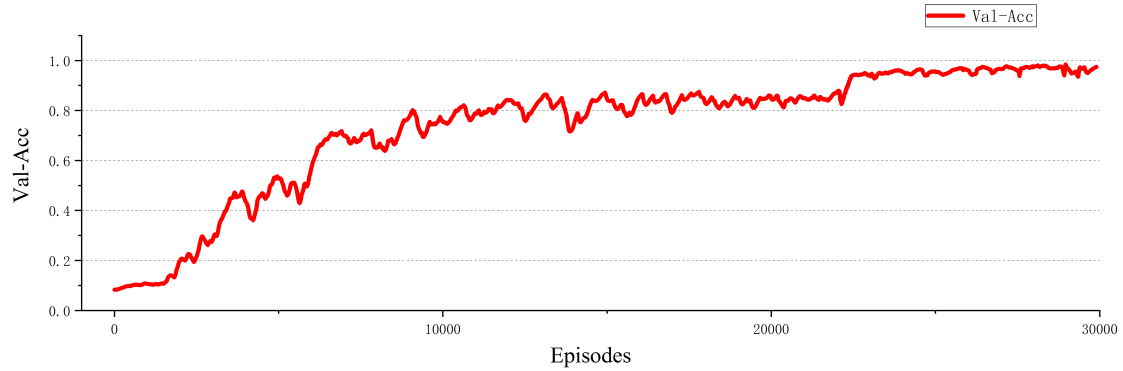


FIGURE 6. Training verification accuracy curve of TriCaps-RL.

value of EPISODES in Equation (9) is set to 22000. After 22000, the loss of the algorithm gradually transits to the triplet loss function shown in (2).

**6.4. Learning process.** In order to visualize how the embedding representations changes with the learning process, the feature embedding representations of five types of test samples is visualized in three-dimensional space after dimensionality reduction. The graphs in Figure 7 show the feature distributions of the five categories as the training process advances. It can be seen that the embedding representations of one kind converge gradually with the increase of the episodes, while the feature embedding representations of different kinds diverge simultaneously.

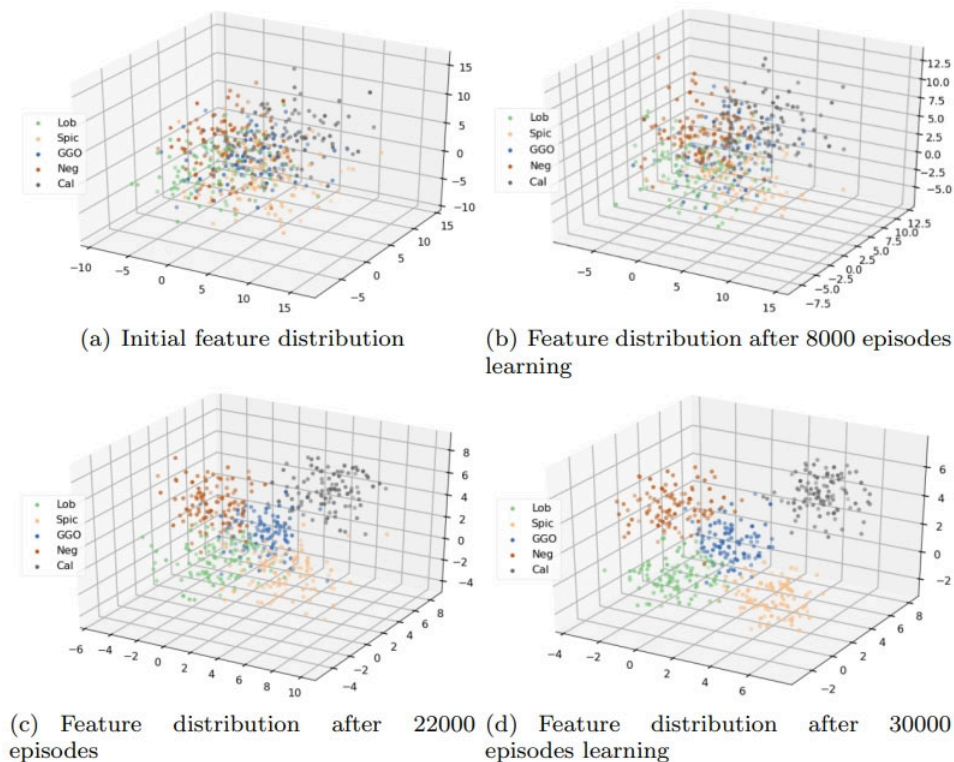


FIGURE 7. The embedding feature representation of five kinds of signs with the increase of the number of learning episodes.

## 6.5. The classification performance TriCaps-RL method.

### 1) ROC

Figure 8 shows the ROC (Receiver Operating Characteristic) curve of TriCaps-RL's classification performance on each category. The AUC of non-central calcification, negative, lobulated, spiculation, and GGO signs are 0.9193, 0.9072, 0.9009, 0.9004 and 0.8957, respectively. It can also be seen that the TriCaps-RL method has the highest overall classification accuracy on non-calcification signs. AUC of lobulation sign, spiculation sign, and GGO sign are lower than that of non-central calcification and negativity.

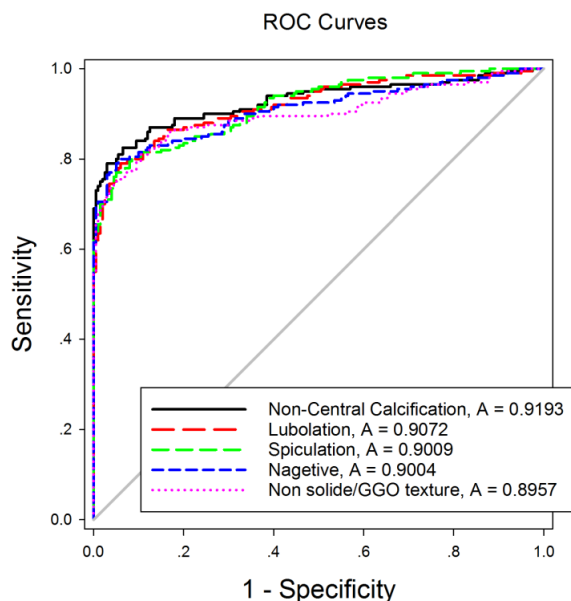


FIGURE 8. ROC curve of TriCaps-RL.

### 2) Accuracy confusion matrix of TriCaps-RL

In order to further analyze the potential classification error distribution, an accuracy confusion matrix was made for the five category samples, as shown in Figure 9.

<b>Ca1</b>	<b>0.9279</b>	<b>0.0337</b>	<b>0.0234</b>	<b>0.012</b>	<b>0.003</b>
<b>Lob</b>	<b>0.0117</b>	<b>0.9243</b>	<b>0.0294</b>	<b>0.0218</b>	<b>0.0128</b>
<b>Spic</b>	<b>0.0008</b>	<b>0.053</b>	<b>0.9097</b>	<b>0.024</b>	<b>0.0125</b>
<b>Neg</b>	<b>0.0317</b>	<b>0.0258</b>	<b>0.0196</b>	<b>0.9192</b>	<b>0.0037</b>
<b>GGO</b>	<b>0.0032</b>	<b>0.0171</b>	<b>0.0262</b>	<b>0.0552</b>	<b>0.8983</b>
	<b>Ca1</b>	<b>Lob</b>	<b>Spic</b>	<b>Neg</b>	<b>GGO</b>

FIGURE 9. The accuracy confusion matrix for each kind of samples.

TABLE 3. CNN network architecture for comparison

Layers	Size of the input	Size of the kernel	Step width
1	$32 \times 32$	5	1
2	$28 \times 28$	2	<i>MaxPooling</i>
3	$14 \times 14$	5	1
4	$10 \times 10$	5	1
5	<i>flatten</i>		
6	512		
7	128		
8	5		

The numbers on the diagonal of the confusion matrix indicate the recognition accuracy of the corresponding categories, while the elements on the non-diagonal are the misclassification rate. As can be seen from Figure 9, the non-central calcification signs have the highest classification accuracy, while the GGO signs have the highest misclassification rate. Most of the misclassifications occurred among lobulation, spiculation, and GGO signs. In addition, the majority of misclassified spiculation sign samples were considered as lobulation signs. By analyzing the visual appearance of the two types of test samples, it was observed that a significant number of samples from both categories exhibit remarkably similar

### 6.6. Performance comparison between TriCaps-RL and DQN.

1) DQN, short for Deep Q Network, is a classical deep reinforcement learning algorithm, is a classical deep reinforcement learning algorithm. For comparison, a new DQN is established in this paper according to the structure shown in Table 3. The training data is the same as that of the TriCaps-RL, and the final test results are calculated and compared with TriCaps-RL. The agent of DQN has three convolutions, and the loss function is shown in Equation (12).

The average classification performance of the two methods is shown in Figure 10.

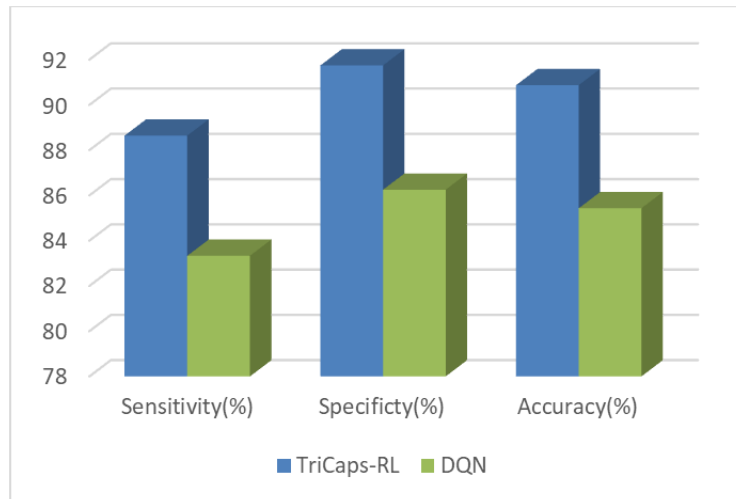


FIGURE 10. Performance comparison of TriCaps-RL and DQN.

From Figure 10, it can be observed that the TriCaps-RL is superior to DQN in sensitivity, specificity, and accuracy, showing that the structure of triple capsule network is reasonable and effective.

2) In order to perform a finer grained analysis and see the performance differences between the two methods in different categories, a confusion matrix on accuracy differences between TriCaps-RL and DQN classification is also constructed, as shown in Figure 11. The floating points on the diagonal represent the difference between TriCaps-RL and DQN in the classification accuracy of the corresponding categories, while the floating points on the non-diagonal represent the difference in classification error rate, showing the error ratio of each category that is misclassified.

The values on the diagonal represent the difference in accuracy between TriCaps-RL and DQN for different categories. For elements not in the diagonal, it means the difference in the classification error rate. As shown in Figure 11, the element values on the diagonal of the confusion matrix are all positive, which means that the TriCaps-RL algorithm obtains higher accuracy in each category. The average value of the diagonal number is 0.1047, and the overall classification performance is better than that of the DQN. The absence of positive values in the non-diagonal values means that TriCaps-RL has a lower error classification rate than DQN in all categories. 3) Contrast of reconstruction of sign

<b>Ca1</b>	<b>0.0594</b>	-0.0267	-0.01849	-0.01211	-0.0021
<b>Lob</b>	-0.0038	<b>0.0479</b>	-0.0223	-0.0112	-0.0106
<b>Spic</b>	-0.0047	-0.077	<b>0.1413</b>	-0.0461	-0.0135
<b>GGO</b>	-0.0649	-0.0332	-0.0205	<b>0.1277</b>	-0.0091
<b>Neg</b>	-0.0097	-0.0217	-0.0424	-0.0735	<b>0.1473</b>
	<b>Ca1</b>	<b>Lob</b>	<b>Spic</b>	<b>GGO</b>	<b>Neg</b>

FIGURE 11. Confusion matrix of accuracy difference between TriCaps-RL and DQN.

samples

A comparison is made with other reinforcement learning methods. The four classes positive samples selected in this paper are viewed as a malignant class, and the negative samples are used as comparisons. Two classes of classification experiments are also performed, and the performances are compared with the following methods:

(1) The reinforcement learning method based on a single CapsNet, referred to as CapsNet-RL, the agent in CapsNet-RL has the same structure as the sub-network in TriCaps-RL algorithm shown in Figure 5, using the loss function shown in Equation (12).

(2) Silva et al. [13] proposed a method for lung nodule classification based on the combination of artificial 3D image features and reinforcement learning method.

(3) Ali et al. [14] suggested a deep reinforcement learning method for lung nodule classification. The input is a group of continuous  $512 \times 512$ -pixel CT images on the Z-axis as a state, and the agent is a CNN network with multilayer convolution and pooling.

(4) Mobiny and Nguyen [10] indicated a lung nodule classification method based on CapsNet, and improved the algorithm of the original capsule network in two aspects: a). The algorithm forces all capsules corresponding to a pixel in the PrimaryCaps layer



TABLE 4. CNN network architecture for comparison

Methods	Sensitivity(%)	Specificity(%)	Accuracy(%)
Silva et al. [12]	-	-	81
Ali et al. [13]	58.9	55.3	64.4
Mobiny et al. [9]	89.11	-	91.84
CapsNet-RL	88.75	90.17	90.62
TriCaps-RL	<b>90.63</b>	<b>93.77</b>	<b>92.89</b>

to have the same routing coefficient, which improves the speed of the algorithm three times. b). In the reconstruction phase, different from the original mask output using node capsules, unlike using the original mask output of the node capsules, the authors use PrimaryCaps prediction vector and one hot coding prediction vector as the input for the deconvolution network.

The sensitivity, specificity, and accuracy of TriCaps-RL and CapsNet-RL are tested on validation set, and compared with the algorithms of Silva, Ali, and Mobiny. The results are shown in Table 4.

As can be seen from Table 4, the TriCaps-RL and CapsNet-RL algorithms outperform the first two algorithms. The performance of TriCaps-RL is significantly better compared to the reinforcement learning method with a single CapsNet.

To summarize our findings and contributions, let's proceed to the conclusion of our study.

**7. Conclusion.** In this paper, to classify pulmonary nodules more finely, a triplet CapsNet and its reinforcement learning method (TriCaps-RL) are proposed. By combining CapsNets with the triplet learning strategy, the TriCaps-RL method first achieves good performance on a small training set, and then uses triple feature representation metrics to boost its performance again. This makes the intra-class expression of sign samples closer, the inter-class expression farther, and the classification performance is further improved. In the first phase, the Q-loss function of a single CapsNet is used to learn a primary action selection strategy. In the second phase, the loss function of the method is gradually transformed into a triplet loss metric. This shift not only solves the problem that the triplet network does not easy to converge, but also can obtain better classification performance.

Experimental results show that the TriCaps-RL learning algorithm is superior to the existing lung nodule classification algorithm. This is the first time CapsNet has been combined with deep reinforcement learning technology to classify pulmonary nodules.

This method can be applied for clinical diagnosis. Through iterative interaction with radiologists during CT reading procedure, the performance of the method can be continuously improved, and thereby the current dilemma of doctors having not enough time to annotate medical samples can be overcome. At the same time, this method can provide increasingly accurate diagnostic reference and gradually improve the diagnosis efficiency of the physicians.

**Acknowledgment.** This work was supported in part by the Natural Science Foundation of China (NSFC) under Grants No. 61962059, 61702037, 62262067, 61773062, Science and Technology Plan of the Beijing Municipal Education Commission Funder Grant No.KM202010011012 and Yan'an Science and Technology Plan Projects Funder Grant No.2022SLGYGG-007. We thank Chaoyang Hai for his technical assistance in the laboratory.

## REFERENCES

- [1] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, and Z. Li, "Human motion recognition based on SVM in VR art media interaction environment," *Human-centric Computing and Information Sciences*, vol. 9, pp. 1–15, 2019.
- [2] K.-K. Tseng, C. Wang, T. Xiao, C.-M. Chen, M. M. Hassan, and V. H. C. de Albuquerque, "Sliding large kernel of deep learning algorithm for mobile electrocardiogram diagnosis," *Computers & Electrical Engineering*, vol. 96, p. 107521, 2021.
- [3] E. K. Wang, C.-M. Chen, M. M. Hassan, and A. Almogren, "A deep learning based medical image segmentation technique in Internet-of-Medical-Things domain," *Future Generation Computer Systems*, vol. 108, pp. 135–144, 2020.
- [4] O. Akter, M. A. Moni, M. M. Islam, J. M. Quinn, and A. Kamal, "Lung cancer detection using enhanced segmentation accuracy," *Applied Intelligence*, vol. 51, pp. 3391–3404, 2021.
- [5] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [6] H. Wang, K. Shao, and X. Huo, "An improved CapsNet applied to recognition of 3D vertebral images," *Applied Intelligence*, vol. 50, pp. 3276–3290, 2020.
- [7] P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain tumor type classification via capsule networks," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, IEEE, 2018, pp. 3129–3133.
- [8] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1386–1393.
- [9] E. Goceri, "CapsNet topology to classify tumours from brain images and comparative evaluation," *IET Image Processing*, vol. 14, no. 5, pp. 882–889, 2020.
- [10] A. Mobiny and H. Van Nguyen, "Fast capsnet for lung cancer screening," in *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 2018, pp. 741–749.
- [11] A. Jiménez-Sánchez, S. Albarqouni, and D. Mateus, "Capsule networks against medical imaging data challenges," in *Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis: 7th Joint International Workshop, CVII-STENT 2018 and Third International Workshop, LABELS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 3*, Springer, 2018, pp. 150–160.
- [12] T. Iesmantas and R. Alzbutas, "Convolutional capsule network for classification of breast cancer histology images," in *Image Analysis and Recognition: 15th International Conference, ICIAR 2018, Póvoa de Varzim, Portugal, June 27–29, 2018*, Proceedings 15, Springer, 2018, pp. 853–860.
- [13] A. C. Silva, V. R. da Silva, A. de Almeida Neto, and A. C. de Paiva, "Diagnosis of lung nodule using reinforcement learning and geometric measures," in *Machine Learning and Data Mining in Pattern Recognition: 4th International Conference, MLDM 2005, Leipzig, Germany, July 9–11, 2005. Proceedings 4*, Springer, 2005, pp. 295–304.
- [14] I. Ali, G. R. Hart, G. Gunabushanam, Y. Liang, W. Muhammad, B. Nartowt, M. Kane, X. Ma, and J. Deng, "Lung nodule detection via deep reinforcement learning," *Frontiers in Oncology*, vol. 8, p. 108, 2018.
- [15] S. G. Armato III, G. McLennan, L. Bidaut, M. F. McNitt-Gray, C. R. Meyer, A. P. Reeves, and B. -S. Zhao et al., "The 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. 915–931, 2011.
- [16] K. Lee, M. Laskin, A. Srinivas, and P. Abbeel, "Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning," in *International Conference on Machine Learning*, PMLR, 2021, pp. 6131–6141.
- [17] A. Bhandary, G. A. Prabhu, V. Rajnikanth, K. P. Thanaraj, S. C. Satapathy, D. E. Robbins, C. Shasky, Y. -D. Zhang, J. M. RS. Tavares, and N. S. M. Raja, "Deep-learning framework to detect lung abnormality—A study with chest X-Ray and lung CT scan images," *Pattern Recognition Letters*, vol. 129, pp. 271–278, 2020.
- [18] T. L. Chaunzwa, A. Hosny, Y. Xu, A. Shafer, N. Diao, M. Lanuti, D. C. Christiani, R. H. Mak, and H. JWL. Aerts, "Deep learning classification of lung cancer histology using CT images," *Scientific Reports*, vol. 11, no. 1, p. 5471, 2021.

- [19] M. A. Khan, S. Rubab, A. Kashif, M. I. Sharif, N. Muhammad, J. H. Shah, Y. -D. Zhang, and S. C. Satapathy, "Lungs cancer classification from CT images: An integrated design of contrast based classical features fusion and selection," *Pattern Recognition Letters*, vol. 129, pp. 77–85, 2020.
- [20] S. Wang, L. Dong, X. Wang, and X. Wang, "Classification of pathological types of lung cancer from CT images by deep residual neural networks with transfer learning strategy," *Open Medicine*, vol. 15, no. 1, pp. 190–197, 2020.
- [21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint*, arXiv:1312.5602, 2013.
- [22] D. E. Ost and M. K. Gould, "Decision making in patients with pulmonary nodules," *American Journal of Respiratory and Critical Care Medicine*, vol. 185, no. 4, pp. 363–372, 2012.