Deep Reinforcement Active Learning for Medical Image Classification

Jingwen Wang[1], Yuguang Yan[2], Yubing Zhang[1(✉)], Guiping Cao[1], Ming Yang[1], and Michael K. Ng[2]

[1] CVTE Research, Guangzhou, China
{wangjingwen7003,zhangyubing,caoguiping,yangming}@cvte.com
[2] The University of Hong Kong, Hong Kong, China
{ygyan,mng}@maths.hku.hk

Abstract. In this paper, we propose a deep reinforcement learning algorithm for active learning on medical image data. Although deep learning has achieved great success on medical image processing, it relies on a large number of labeled data for training, which is expensive and time-consuming. Active learning, which follows a strategy to select and annotate informative samples, is an effective approach to alleviate this issue. However, most existing methods of active learning adopt a hand-design strategy, which cannot handle the dynamic procedure of classifier training. To address this issue, we model the procedure of active learning as a Markov decision process, and propose a deep reinforcement learning algorithm to learn a dynamic policy for active learning. To achieve this, we employ the actor-critic approach, and apply the deep deterministic policy gradient algorithm to train the model. We conduct experiments on two kinds of medical image data sets, and the results demonstrate that our method is able to learn better strategy compared with the existing hand-design ones.

Keywords: Active learning · Deep reinforcement learning · Medical image classification.

1 Introduction

In the last decades, benefiting from the powerful ability of representation learning, deep learning has achieved great success on object recognition, natural image understanding, and medical image analysis [7,13]. Nevertheless, existing methods heavily rely on a large of high-quality labeled data, which is expensive and

J. Wang and Y. Yan—are the co-first authors. Y. Zhang—is the corresponding author. This work was supported by HKRGC GRF 12306616, 12200317, 12300218, 12300519, and 17201020.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-59710-8_4) contains supplementary material, which is available to authorized users.

A. L. Martel et al. (Eds.): MICCAI 2020, LNCS 12261, pp. 33–42, 2020.
https://doi.org/10.1007/978-3-030-59710-8_4

time-consuming. This issue becomes even severer in medical image analysis, since it requires experienced experts to annotate medical images.

Active learning is an effective approach to address the problem of labeled data scarcity [11,16]. In the iterative procedure of active learning, some informative unlabeled samples are selected and annotated. After that, the classifier is trained with the help of new labeled data and is expected to achieve better performance than before. Most existing methods of active learning prefer to select and annotate samples with high uncertainty, since these samples can provide more information and are usually difficult to be classified correctly. Motivated by this, some methods based on uncertainty are proposed, such as least confidence [11], margin sampling [10], entropy [12], *etc.* However, these methods rely on fixed hand-design strategies, which are not able to handle the dynamic procedure of model training. As the model changes, the predefined strategy may be inappropriate to select the most informative samples.

In this paper, to address the above issue, we propose a new active learning algorithm named **D**eep **R**einforcement **L**earning for **A**ctive learning (DRLA) for medical image classification. Rather than adopting a hand-design data selection strategy, we seek to learn a dynamic policy to select samples for annotation. To this end, we model the procedure of active learning as a Markov decision process, and apply deep reinforcement learning [8,14] to learn a data selection strategy, which takes the state of the classifier into consideration, thus can obtain a better strategy compared with hand-design ones. Specifically, we employ the actor-critic approach [3] to generate and judge decisions of data selection, and apply the deep deterministic policy gradient algorithm (DDPG) [6] to train the model. We conduct experiments on two kinds of medical image data sets to evaluate the performance of our proposed method.

1.1 Related Works

Active learning aims to select and annotate informative samples for improving the performance [4,11,16]. The most common strategy is based on data uncertainty. In [11], the least confidence method is proposed to select the samples whose probabilities of the most probable classes are still low. The margin sampling method [10] calculates the margin between the first and second most probable classes for each sample, and selects the samples with small margin values. The entropy method measures the uncertainty of each sample based on the entropy of the predicted class label probabilities [12]. In [17], a fusion strategy is proposed to combine the above methods. In [19], a deep active learning method based on uncertainty and similarity information is proposed for biomedical image segmentation.

Reinforcement learning is a classic algorithm in artificial intelligence [14]. Thanks to the great progress of deep neural network, deep reinforcement learning has shown powerful ability in learning policy for decision problems. Deep Q-learning extends traditional Q-learning by leveraging deep neural network to learn a Q-value function [8]. After that, deep deterministic policy gradient algorithm (DDPG) is proposed to adapt deep Q-learning to handle the continuous

action space [6]. In [2], the Twin Delayed Deep Deterministic policy gradient algorithm (TD3) further extends DDPG by maintaining a pair of critics along with a single actor, which obtains better performance and efficiency.

2 Methodology

2.1 Overview

Figure 1 illustrates the main idea of our proposed method DRLA. A classifier network for disease diagnosis is trained on labeled training data, including samples with label in advance (*i.e.*, (X_l, Y_l)), and samples which are selected and annotated in the procedure of active learning. During the learning process, an actor network is devoted to selecting the most informative samples from unlabeled training data (*i.e.*, X_u) according to the current state and a learned policy. After that, an annotator is responsible to annotate the selected samples. As a result, we have more and more labeled training data to update the classifier gradually. Last but not the least, a critic network is trained to evaluate if the selection of the actor network is effective to improve the performance of the classifier. By employing a deep reinforcement learning approach to train the actor network and the critic network, we can select and annotate the most informative samples that are beneficial for training an effective classifier, and further improve the classification performance.

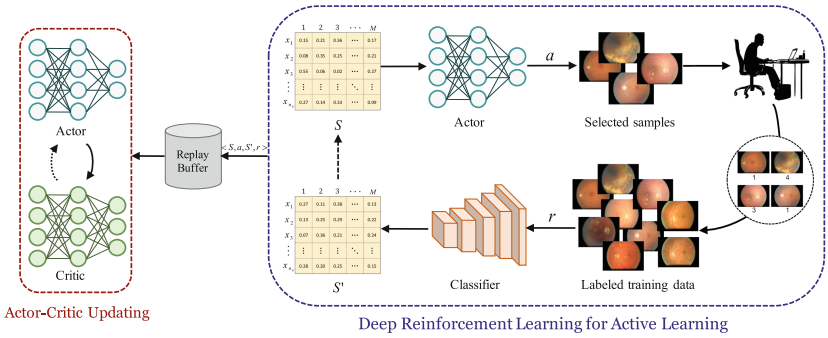


Fig. 1. The illustration of our proposed method DRLA.

2.2 Classifier Training

Let the parameters of the classifier network be θ_d. At the beginning of the learning procedure, we can use labeled training data to pretrain the classifier. After that, we apply deep reinforcement active learning to select and annotate samples, and further train the classifier to enhance the performance. Given the

labeled training samples $X_l = \{x_i\}_{i=1}^{n_l}$ with corresponding labels $Y_l = \{y_i\}_{i=1}^{n_l}$, where n_l is the number of labeled data. The classifier network is trained by minimizing the cross-entropy loss, which is defined as

$$\mathcal{L}_{ce} = -\sum_{i=1}^{n_l}\sum_{j=1}^{M} \mathbb{I}(y_i = j) \log \Pr(y_i = j \mid x_i; \theta_d), \tag{1}$$

where M is the number of classes, $\mathbb{I}(\cdot)$ is the indicator function, and $\mathbb{I}(y_i = j)$ judge if the label of the i-th sample is j or not. $\Pr(y_i = j \mid x_i; \theta_d)$ is the softmax output of the classifier given x_i for the j-th class, which indicates the probability of x_i belonging to label j obtained from the classifier with parameters θ_d.

2.3 Deep Reinforcement Active Learning

In this part, we propose a new active learning method named DRLA to learn a policy, which guides the actor network to select samples for annotation. In the following, we discuss our proposed method in detail.

State. In order to select the samples which are the beneficial for improving the classification performance, the prediction of the current classifier should be taken into consideration. Motivated by this, we design the state $S \in (0, 1]^{n_u \times M}$ as a matrix including all the predicted values for unlabeled training samples X_u, where n_u is the number of unlabeled training samples, and M is the number of the classes. Mathematically, the (i, j)-th element of S is defined as

$$S_{ij} = \Pr(y_i^u = j \mid x_i^u; \theta_d), \tag{2}$$

where x_i^u is an unlabeled training sample, and y_i^u is the corresponding unknown label.

Action. Define θ_a is the parameters of the actor network. Since the target of the actor network is to select samples from unlabeled training data set for annotation, we define the action as a vector $a \in (0, 1)^{n_u}$, each element of which corresponds to an unlabeled training sample. The sigmoid function is adopted as the activation function for each element to obtain a value between 0 and 1. A policy $\pi(S; \theta_a)$ is learned to generate action a based on the state S. After obtaining the action vector, we rank all the candidate samples except the samples which are selected already in descending order, and select the first n_s samples with the highest values for annotation. The selected samples and the labels provided by the annotator are denoted as $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, and the augmented labeled training data are denoted as $(X_l, Y_l) := (X_l, Y_l) \cup \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$.

State Transition. After the selected samples are annotated and added into the labeled training data, we can update the classifier with the augmented training data set (X_l, Y_l) by minimizing the cross-entropy loss in Eq. (1). After that, we can use the new classifier to obtain the new state matrix S' based on Eq. (2).

Reward. In order to enhance the performance of the classifier, we propose to make the actor concentrate more on those samples which are highly possible

to be misclassified by the classifier. To achieve this, we design a novel reward function by considering the predicted values and true labels obtained from the annotator. In specific, for the selected sample x_i^s, define k_i as the true label obtained from the annotator, and $\hat{k}_i$ as the predicted label obtained from the classifier, *i.e.*, $\hat{k}_i = \max_j \Pr(y_i^s = j \mid x_i^s; \theta_d)$. The reward is defined as

$$r(S, a) = \frac{1}{n_s} \sum_{i=1}^{n_s} \Pr(y_i^s = \hat{k}_i \mid x_i^s; \theta_d) - \Pr(y_i^s = k_i \mid x_i^s; \theta_d). \tag{3}$$

If the sample x_i^s is classified correctly, then $k_i = \hat{k}_i$, and $\Pr(y_i^s = \hat{k}_i \mid x_i^s; \theta_d) - \Pr(y_i^s = k_i \mid x_i^s; \theta_d) = 0$. On the other hand, a high reward indicates that the selected samples are classified incorrectly. This implies that these samples with the wrong predictions should be paid more attention by the classifier. Therefore, these samples are encouraged to be selected by the actor network.

At state S, reinforcement learning aims to maximize the expected reward in the future, which is defined as a Q-value function. Similar to Q-learning in traditional reinforcement learning, the Q-value function is used to evaluate the state-action pair (S, a), and is represented by the Bellman equation as $Q(S, a; \theta_c) = \mathbb{E}\big[\gamma Q(S', \pi(S'; \theta_a); \theta_c) + r(S, a)\big]$, where γ is the delay parameter. Here we adopt a critic network with parameters θ_c to approximate the Q-value function. Inspired by deep Q-Learning [8], we aim to learn a greedy policy for the actor by solving the following problem

$$\max_{\theta_a} Q(S, \pi(S; \theta_a); \theta_c). \tag{4}$$

We define $\tilde{Q}(S, a; \theta_c) = \gamma Q(S', \pi(S'; \theta_a); \theta_c) + r(S, a)$, and train the critic network by solving the following problem

$$\min_{\theta_c} \big(\tilde{Q}(S, a; \theta_c) - Q(S, a; \theta_c)\big)^2. \tag{5}$$

Training with Target Networks. In order to stabilize the training of the actor and critic networks, we follow [6] to employ a separate target network to calculate $\tilde{Q}(S, a; \theta_c)$. According to Problem (5), $\tilde{Q}(S, a; \theta_c)$ depends on the new state S', the actor to output action $\pi(S'; \theta_a)$, and the critic to evaluate $(S', \pi(S', \theta_a))$. We adopt a separate target actor network parameterized by $\theta_{a'}$ and a separate target critic network parameterized by $\theta_{c'}$ to calculate $\tilde{Q}(S, a; \theta_c)$. As a result, we rewrite Eq. (5) as

$$\min_{\theta_c} \big(\gamma Q'(S', \pi'(S'; \theta_{a'}); \theta_{c'}) + r(S, a) - Q(S, a; \theta_c)\big)^2, \tag{6}$$

where $\pi'(\cdot; \theta_{a'})$ is the target policy estimated by the target actor, and $Q'(\cdot, \cdot; \theta_{c'})$ is the function of the target critic. This problem can be optimized by the deep deterministic policy gradient algorithm (DDPG) [6].

At the last of each epoch, the target actor and critic are updated by

$$\theta_{a'} := \lambda \theta_a + (1 - \lambda)\theta_{a'}, \quad \theta_{c'} := \lambda \theta_c + (1 - \lambda)\theta_{c'}, \tag{7}$$

where $\lambda \in (0, 1)$ is a trade-off parameter.

To train the actor and critic in the mini-batch paradigm, we use a replay buffer to store samples $\{(S, a, S', r)\}$. As a result, we can uniformly select training samples from the replay buffer to update the actor and critic networks [6].

Algorithm 1 summarizes our proposed method.

Algorithm 1. Deep Reinforcement Learning for Active learning (DRLA)

Input: Labeled training data (X_l, Y_l), unlabeled training data X_u.
Initialize: Pretrain the classifier to get $f(\cdot; \theta_d)$ based on labeled training data (X_l, Y_l).
for each epoch **do**
  Compute the state S according to Eq. (2).
  Select n_s unlabeled training sample $\{x_i^s\}_{i=1}^{n_s}$ based on the actor $a = \pi(S; \theta_a)$.
  Annotate $\{x_i^s\}_{i=1}^{n_s}$ to get $\{(x_i^s, y_i^s)\}_{i=1}^{n_s}$.
  Update the classifier parameters θ_d using $(X_l, Y_l) := (X_l, Y_l) \cup \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$.
  Calculate the state S' based on Eq. (2), and the reward r based on Eq. (3).
  Save the sample (S, a, S', r) into the replay buffer.
  for each training sample for actor and critic **do**
    Update the critic by optimizing Problem (6).
    Update the actor by optimizing Problem (4).
    Update the target actor and critic according to Eq. (7).
  end for
end for

3 Experiments

3.1 Data Sets and Evaluation Metrics

- **chestCT**[1] is a Computed Tomography (CT) data set for lung disease detection with four kinds of diseases, including pulmonary nodule, pulmonary cord, arteriosclerosis and calcification of lymph node. It contains 1,470 CT scans. We randomly pick up scans from them to construct training and testing data sets. After that, we take the regions with the ground-truth label to obtain samples, and then randomly pick up 3,500 samples as training samples and 3,500 samples as testing samples.
- The **Retinopathy** data set[2] contains 35,126 fundus images collected by different devices from different environments. Each fundus image is rated from 0 to 4 according to the presence and degree of diabetic retinopathy (DR), *i.e.*, no DR, mild, moderate, severe, and proliferative DR. We randomly pick up 2,230 images as training data and 2,230 images as testing data.

[1] https://tianchi.aliyun.com/competition/entrance/231724/introduction.
[2] https://www.kaggle.com/c/diabetic-retinopathy-detection/.

We adopt two performance metrics, *i.e.*, Macro F1 and Micro F1 scores, for evaluation. Taking the j-th class as the positive label while the other classes as the negative label, we can define TP_j, TN_j, FP_j and FN_j as the numbers of true positive, true negative, false positive and false negative, respectively. The two evaluation metrics are defined as Macro F1 $= \frac{1}{M}\sum_{j=1}^{M}\frac{2\cdot TP_j}{2\cdot TP_j+FN_j+FP_j}$, Micro F1 $= \frac{2\cdot\sum_{j=1}^{M} TP_j}{2\cdot\sum_{j=1}^{M} TP_j+\sum_{j=1}^{M} FN_j+\sum_{j=1}^{M} FP_j}$.

3.2 Experimental Settings

In the experiments, we compare our method with several active learning methods, including random selection (RANDOM), least confidence (LC) [11], margin sampling (MS) [10], entropy (EN) [12], and FUSION [17]. The FUSION method combines the three above mentioned criteria, *i.e.*, LC, MS and EN. In specific, FUSION selects top $\frac{K}{2}$ samples according to LC, MS and EN, respectively. After that, FUSION removes the replicate ones from the $\frac{3K}{2}$ samples, and randomly selects K samples from them to annotate. We also conduct a method named "ALL", which takes all the training data as labeled ones to train the model.

For the chestCT data set, we randomly select 5% samples of each class from the training set to initialize the network, and the rest are for the incremental learning process. In each epoch, we randomly select 5 samples from the unlabeled training set to annotate, and then add them into the labeled training data to update the classifier.

For the Retinopathy data set, we randomly select 10% images of each class from the training set to initialize the network, and the rest are for the incremental learning process. In each epoch, we randomly select 1 sample from the unlabeled training set to annotate, and then add them into the labeled training data to update the classifier.

All the methods are implemented on the PyTorch platform [9]. We use ResNet-50 [5] with 3D convolutional layers [15] as the architecture for chestCT, and ResNet-50 pretrained on ImageNet data set [1] to initialize the classifier for the Retinopathy data set. We adopt the SGD optimizer with the learning rate 0.0001 to train the classifier network. The actor network and the critic network have the same architecture, which consists of three fully connected layers. Both of them are trained using the Adam optimizer with the learning rate 0.001. We set the delay factor as $\gamma = 0.99$, the trade-off parameter as $\lambda = 0.005$, and the batch size as 16. For the DDPG algorithm, we adopt the noise exploration mechanism used in [18].

3.3 Results and Discussion

Tables 1 and 2 present the results on chestCT. Our proposed method DRLA outperforms the compared active learning methods. This demonstrates that compared with the hand-design strategies used in the compared methods, DRLA is able to learn a more effective strategy to select informative samples for improving the performance. Besides, to achieve 0.70 F1 scores, DRLA only needs around 40% labeled training data, while the other active learning method require around 68% labeled training data. This indicates that DRLA can reduce the need of labeled data.

Table 1. Macro F1 results on the chestCT data set.

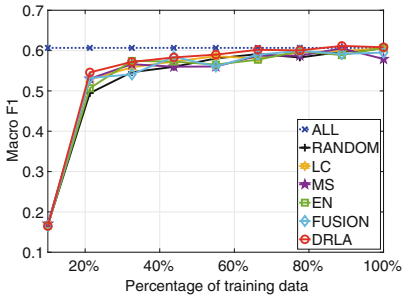
Percentage of training samples	26%	40%	54%	68%	82%	100%
ALL	–	–	–	–	–	0.7494
RANDOM	0.6530	0.6994	0.7014	0.7101	0.7125	0.7176
LC	0.6813	0.7193	0.7235	0.7383	0.7396	0.7389
MS	0.6431	0.6902	0.7007	0.7173	0.7182	0.7207
EN	0.6730	0.7194	0.7287	0.7353	0.7330	0.7348
FUSION	0.6733	0.6845	0.7112	0.7198	0.7190	0.7192
DRLA	**0.6869**	**0.7362**	**0.7419**	**0.7451**	**0.7481**	**0.7525**

Table 2. Micro F1 results on the chestCT data set.

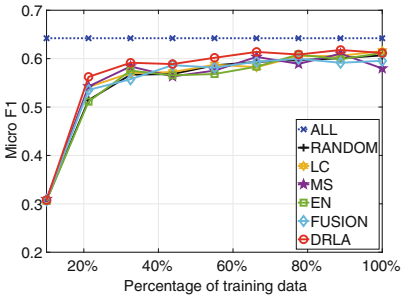
Percentage of training samples	26%	40%	54%	68%	82%	100%
ALL	–	–	–	–	–	0.7462
RANDOM	0.6574	0.7023	0.7029	0.7109	0.7131	0.7180
LC	0.6829	0.7197	0.7231	0.7363	0.7369	0.7363
MS	0.6449	0.6926	0.7011	0.7143	0.7169	0.7180
EN	0.6857	0.7200	0.7280	0.7329	0.7311	0.7343
FUSION	0.6914	0.6891	0.7137	0.7197	0.7191	0.7203
DRLA	**0.6940**	**0.7343**	**0.7409**	**0.7451**	**0.7480**	**0.7537**

Figure 2 shows the results of learning procedures on the two data sets, respectively. We observe that as the number of labeled training data increases, all the active learning methods can obtain better performance. Besides, after receiving 20% labeled training data, our proposed method DRLA consistently achieves the best or highly comparable performance compared with the other active learning methods. This further verifies that DRLA is able to select informative samples to improve the classification performance.

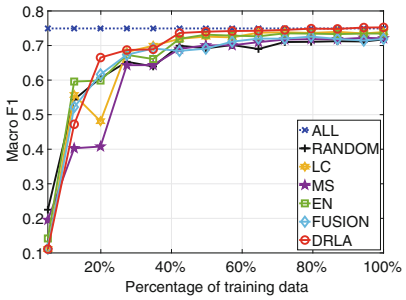
More experimental results could be found in Supplementary Material.



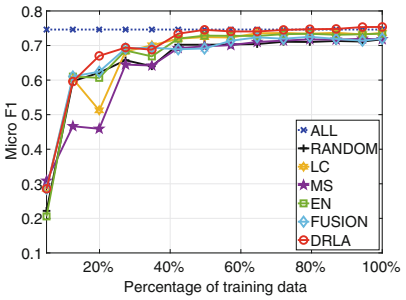
(a) Macro F1 score on the Retinopathy data set.



(b) Micro F1 score on the Retinopathy data set.



(c) Macro F1 score on chestCT.



(d) Micro F1 score on chestCT.

Fig. 2. Macro F1 and micro F1 results on the data sets.

4 Conclusion

In this paper, we propose a deep reinforcement active learning algorithm for medical image classification. To learn a dynamic strategy for active learning, we apply deep reinforcement learning to learn a policy to select samples for annotation, and employ deep deterministic policy gradient algorithm under the actor-critic paradigm to train the model. We conduct experiments on two medical image data sets to demonstrate the effectiveness of the proposed method.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
2. Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: International Conference on Machine Learning, pp. 1587–1596 (2018)
3. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Proceedings of International Conference on Machine Learning, pp. 1861–1870 (2018)

4. Hatamizadeh, A., et al.: Deep active lesion segmentation. In: International Workshop on Machine Learning in Medical Imaging, pp. 98–105 (2019)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
6. Lillicrap, T.P., et al.: Continuous control with deep reinforcement learning. In: Proceedings of International Conference on Learning Representations (2015)
7. Litjens, G., et al.: A survey on deep learning in medical image analysis. Med. Image Anal. **42**, 60–88 (2017)
8. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
9. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, pp. 8024–8035 (2019)
10. Scheffer, T., Decomain, C., Wrobel, S.: Active hidden Markov models for information extraction. In: Hoffmann, F., Hand, D.J., Adams, N., Fisher, D., Guimaraes, G. (eds.) IDA 2001. LNCS, vol. 2189, pp. 309–318. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44816-0_31
11. Settles, B.: Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences (2009)
12. Shannon, C.E.: A mathematical theory of communication. Bell Syst. Tech. J. **27**(3), 379–423 (1948)
13. Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. Annu. Rev. Biomed. Eng. **19**, 221–248 (2017)
14. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (2018)
15. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of IEEE International Conference on Computer Vision, pp. 4489–4497 (2015)
16. Tuia, D., Volpi, M., Copa, L., Kanevski, M., Munoz-Mari, J.: A survey of active learning algorithms for supervised remote sensing image classification. IEEE J. Sel. Top. Signal Process. **5**(3), 606–617 (2011)
17. Wang, K., Zhang, D., Li, Y., Zhang, R., Lin, L.: Cost-effective active learning for deep image classification. IEEE Trans. Circ. Syst. Video Technol. **27**(12), 2591–2600 (2016)
18. Wawrzynski, P.: Control policy with autocorrelated noise in reinforcement learning for robotics. Int. J. Mach. Learn. Comput. **5**(2), 91 (2015)
19. Yang, L., Zhang, Y., Chen, J., Zhang, S., Chen, D.Z.: Suggestive annotation: a deep active learning framework for biomedical image segmentation. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D.L., Duchesne, S. (eds.) MICCAI 2017. LNCS, vol. 10435, pp. 399–407. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66179-7_46