




# Prediction Stability as a Criterion in Active Learning

Junyu Liu<sup>1</sup>(✉) , Xiang Li<sup>2</sup>, Jiqiang Zhou<sup>2</sup>, and Jianxiong Shen<sup>3</sup>

<sup>1</sup> Graduate School of Informatics, Kyoto University, Kyoto, Japan  
liu.junyu.82w@st.kyoto-u.ac.jp

<sup>2</sup> Hikvision Research Institute, Hangzhou, China

<sup>3</sup> Institut de Robòtica i Informàtica Industrial, CSIC-UPC, Barcelona, Spain

**Abstract.** Recent breakthroughs made by deep learning rely heavily on a large number of annotated samples. To overcome this shortcoming, active learning is a possible solution. Besides the previous active learning algorithms that only adopted information after training, we propose a new class of methods named sequential-based method based on the information during training. A specific criterion of active learning called prediction stability is proposed to prove the feasibility of sequential-based methods. We design a toy model to explain the principle of our proposed method and pointed out a possible defect of the former uncertainty-based methods. Experiments are made on CIFAR-10 and CIFAR-100, and the results indicates that prediction stability was effective and works well on fewer-labeled datasets. Prediction stability reaches the accuracy of traditional acquisition functions like entropy on CIFAR-10, and notably outperformed them on CIFAR-100.

**Keywords:** Active learning · Classification · Prediction stability

## 1 Introduction

Recent breakthroughs made by deep learning heavily relied on Supervised Learning (SL) with large amount of annotated datasets [10, 12]. But in the practical applications, large amount of labels are expensive and time-consuming [13]. Lack of labels is an important obstacle to adopt SL methods. To achieve similar accuracy to SL with less labels, (pool-based) active learning (AL) [11] has become a possible solution. These strategies have succeeded in many realms such as image processing [18] and natural language processing(NLP) [17].

The goal of active learning is to select the least number of typical samples and train the model to reach the same accuracy as one trained on all the samples. It's not difficult to find out that the core of active learning methods is the strategy of sample selection called acquisition function. Most of the previous works belong to the pool-based method, which selects a subset of samples

---

Junyu Liu did this work during his internship at the Hikvision Research Institute.

© Springer Nature Switzerland AG 2020

I. Farkas et al. (Eds.): ICANN 2020, LNCS 12397, pp. 157–167, 2020.

[https://doi.org/10.1007/978-3-030-61616-8\\_13](https://doi.org/10.1007/978-3-030-61616-8_13)

after a whole training process on the existing labeled dataset and then goes on [3, 4, 7, 8, 16, 19]. Basing on the learning process of pool-based active learning, the samples selected are expected to be the ones with the most information. In many works, the selected samples were the most uncertain ones. The basic ideas included using confidence, max-entropy [16], mutual information [7], mean standard deviation [8] or variation-ratio [3] of samples as a measurement. Recent works of AL adopted strategies based on Bayesian Convolutional Neural Networks [4] and Generative Adversarial Nets (GAN) [19]. Although the principles of the networks were different from typical classification convolutional neural networks (CNN), the methods still generated or chose samples with the highest uncertainty. Another class of work selected samples by the expectation of model change. For instance, expected gradient length [15] choose samples expected to cause the largest gradients to the current model. After approximation of the algorithm, the selected samples were similar to adversarial examples [5]. Some works concentrate on exploring the typical samples of the whole dataset. For example, core-set [14] choose samples that are at the center of a neighbor area, and expect all the selected samples to cover the whole feature space.

Present active learning methods are different in strategy and implementation, but we can classify all the methods mentioned above as *spatial-based* ones. That is, although different methods concentrate on different parts of the AL process (prediction, model updating, etc.), the information took in to account all came from the prediction of the well-trained models before selection. The whole process was a flat one without information from the time course. Here we propose sequential-based methods, and as a verification of it, we propose a new criterion of sample selection in image classification called the *prediction stability*, which describes the oscillation of predictions across the epochs during training. Instead of starting from a well-trained model, this model also gathered information for the selection process while training the model. Among different epochs of a training process, the fluctuation of prediction on a sample is taken as the measure of uncertainty of the feature space around this sample. We designed a toy model to explain the principle of our proposed method and pointed out a possible defect of the former uncertainty-based methods. The results of our experiments also agreed with our assumption and prove the proposed method as an effective one.

The following parts of this paper are divided into 4 sections. The second section introduces the relation to prior work. The third is our methodology. The fourth section provides the experimental results. And the final part is the conclusion.

## 2 Relation to Prior Work

When comparing our proposed method with present AL algorithms mentioned in the introduction part, there are two major differences. First, our sequential-based method not only extracts features after training but also during the training process. Second, the previously proposed measures of the amount of information are

based on more apparent criteria including uncertainty, the influence on the model and typical samples. They care more about the scales of the final predictions, but prediction stability is a new criterion to catch the indirect information of relative prediction changes.

### 3 Methodology

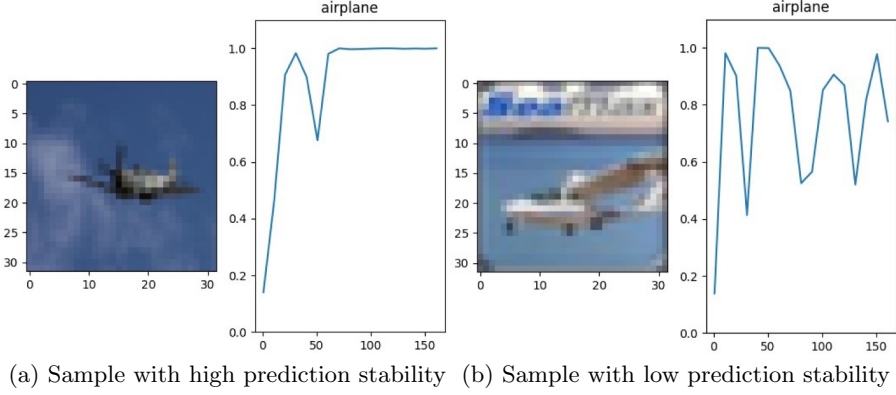
We can define the dataset of all samples as  $X = \{x_i | i = 1 \dots n\}$ , with  $X^L \subseteq X$  representing the labeled set containing  $n_l$  labels, and the complementary set  $X^U = X \setminus X^L$  is the set of unlabeled  $n_u$  samples. The budget of AL is defined as  $B$ . For pool-based active learning, after initialization, in each round of AL, the model will select  $b$  samples from  $X^U$  for annotation and put the set of them  $S \subseteq X^U$  into  $X^L$ , then the model is retrained on the new  $X^L$  set. This process repeats until the total number of selected samples reaches the budget. In previous works [7, 8, 16], the acquisition functions of subset  $S$  can be concluded as (1). In this equation,  $f(\cdot)$  is the feature extracting function, and  $g(\cdot)$  outputs the scores of samples.

$$S = \underset{S}{\operatorname{argmax}} \left[ \sum_{i=1}^b g(f(s_i)) \right], s_i \in S \quad (1)$$

The previous spatial-based methods mentioned in the introduction part concentrate on the quality of final predictions. All the innovations focus on the measurements of the final prediction. Different from this kind of methods, we propose sequential-based methods that make use of the information during training. We'll prove the necessity of information during training later. Defining number of epochs in training as  $N_e$ , and  $f_n(\cdot)$  as the  $f(\cdot)$  function in  $n$ -th epoch, the acquisition function can be rewritten as Eq. 2.

$$S = \underset{S}{\operatorname{argmax}} \left[ \sum_{i=1}^b g(f_1(s_i), \dots, f_{N_e}(s_i)) \right], s_i \in S \quad (2)$$

As an application of sequential-based methods, we propose *prediction stability*, a new criterion of selecting the subset  $S$  in active learning. For implementation, we also adopt the common CNN model as the feature extractor and classifier. An important distinction with former spatial-based methods is that this criterion focuses not on the final scales of outputs, but the fluctuation of scales during training. As Fig. 1 shows, looking through the whole training process, features of samples like (a) tend to be relatively stable, but other samples like (b) oscillates from the beginning to the end. Instinct speculation is that samples like Fig. 1(b) should be selected for labeling. To do quantitative analysis, we test some common-used measures of fluctuation of data and choose the average variance of each unit of outputs across epochs as the measure of prediction stability. The diagrams in Fig. 1 also shows that, due to under-fitting, the former epochs of training are definitely to violate severely. Therefore only epochs in the later training process should be included in the calculation. After the experiment,



**Fig. 1.** Example of samples with different prediction stability during training. The horizontal axis in the right diagrams is the index of epochs, and the vertical axis shows the scale of a unit of the output vector.

we find that the selected epochs are actually in the over-fitting area, which is relatively stable. Also, considering the time complexity, only several epochs are chosen in the end.

The definition of prediction stability can be written as Eq. 3:

$$g(x) = \sum_{c=1}^C \text{var}(f_{e_1}(x)_c, \dots, f_{e_n}(x)_c) \quad (3)$$

Where  $C$  is the length of the output vectors by  $f(\cdot)$ ,  $f(x)_c$  is the  $c$ -th unit of output  $f(x)$ , and  $\{e_1, e_2, \dots, e_n\} = E$  is the set of index of selected epochs. The whole framework of prediction stability is displayed in Algorithm 1.

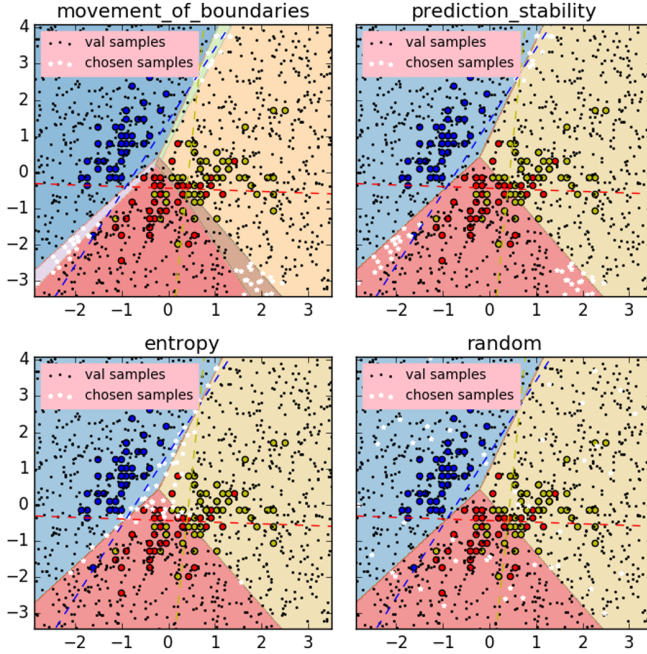
---

**Algorithm 1.** Prediction Stability

---

**Input:** CNN model  $M$ , dataset  $X = \{x_i | i = 1 \dots n\}$ , initial sampling number  $k$ , number of epochs per training process  $N_e$ , set of index of selected epochs  $E$ , budget  $B$ , subset of samples selected each round  $S$ .

- 1: Generate first  $k$  samples randomly, and produce labels for them;
  - 2: **repeat**
  - 3:   **for**  $i = 1 \rightarrow N_e$  **do**
  - 4:     Train the model  $M$  on labeled samples;
  - 5:     **if**  $i \in E$  **then**
  - 6:       Predict outputs  $P_i$  of  $M$  on unlabeled set.
  - 7:     Get prediction stability of each image along selected epochs;
  - 8:     Select top  $|S|$  samples with lowest prediction stability, generate labels and put them into labeled sample pool;
  - 9: **until** Reach the budget  $B$
-



**Fig. 2.** Toy example applying different acquisition functions to iris dataset. The axes are the first two features (sepal length, sepal width) of the samples. Plains of different colors reflect the decision boundaries of each category (setosa, versicolor and virginica). Dashed lines are the one-against-all classifiers trained on the original iris dataset. Colored points are original samples of different categories in the iris dataset. The validation set consists of samples randomly generated around the original samples. White stars are samples selected from the validation set by different acquisition functions. The top-left image shows the changes of decision boundaries during training. The small patches are areas that belonged to different categories during training, and white stars are the same as those selected by prediction stability.

To help explain the principle of prediction stability, we made a toy example on the iris dataset [1, 2], as demonstrated in Fig. 2. The main body of this example came from the sample code in the document of sklearn<sup>1</sup>. This task aimed to do a 3-class classification (setosa, versicolor and virginica). We adopted SVM-based multi-class stochastic gradient descent (SGD) linear-classifiers. For the convenience of visualization, the task was done on the first two features (sepal length, sepal width) of the iris dataset. The axes in Fig. 2 are the first two features of the samples, and therefore the plains are the feature space of the samples, which is the set of all the possible values of the samples. One class (setosa, blue points in Fig. 2) is linearly separable from the other two; the latter (versicolor and virginica) are not linearly separable from each other.

<sup>1</sup> [https://scikit-learn.org/stable/auto\\_examples/linear\\_model/plot\\_sgd\\_iris.html](https://scikit-learn.org/stable/auto_examples/linear_model/plot_sgd_iris.html).

The distribution of selected samples reflected the principle of different acquisition functions. Randomly selected samples roughly obeyed uniform distribution. Samples selected by entropy were around the T-crossing of three categories and were surrounded by samples of the training set. But for prediction stability, the samples were gathering at the outer side of the decision boundaries. During the training process, the decision boundaries between different categories were swinging roughly around the crossing point of the boundaries. The sway was within a sector-shaped area, like the small patches in the top-left image of Fig. 2. Because the boundary was a straight line, the outer part of the sector was influenced most by the change of the boundary, which was the area where the selected samples (white stars in the top images of Fig. 2) located.

The uncertainty-based acquisition functions tended to make finer-grained borders near the crossing of multi-categories, where the predicted probabilities of a sample to belong to different categories were close to each other. But as shown in this example, these methods sacrificed the accuracy at the feature space far from the training set. Rather, the samples selected by prediction stability, which located at the boundaries with the least number of training samples, were the ones the classifiers were of least certainty because no information about this area was obtained from the training data.

## 4 Experimental Results

### 4.1 Implementation Details

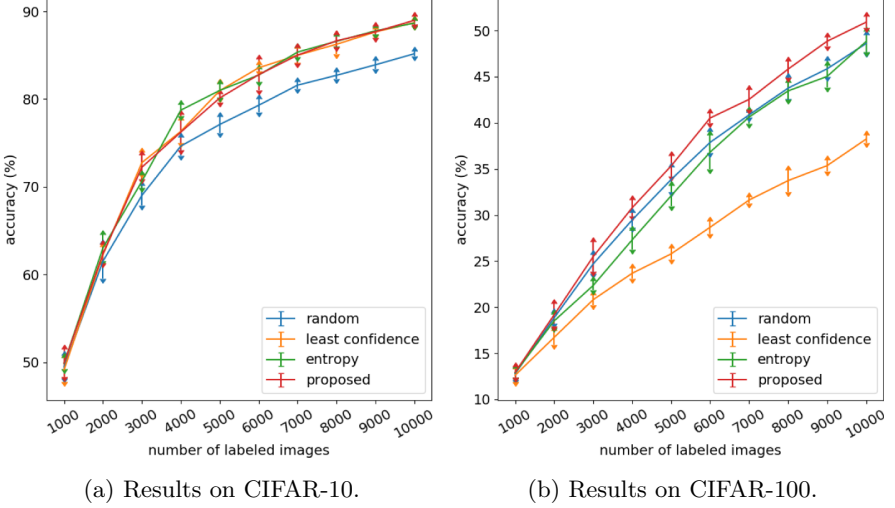
**Datasets.** CIFAR-10 and CIFAR-100 [9] were used for the evaluation of our proposed method. The samples of the two datasets are all  $32 \times 32$  small image patches. Each dataset contains 50000 training samples and 10000 test samples respectively. The training and test samples are equally distributed into all categories. But the difference is that CIFAR-10 only has 10 classes, and CIFAR-100 contains 100 classes. Therefore, the number of samples in each class of CIFAR-10 is 10 times that of CIFAR-100.

**Architecture Details.** As for the model  $M$  for feature extraction, we employed ResNet-18 [6], which is a relatively deep architecture, and a popular choice among recent works on AL. This network mainly consists of the first convolution layer and the following 4 residual blocks. The implementation was based on an open-source framework<sup>2</sup>. The softmax outputs of the network, which were the final score vector of categories, were chosen as the output in this work.

All the models in this work were implemented on an NVIDIA TITAN Xp GPU. During training, the batch size was 128, and 164 epochs were utilized in each training process. In our experiments, for each dataset, a subset containing 1000 samples was selected for the first training process. Since biases of numbers among different classes in the initially labeled dataset might heavily influence

<sup>2</sup> <https://github.com/bearpaw/pytorch-classification.git>.

the selection after the first training process, an equal number of samples were randomly selected from each category of the dataset in the beginning. 1000 samples were selected and labeled after each training process, and the final size of the labeled dataset was 10000. To overcome the influence of random factors and get objective results, we generated 6 sets of initially labeled samples at first and did the first training processes of all the methods on the same 6 datasets. The final results of each method were the average of the six trails.



**Fig. 3.** Results on CIFAR-10 and CIFAR-100 (1 standard deviation; across 6 trials).

## 4.2 CIFAR-10

The results on CIFAR-10 is displayed by Fig. 3(a). Because the output features were the probability of all classes, entropy and least confidence (ranking by the largest score among categories for each sample) measure were calculated on the outputs directly. For the calculation of prediction stability, we finally selected 5 epochs starting from the last one with an interval of 5.

$$e_i = N_e - (i - 1) \times interval, i = 1, 2, 3, 4, 5 \quad (4)$$

The results showed that although information about the value of outputs was not included directly, the proposed prediction stability method still overwhelmed random selection, and achieved similar performance with acquisition functions like entropy and least confidence on CIFAR-10.

### 4.3 CIFAR-100

The performance of each method on CIFAR-100 is exhibited in Fig. 3(b). To perform prediction stability on CIFAR-100, the interval of epoch selection was set to 1. Previous works in the introduction part hardly reported their results on this dataset, but our results on CIFAR-100 showed different tendencies with CIFAR-10. Entropy and least confidence, especially the least confidence, suffered from deterioration of performance. The accuracy of both acquisition functions was lower than random selection. But our proposed method proves better performance and outperforms random selection.

We believe the better performance of our proposed model on CIFAR-100 than CIFAR-10 is caused by the number of training samples in the feature space. The major difference between the two datasets is CIFAR-100 has fewer samples in each class, which means the feature space of each class is more sparse and has fewer labels to distinguish the border. As shown in the toy model, the uncertainty-based method tends to make a fine-grained border around the labeled samples and ignore the unknown area of the feature space. Therefore it worked worse when there were less labeled samples. The result of the two datasets showed that prediction stability has a better capacity for fewer-labeled datasets.

### 4.4 Ablation Study

**Measure of Prediction Stability.** Experiments were made to test the performance of different measures of prediction stability, as displayed in Fig. 4. We tested the absolute increase among features of different epochs. This measure is represented by Eq. 5.

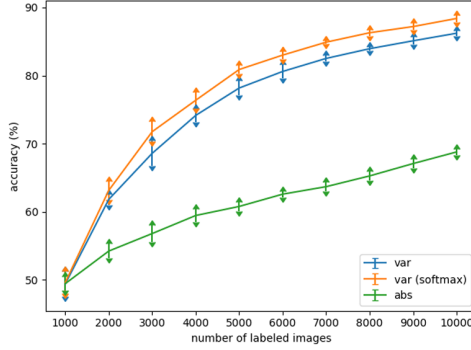
$$F(x) = \sum_{i=2}^{|E|} |f_{e_i}(x) - f_{e_{i-1}}(x)| \quad (5)$$

Taking the absolute increase as a measure led to a nearly 40% drop in performance. It suggests that it's not the tendency of change, but the distribution of output, that determines the performance of prediction stability.

Also, we tested the result of taking variance as the acquisition function but removed the softmax calculation. A deterioration of the result could also be observed clearly, which proved the necessity of the softmax layer's function of normalization. The output features of different samples were transferred into comparable probabilities, and therefore the differences in absolute scales of output features didn't influence the variances.

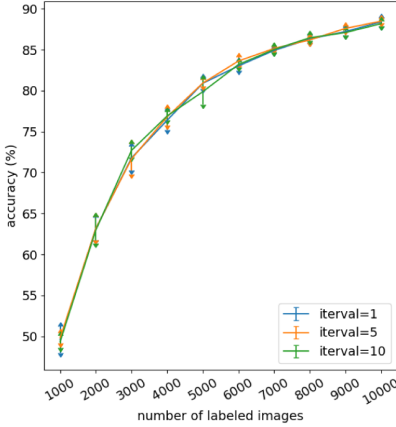
**Interval of Epoch Selection.** Experiments were made to test the influence of epoch selection on the results of prediction stability. The epoch selection process was based on Eq. 4. Results on the two datasets are different, as exhibited in Fig. 5. Although accuracy was slightly better when the interval equaled 5,



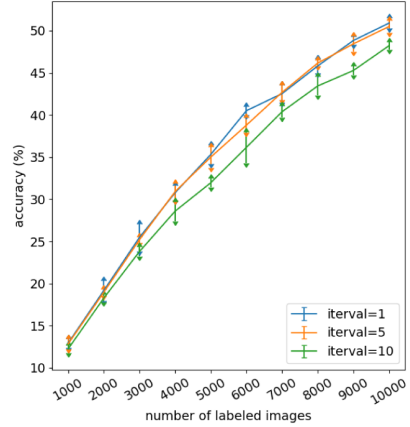


**Fig. 4.** Results on CIFAR-10 with different measure of prediction stability (1 standard deviation; across 6 trials).

CIFAR-10 was not sensitive to interval change. But in CIFAR-100, the accuracy declined as the interval of epoch increased. This happened may because the models trained on CIFAR-100 over-fitted at later epochs than CIFAR-10. That is, when the interval was 10, the result of some epochs of CIFAR-100 was still not in the relatively stable state and caused a decrease in accuracy.



(a) Result on CIFAR-10



(b) Result on CIFAR-100

**Fig. 5.** Results on different intervals of prediction stability (1 standard deviation; across 6 trials).

## 5 Conclusion

In this paper, we proposed a new class of AL methods named sequential-based AL method. A new criterion, prediction stability was proposed as an application

of the sequential-based method. We designed an example to demonstrate the principle of prediction stability and unveiled that the previous uncertainty-based methods tend to ignore unknown areas in the feature space. Testing results of prediction stability on CIFAR-10 and CIFAR-100 proved the feasibility of the sequential-based method class.

As for the future work, we will focus on fusing our proposed method with uncertainty-based AL methods, because the information extracted by two kinds of methods are complementary.

## References

1. Dua, D., Graff, C.: UCI machine learning repository (2017). <http://archive.ics.uci.edu/ml>
2. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Ann. Eugenics* **7**(2), 179–188 (1936)
3. Freeman, L.C.: *Elementary Applied Statistics: For Students in Behavioral Science*. Wiley, New York (1965)
4. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1183–1192. JMLR.org (2017)
5. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. *Computer Science* (2014)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
7. Houthby, N., Huszár, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. *arXiv preprint [arXiv:1112.5745](https://arxiv.org/abs/1112.5745)* (2011)
8. Kampffmeyer, M., Salberg, A.B., Jenssen, R.: Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1–9 (2016)
9. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Technical report. Citeseer (2009)
10. Law, H., Deng, J.: CornerNet: detecting objects as paired keypoints. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) *Computer Vision – ECCV 2018*. LNCS, vol. 11218, pp. 765–781. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01264-9\\_45](https://doi.org/10.1007/978-3-030-01264-9_45)
11. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Croft, B.W., van Rijsbergen, C.J. (eds.) *SIGIR 1994*, pp. 3–12. Springer, London (1994). [https://doi.org/10.1007/978-1-4471-2099-5\\_1](https://doi.org/10.1007/978-1-4471-2099-5_1)
12. Liu, J., et al.: An original neural network for pulmonary tuberculosis diagnosis in radiographs. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) *ICANN 2018*. LNCS, vol. 11140, pp. 158–166. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01421-6\\_16](https://doi.org/10.1007/978-3-030-01421-6_16)
13. Qu, Z., Liu, J., Liu, Y., Guan, Q., Yang, C., Zhang, Y.: OriNet: a regression system for latent fingerprint orientation field extraction. In: Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I. (eds.) *ICANN 2018*. LNCS, vol. 11141, pp. 436–446. Springer, Cham (2018). [https://doi.org/10.1007/978-3-030-01424-7\\_43](https://doi.org/10.1007/978-3-030-01424-7_43)

14. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. arXiv preprint [arXiv:1708.00489](https://arxiv.org/abs/1708.00489) (2017)
15. Settles, B., Craven, M.: An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 1070–1079. Association for Computational Linguistics (2008)
16. Shannon, C.E.: A mathematical theory of communication. *Bell Syst. Tech. J.* **27**(3), 379–423 (1948)
17. Tanguy, L., Tulechki, N., Urieli, A., Hermann, E., Raynal, C.: Natural language processing for aviation safety reports: from classification to interactive analysis. *Comput. Ind.* **78**, 80–95 (2016)
18. Zhou, Z., Shin, J., Zhang, L., Gurudu, S., Gotway, M., Liang, J.: Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7340–7351 (2017)
19. Zhu, J.J., Bento, J.: Generative adversarial active learning. arXiv preprint [arXiv:1702.07956](https://arxiv.org/abs/1702.07956) (2017)