



# MVCAL: Multi View Clustering for Active Learning

Yi Fan<sup>✉</sup>, Biao Jiang, Di Chen, and Yu-Bin Yang<sup>✉</sup>

State Key Laboratory for Novel Software Technology, Nanjing University,  
Nanjing 210023, China

fanyiplus@smail.nju.edu.cn, yangyubin@nju.edu.cn

**Abstract.** Various active learning methods with ingenious sampling strategies have been proposed to solve the lack of labeled samples in supervised learning, but most are designed for specific tasks. In this paper, we propose a simple but task-agnostic active sampling method. We introduce ‘multi-view clustering module’ to extract multiple feature maps at different levels for unsupervised clustering. According to clustering distribution, we calculate consistency, representativeness and stability to guide sampling and training. Among them, consistency measures the similarity between clustering results of two views, representativeness reflects the distance between a sample and the corresponding cluster center, and stability reflects the model’s feature representation and recognition ability for the same sample. Our method does not depend on the specific network, and can be constructed as a two-stage sampling module to supplement the existing sampling algorithm. Experiments results on image classification and object detection tasks show that our method can further enhance the effect of active learning on the basis of baseline methods.

**Keywords:** Computer Vision · Active Learning · Image Classification · Object Detection · Clustering

## 1 Introduction

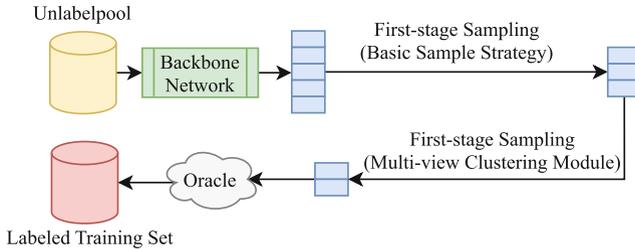
In computer vision, complex networks and tasks require high-quality labeled data. The increasing labeling overhead has hampered access to this data to some extent. Active learning (AL) aims to use as few labeled samples, which may contain more information, as possible to obtain the same effect as fully supervised training. In a classic pool-based active learning scenario with a limited training set, numerous unlabeled samples form a candidate sample pool (called *unlabeledpool*). The model continuously selects critical samples from the unlabeledpool through a sampling strategy for annotation to expand the training set, so as to optimize the current model iteratively. Existing AL basically follows the above

---

This work is funded by the Natural Science Foundation of China under Grant No. 62176119.

framework by designing different active sampling strategies. For example, the classical Least Confidence (LC), Margin, and Entropy algorithms in classification tasks [9, 14, 16] measure the prediction uncertainty of the current model to guide sampling. In object detection tasks, there are both the sampling method for classification branch only [24], which is transferred from the classification task, and the method using the stability predicted by regression box [11] as the sampling index from the perspective of regression branch.

However, the AL sampling strategies in the above methods depend on specific tasks. Although they can be adapted to other tasks after appropriate modifications, they often do not work as well on the new task. In recent years, researchers began to explore and design a task-agnostic AL method, hoping to provide a general sampling strategy. For example, [28] proposes a task-agnostic loss prediction module to predict sample loss directly to guide sampling. [22] proposes a method of active sampling by measuring data distribution called *Coreset*. Unfortunately, the sampling standard of the above methods still has some one-sidedness. [28] only considers the feedback of the model and ignores the characteristics of the data, while [22] only considers the feature distribution of the data on the macro level. As deep learning methods, they do not make full use of powerful feature representation ability of neural networks.



**Fig. 1.** A basic AL architecture with MVC module. For unlabeled samples, the backbone network is used to obtain features. Then the Basic Sample Strategy and Multi-view Clustering Module are used for First-stage Sampling and Second-stage Sampling respectively. After the two stages of sampling, the obtained samples are labeled as Labeled Training Set.

For most computer vision tasks, backbone networks are used to extract the features of input images. The process of analyzing these features can be independent of specific tasks. Inspired by this, we propose a plug-and-play Multi-view Clustering (MVC) module that can be conveniently embedded in the backbone network to improve the performance of deep learning models. A basic AL architecture with our MVC module is shown in Fig. 1. We believe it can be applied to any task that uses a deep network. In the first-stage sampling, the candidate sampling set is obtained through the existing basic sampling strategy. Then, the MVC module is adopted for the second-stage sampling to further screen out the key samples. Meanwhile, in the training process, we also conduct the MVC of

batch data and then calculate the overall consistency of the batch data distribution as consistency loss to optimize the feature extraction capability of the model.

Since the MVC module does not depend on specific task types and can supplement the effect of existing active sampling methods, we believe it is a convenient and effective task-agnostic method. Besides, our method has strong scalability and can be combined with several existing AL methods. Experiments show that in image classification and object detection tasks, taking three existing active sampling methods as the baseline, the model effect of adding the MVC module exceeds the baseline methods.

Our contributions can be summarized as follows:

- We propose a novel active sampling method with the MVC module, which is task-agnostic and can be directly embedded in any tasks with deep networks.
- We use MVC as a supplement to the existing active sampling method. The existing sampling method is used for first-stage sampling, followed by second-stage MVC sampling, which can further improve the effect of active learning.
- We evaluate the proposed method with two learning tasks including image classification and object detection. Experimental results show that the proposed method significantly outperforms baseline methods.

## 2 Related Work

Active learning (AL) has been studied for decades and many excellent methods have emerged [1, 4]. According to different application scenarios, AL can be divided into pool-based, stream-based and Query Synthesis active learning [7]. However, this division method can not clearly reflect the characteristics of different active sampling strategies. Therefore, we can also divide them into uncertainty-based, distribution-based, expected model change and metaheuristic active learning based on different sampling strategies.

### 2.1 Uncertainty-Based Methods

Uncertainty sampling is one of the most classical sampling strategies in AL. In multi-classification tasks, the uncertainty can be calculated by Least Confidence [14], Margin [9], and Entropy [16] algorithms. In addition, SVMs [26] can also define the distance from the decision boundary as uncertainty. Recently, uncertainty-based methods have been applied to many tasks such as video moment retrieval [8] and image segmentation [10].

### 2.2 Distribution-Based Methods

Uncertainty-based methods measure the information of samples from the perspective of models, while distribution-based methods mine representative samples in the overall data distribution of unlabeled sample pool. The typical approach is to conduct unsupervised clustering of data [17], and then calculate representative and diversity scores according to the distance between samples, so as

to sample diversity but overall representative samples. [6] calculate the distance between the sample and its nearest neighbors, and then samples that can better represent the data features of the current fixed region will be sampled. Coreset [5] further defines active learning as the generation process of candidate sample set, that is, by sampling key samples to form a subset that can represent the characteristics of the whole unlabeled pool, training on this subset can obtain the same task effect.

### 2.3 Expect Model Change Methods

Deep learning models are usually optimized by gradient descent and minimize prediction losses during training, which inspires researchers to design sampling algorithms from the perspective of model adjustment. [20] predict the gradient descent degree in the training process and samples with larger values will be sampled. LLAL [28] predicts the task loss in the training process, which will be used to guide active sampling. Since expect model change methods can effectively save the consumption of model training, it has been applied in diverse fields such as [13, 23, 25].

### 2.4 Metaheuristic Methods

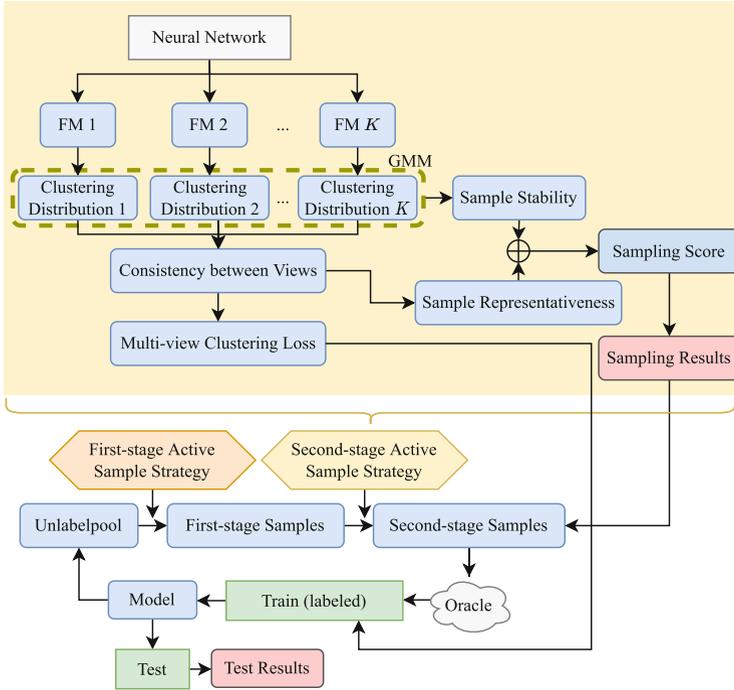
Metaheuristic methods have gained significant attention in training various neural networks due to their ability to optimize complex problems by exploring the problem space efficiently. [19] proposes the distributed wound treatment optimization method for training CNN models. [12] proposes the neuroevolutionary approach to control complex multicoordinate interrelated plants. [29] introduces the concept of simulated annulment in convolutional neural networks, and uses metaheuristics to remove unnecessary connections in the network, simplifying the model and improving its efficiency. [3] proposes a novel convolutional neural network model based on the beetle antennae search optimization algorithm for computerized tomography diagnosis. These studies offer promising solutions for enhancing the performance and efficiency of neural network models in a variety of domains.

The above methods define the key samples from different perspectives, and then derive a variety of sampling strategies. However, single sampling strategy can not avoid the problem of one-sided sampling. In addition, while task-agnostic methods already exist, they all fail to effectively utilize the powerful feature representation ability of neural networks, and room for improvement still exists.

## 3 Method

In neural networks, the feature map (FM) is a universal feature representation layer available by the combination of convolutions. Different combinations generate various FMs reflecting the characteristics of samples at different views. Based on this, we propose multi-view clustering active learning (MVCAL). The overall

framework is shown in Fig. 2. The core of MVCAL, the MVC module, performs clustering by extracting multiple FMs corresponding to multiple views. Based on the clustering results, the representativeness and stability of samples will be calculated as the sampling strategy. Meanwhile, consistency will be calculated as part of the loss function to improve the feature extraction ability of the model in the training process.



**Fig. 2.** Our proposed MVCAL. On the bottom is the two-stage sampling process. On the top is the concrete structure of the MVC module: Firstly, FM of different levels (corresponding to multiple views) are extracted from the backbone network as the input of unsupervised clustering. Then, sample stability and sample representativeness are calculated respectively according to the clustering results of multiple views, which are integrated as sampling scores.

The MVC module is task-agnostic and can be easily embedded into existing networks. Therefore, we expand the traditional AL into a two-stage AL based on MVC. By combining MVC with other different sampling strategies, the evaluation index of AL can be more comprehensive, and critical samples can be sampled simultaneously from the perspective of model expectation and data distribution.

### 3.1 FM Clustering in Different Views

In the MVC module, we first extract multiple FMs for each sample. Then we use one of the most classic clustering models, Gaussian mixture model (GMM) [27], to estimate the distribution and conduct clustering in each view. In GMM, the distribution of samples is expressed by

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^K \alpha_k \phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k), \quad (1)$$

where  $\mathbf{x}$  denotes the input sample,  $K$  denotes the number of Gaussian model,  $\phi(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k)$  denotes the  $k$ -th Gaussian distribution with mean  $\boldsymbol{\mu}_k$  and variance  $\boldsymbol{\sigma}_k$ ,  $\alpha_k$  denotes the weight coefficient, i.e., the probability that the observation sample belongs to the  $k$ -th Gaussian model. So for a total of  $U$  views, we obtain  $U$  clustering distributions,  $p_1(\mathbf{x}|\boldsymbol{\theta}_1)$ ,  $p_2(\mathbf{x}|\boldsymbol{\theta}_2)$ , ...,  $p_U(\mathbf{x}|\boldsymbol{\theta}_U)$ . According to these clustering distributions, we can divide each sample into different classes in each view.

### 3.2 Consistency Between Views

A well-trained model should be able to extract common features at different levels, so as to make the clustering distribution of each view as consistent as possible. Therefore, the consistency between two views is used to measure the similarity between their clustering results. In this paper, we choose a simple but effective algorithm, Rand statistic [21], to calculate the consistency.

Denote the clustering label of a sample  $\mathbf{x}_i$  in view  $V_m$  as  $l_{V_m}(\mathbf{x}_i)$ . Then for all the sample pairs in two different views,  $V_m$  and  $V_n$ , we get  $s(s-1)/2$  sample pairs ( $s$  denotes sample size),  $(\mathbf{x}_i, \mathbf{x}_j)$  ( $i \neq j$ ). In these sample pairs, we use  $s_p$  to denote those who satisfying both  $l_{V_m}(\mathbf{x}_i) = l_{V_m}(\mathbf{x}_j)$  and  $l_{V_n}(\mathbf{x}_i) = l_{V_n}(\mathbf{x}_j)$ , or satisfying both  $l_{V_m}(\mathbf{x}_i) \neq l_{V_m}(\mathbf{x}_j)$  and  $l_{V_n}(\mathbf{x}_i) \neq l_{V_n}(\mathbf{x}_j)$ , and  $s_n$  to denote other sample pairs. Then the consistency between  $V_m$  and  $V_n$  can be calculated as

$$R(V_m, V_n) = \|s_p\| / (s_p + s_n), \quad (2)$$

where  $\|\cdot\|$  denotes the number of element in a set.

### 3.3 Training Strategy

In the training process, the parameters of networks are optimized to perform the specific task better and extract FM better simultaneously. So the loss function is mainly composed of two parts, task loss (TL),  $\mathcal{L}_{\text{task}}$ , and multi-view clustering loss (MVCL),  $\mathcal{L}_{\text{MVC}}$ . TL is the loss of a specific task, such as the cross entropy of classification [18]. MVCL is related to consistency, and can be calculated as

$$\mathcal{L}_{\text{MVC}} = \sum_{m=1}^U \sum_{n=1}^U (1 - R(V_m, V_n)). \quad (3)$$

Finally, the total loss is calculated as

$$\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda \cdot \mathcal{L}_{\text{MVC}}, \quad (4)$$

where  $\lambda$  denotes the weight between two items.

### 3.4 Sampling Strategy

In the sampling process, we use representativeness and stability as indicators to measure the quality of a sample.

Representativeness is a commonly used sampling strategy in the field of AL. [17] proposed clustering as the data preprocessing process and active sampling through representativeness. After that, AGPR [27] also proposes a method of sampling by pixel comparison of the whole image. Different from the existing methods, our method selects the distribution with the highest consistency among all the views to calculate the representativeness, where the consistency of view  $V_m$  can be calculated as

$$\text{Cons}(V_m) = \sum_{n=1, n \neq m}^U R(V_m, V_n). \quad (5)$$

Then the representativeness of sample  $\mathbf{x}_i$  is just the probability density of the selected distribution, expressed as

$$\begin{aligned} \text{Rep}(\mathbf{x}_i) &= p_o(\mathbf{x}_i | \boldsymbol{\theta}_o), \\ \text{where } o &= \arg \max_m p_m(\mathbf{x} | \boldsymbol{\theta}_m). \end{aligned} \quad (6)$$

By this design, the representativeness can reflect the distance between the sample and the center of the fixed cluster. The larger its value is, the closer it is to the cluster center, i.e., it has better representativeness.

Unlike sample representativeness represents a class of samples with key common features, sample stability measures the stability of distribution in various views, which reflects the model's feature representation and recognition ability for the same sample. Assume the set of samples owning the same cluster label as  $\mathbf{x}_i$  in  $V_m$  is  $S_{V_m}(\mathbf{x}_i)$ , then the stability of  $\mathbf{x}_i$  in  $V_m$  and  $V_n$  is

$$\text{Stab}_{mn}(\mathbf{x}_i) = \frac{S_{V_m}(\mathbf{x}_i) \cap S_{V_n}(\mathbf{x}_i)}{S_{V_m}(\mathbf{x}_i) \cup S_{V_n}(\mathbf{x}_i)}. \quad (7)$$

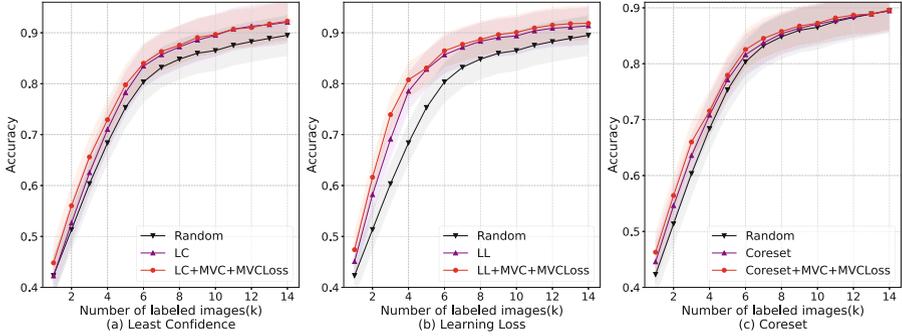
Finally, the stability of sample  $\mathbf{x}_i$  can be defined as

$$\text{Stab}(\mathbf{x}_i) = \sum_{m=1}^U \sum_{n=m+1}^U (\text{Stab}_{mn}(\mathbf{x}_i)). \quad (8)$$

Now we can calculate the score of sample  $\mathbf{x}_i$  as

$$S(\mathbf{x}_i) = \text{Rep}(\mathbf{x}_i) + \text{Stab}(\mathbf{x}_i) \quad (9)$$

to decide which samples should be sampled.



**Fig. 3.** Results for image classification on CIFAR-10. LC, LL and Coreset mean the one-stage AL using the least confidence, learning loss and coreset strategy, respectively. +MVC means adding the MVC as the second-stage sampling. +MVCLoss means adding  $\mathcal{L}_{MVC}$  in the training process. We took the average of the three experiments as the final result.

## 4 Experiments

### 4.1 Image Classification

For image classification, we use Resnet-18 as the backbone network and CIFAR-10 as dataset. Due to the numerous samples in the unlabeled pool at the beginning, it is expensive to use all unlabeled samples for prediction. Therefore, we follow the practice in [2]. In each round of AL, we first select 10 000 images as candidate sets in a random way. Then in the first stage of MVCAL, 2000 images are sampled out of 10 000, and in the second stage, 1000 images are further sampled from 2000.

**Experimental Setup.** The number of clustering centers is specified as 10. In Resnet-18, FM of the last four convolution layers are taken as views, and their sizes are  $64 \times 32 \times 32$ ,  $128 \times 16 \times 16$ ,  $256 \times 8 \times 8$ ,  $512 \times 4 \times 4$  respectively. The learning rate is set to  $10^{-3}$ , and we train 200 epochs each iteration. We use Adam optimizer with  $\alpha_1 = 0.9$  and  $\alpha_2 = 0.99$ .

We use LC [14], learning loss (LL) [11] and Coreset [22] as baseline methods respectively. The results are shown in Fig. 3. Results show that all the methods have better results than the random baseline. After adding MVC and MVCLoss, further improvements are achieved. For the LL-based method, the improvement of MVC and MVCLoss is the most obvious. This is in line with expectations because LL does not evaluate the distribution of the data, and the sampling index is single. For LC-based, the results are similar. For Coreset-based, the improvement of MVC and MVCLoss is not apparent. This may be because the coreset method itself is distribution-based, and the MVC model also measures the distribution characteristics. Nevertheless, our method can still bring improvement, which shows that our method is better than coreset in mining sample distribution information.

The improvement of ‘+MVC’ methods in the first half of the training cycle is the most obvious in the whole training cycle, which shows that our method can effectively accelerate the convergence speed of the model, and can also obtain a weak final effect improvement on LL-based. These demonstrate after adopting our method, we can obtain better classification results.

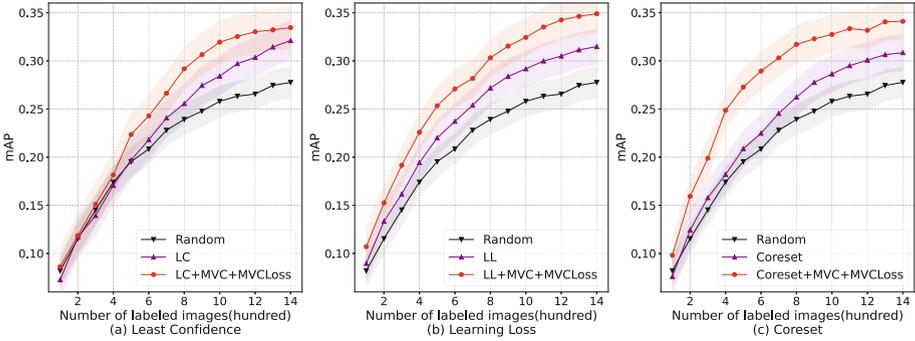


Fig. 4. Results for object detection on Pascal VOC07. Similar to Fig. 3.

### 4.2 Object Detection

We conduct experiments on object detection to verify the excellent task-agnostic of our method. We use SSD [15] as the backbone network and Pascal VOC2007 as dataset. Since Pascal VOC2007 does not contain too many samples, we no longer build candidate sets for sampling, but actively sample 200 images from the unlabelpool each round.

**Experiment Setup.** The number of clustering centers is set to 6 and 20 respectively. In SSD, FM for MVC is extracted from layer 4\_3, 7, 8\_2, 9\_2, 10\_2, 11\_2 [15], same with [28]. We use Adam optimizer with  $\alpha_1 = 0.9$  and  $\alpha_2 = 0.99$ . Each round of AL trains 6 epochs, of which the learning rate of the first 4 epochs is set to  $10^{-3}$  and that of the last 2 epochs is set to  $5^{-4}$ .

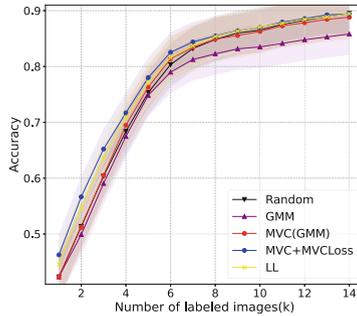
We also use LC [14], LL [11] and Coreset [22] as baseline methods respectively. The results are shown in Fig. 4. It can be seen that compared with baseline methods(one-stage), our method shows a significant performance improvement. This indicates that for complex visual tasks such as object detection, the existing one-stage sampling method is ineffective in assessing images’ sampling information with multiple candidate instances. Our method measures the stability of clustering results under multiple views by adding the MVC module. The more instances there are, the more significant the impact on the consistency of clustering, so it can achieve a significant improvement.

In addition, our method shows almost the highest performance for all AL cycles. In the last cycle, our method achieves mAPs of 0.3346 (LC-based), 0.3489

(LL-based) and 0.3431 (Coreset-based). The results are 3.34%, 10.8% and 11.7% higher than the LC-based, LL-based and Coreset-based methods respectively.

### 4.3 Ablation Study

We conduct ablation experiments on CIFAR-10 by removing parts of our method. Results are shown in Fig. 5. We can see that the effect of single-view clustering is even worse than that of random strategy, which indicates that in simple tasks, single-view clustering pays too much attention to representative samples and ignores the impact of others. In contrast, inherent random search characteristics can better avoid overfitting in a random strategy. The MVC sampling, which measures sample stability and cluster consistency simultaneously, can improve the results to a certain extent compared with random. After adding  $\mathcal{L}_{MVCL}$ , the results can be further improved, and even better than the existing typical task-agnostic method LL. This fully proves the effectiveness of our proposed MVC module.



**Fig. 5.** Ablation study. ‘Random’ means random sampling. ‘GMM’ means sampling only according to representativeness. ‘MVC’ means sampling according to representativeness and stability. Both GMM and MVC do not contain  $\mathcal{L}_{MVCL}$ .

## 5 Conclusion and Further Work

In this paper, we propose a task-agnostic active sampling module, MVC, and further embed it into the existing AL methods to construct a two-stage AL framework. The MVC module plays a critical role in both training and sampling. In the training process, it is used to calculate the overall clustering consistency of batch data and optimize the parameters of networks. In the sampling process, it calculates the stability and representativeness of samples to make up for the deficiency in the one-stage sampling. Extensive experiments on image classification and object detection tasks show that our method outperforms three traditional

ALs. This proves that our method is suitable for different tasks and different baseline AL methods.

In the future, we will verify our method in more tasks such as natural language processing and speech recognition. Moreover, we acknowledge the high computational cost associated with the clustering methods used in our method. To address this limitation, we will explore and develop more efficient clustering methods that can maintain or improve the performance while reducing computational overhead. By optimizing the clustering process, we aim to enhance the scalability and practicality of our method, making it more accessible and feasible for real-world AL scenarios.

## References

1. Bachman, P., Sordoni, A., Trischler, A.: Learning algorithms for active learning. In: International Conference on Machine Learning, pp. 301–310. PMLR (2017)
2. Beluch, W.H., Genewein, T., Nürnberger, A., Köhler, J.M.: The power of ensembles for active learning in image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9368–9377 (2018)
3. Chen, D., Li, X., Li, S.: A novel convolutional neural network model based on beetle antennae search optimization algorithm for computerized tomography diagnosis. *IEEE Trans. Neural Netw. Learn. Syst.* **34**, 1418–1429 (2021)
4. Gal, Y., Islam, R., Ghahramani, Z.: Deep Bayesian active learning with image data. In: International Conference on Machine Learning, pp. 1183–1192. PMLR (2017)
5. Guo, C., Zhao, B., Bai, Y.: DeepCore: a comprehensive library for coresets selection in deep learning. In: Strauss, C., Cuzzocrea, A., Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) Database and Expert Systems Applications, DEXA 2022, Part I. Lecture Notes in Computer Science, vol. 13426, pp. 181–195. Springer, Cham (2022). [https://doi.org/10.1007/978-3-031-12423-5\\_14](https://doi.org/10.1007/978-3-031-12423-5_14)
6. Hasan, M., Roy-Chowdhury, A.K.: Context aware active learning of activity recognition models. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4543–4551 (2015)
7. He, T., Jin, X., Ding, G., Yi, L., Yan, C.: Towards better uncertainty sampling: active learning with multiple views for deep convolutional neural network. In: 2019 IEEE International Conference on Multimedia and Expo (ICME), pp. 1360–1365. IEEE (2019)
8. Ji, W., et al.: Are binary annotations sufficient? Video moment retrieval via hierarchical uncertainty-based active learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 23013–23022 (2023)
9. Joshi, A.J., Porikli, F., Papanikolopoulos, N.: Multi-class active learning for image classification. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2372–2379. IEEE (2009)
10. Kang, C.J., Peter, W.C.H., Siang, T.P., Jian, T.T., Zhaofeng, L., Yu-Hsing, W.: An active learning framework featured Monte Carlo dropout strategy for deep learning-based semantic segmentation of concrete cracks from images. *Struct. Health Monit.* **22**, 3320–3337 (2023). <https://doi.org/10.1177/14759217221150376>
11. Kao, C.-C., Lee, T.-Y., Sen, P., Liu, M.-Y.: Localization-aware active learning for object detection. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV

2018. LNCS, vol. 11366, pp. 506–522. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-20876-9\\_32](https://doi.org/10.1007/978-3-030-20876-9_32)
12. Kondratenko, Y., Kozlov, O., Gerasin, O.: Neuroevolutionary approach to control of complex multicoordinate interrelated plants. *Int. J. Comput.* **18**(4), 502–514 (2019)
  13. Kosugi, S., Yamasaki, T.: Crowd-powered photo enhancement featuring an active learning based local filter. *IEEE Trans. Circ. Syst. Video Technol.* **33**, 3145–3158 (2023)
  14. Lewis, D.D.: A sequential algorithm for training text classifiers: corrigendum and additional data. *ACM SIGIR Forum* **29**, 13–19 (1995)
  15. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46448-0\\_2](https://doi.org/10.1007/978-3-319-46448-0_2)
  16. Luo, W., Schwing, A., Urtasun, R.: Latent structured active learning. In: *Advances in Neural Information Processing Systems*, vol. 26 (2013)
  17. Nguyen, H.T., Smeulders, A.: Active learning using pre-clustering. In: *Proceedings of the Twenty-First International Conference on Machine Learning*, p. 79 (2004)
  18. Nielsen, M.A.: *Neural Networks and Deep Learning*, vol. 25. Determination Press, San Francisco (2015)
  19. Ponce, H., Moya-Albor, E., Brieva, J.: Towards the distributed wound treatment optimization method for training CNN models: analysis on the MNIST dataset. In: *2023 IEEE 15th International Symposium on Autonomous Decentralized System (ISADS)*, pp. 1–6. IEEE (2023)
  20. Roy, N., McCallum, A.: Toward optimal active learning through Monte Carlo estimation of error reduction. In: *ICML, Williamstown*, vol. 2, pp. 441–448 (2001)
  21. Schütze, H., Manning, C.D., Raghavan, P.: *Introduction to Information Retrieval*, vol. 39. Cambridge University Press, Cambridge (2008)
  22. Sener, O., Savarese, S.: Active learning for convolutional neural networks: a core-set approach. In: *International Conference on Learning Representations* (2018)
  23. Sivaraman, G., Jackson, N.E.: Coarse-grained density functional theory predictions via deep kernel learning. *J. Chem. Theor. Comput.* **18**(2), 1129–1141 (2022)
  24. Sivaraman, S., Trivedi, M.M.: Active learning for on-road vehicle detection: a comparative study. *Mach. Vis. Appl.* **25**, 599–611 (2014)
  25. Ureel, Y., et al.: Active learning-based exploration of the catalytic pyrolysis of plastic waste. *Fuel* **328**, 125340 (2022)
  26. Vijayanarasimhan, S., Grauman, K.: Large-scale live active learning: training object detectors with crawled data and crowds. *Int. J. Comput. Vis.* **108**, 97–114 (2014)
  27. Wang, H., Gao, X., Zhang, K., Li, J.: Single-image super-resolution using active-sampling gaussian process regression. *IEEE Trans. Image Process.* **25**(2), 935–948 (2015)
  28. Yoo, D., Kweon, I.S.: Learning loss for active learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 93–102 (2019)
  29. Zhou, C.: Simulated annulling in convolutional neural network. In: *2022 2nd International Symposium on Artificial Intelligence and its Application on Media (ISA-IAM)*, pp. 38–42. IEEE (2022)