CSE584 Homework #1

Xinzhang Xiong

Paper #1: A Survey of Deep Active Learning

1. What problem does this paper try to solve, i.e., its motivation

This survey paper addresses the inefficiency of traditional active learning(AL) in the context of modern deep learning(DL), which requires substantial labeled data to train effectively. Traditional AL systems struggle with high-dimensional data typical of deep learning applications. The motivation is to explore whether AL can effectively reduce the costs of data annotation while retaining the powerful learning capabilities ofDL.

2. How does it solve the problem?

The paper proposes a comprehensive survey of existing DeepAL methods and introduces a formal classification of these methods. It emphasizes strategies for integrating deep learning models with active learning techniques to select informative samples for labeling, aiming to optimize the performance of deep learning models with fewer labeled instances.

3. A list of novelties/contributions

First comprehensive survey in the field of DeepAL.

Systematic classification method for existing DeepAL works and discusses various strategies used in the field.

Analysis of challenges and solutions of combining DL with AL, particularly the issues of model uncertainty and the efficiency of query strategies.

Future developments in DeepAL, including the improvement of query strategies and the integration of unsupervised learning methods.

4. What do you think are the downsides of the work?

The integration of AL and DL often leads to increased computational complexity, which might limit the practical deployment of these models in resource-constrained environments.

The success of active learning strategies heavily depends on the initial quality and distribution of the data, which might not always be optimal.

Paper #2: State-Relabeling Adversarial Active Learning

1. What problem does this paper try to solve, i.e., its motivation

This paper addresses the challenge in active learning (AL) of efficiently selecting the most informative unlabeled samples to be labeled. Previous AL methods may not efficiently handle the complexity and high dimensionality of data used in deep learning, leading to inefficiencies in sample selection and annotation efforts.

2. How does it solve the problem?

The proposed solution, the state relabeling adversarial active learning model (SRAAL) model, integrates adversarial learning with a novel state relabeling approach. This model uses a unified representation generator and a labeled/unlabeled state discriminator:

- **Representation Generator:** Combines unsupervised image reconstruction and supervised target learning to generate a robust data representation embedding both the data's semantic and its labeled/unlabeled state.
- **State Discriminator:** Employs an online uncertainty indicator to assign a continuous importance score to each unlabeled sample, rather than a binary labeled/unlabeled state, thereby enabling more nuanced selection of samples for labeling.

3. A list of novelties/contributions

Introduces a method to assess and continuously update the importance of unlabeled samples, which enhances the active learning process by prioritizing more informative samples.

Integrates adversarial learning into active learning to better utilize both labeled and unlabeled data for model training.

Proposes a k-center based approach for initializing the labeled pool, which provides a diverse and representative initial set of samples for more effective learning.

4. What do you think are the downsides of the work?

The model might face scalability issues, especially in scenarios involving extremely large datasets, given the complexities involved in the adversarial training and state relabeling processes.

Paper #3: A DEEP ACTIVE LEARNING SYSTEM FOR SPECIES IDENTIFICATION AND COUNTING IN CAMERA TRAP IMAGES

1. What problem does this paper try to solve, i.e., its motivation

The paper focuses on the laborious and costly process of manually reviewing millions of images from camera trap surveys to extract useful data about wildlife populations. This manual review process often leads to delays and data losses due to resource limitations. The motivation is to leverage deep learning and active learning to dramatically reduce the manual effort required while maintaining high accuracy in identifying and counting species captured in camera trap images.

2. How does it solve the problem?

The authors propose an active learning system that combines machine intelligence with human review to optimize the labeling process. The system uses a pre-trained object detection model to identify and count animals in images, significantly reducing the number of images requiring human review. It incorporates transfer learning to adapt models trained on large datasets to new, smaller datasets, and employs active learning strategies to intelligently select the most informative images for manual labeling, thus maximizing the efficiency of the labeling process.

3. A list of novelties/contributions

The method reduces over 99.5% reduction in manual labeling effort with the same performance.

Combines transfer learning and active learning to efficiently use pre-existing large datasets for training models on new, smaller datasets.

Utilizes pre-trained models to identify and count animals, reducing the dependency on manual labeling and improving initial accuracy before any active learning cycle.

4. What do you think are the downsides of the work?

The effectiveness of the system heavily depends on the quality and relevance of the pre-trained models used.

The system's reliance on existing datasets may propagate biases present in those datasets and could raise data privacy concerns if sensitive locations are involved.

A Survey of Deep Active Learning

PENGZHEN REN* and YUN XIAO*, Northwest University
XIAOJUN CHANG, RMIT University
PO-YAO HUANG, Carnegie Mellon University
ZHIHUI LI†, Qilu University of Technology (Shandong Academy of Sciences)
BRIJ B. GUPTA, National Institute of Technology Kurukshetra, India
XIAOJIANG CHEN and XIN WANG, Northwest University

Active learning (AL) attempts to maximize a model's performance gain while annotating the fewest samples possible. Deep learning (DL) is greedy for data and requires a large amount of data supply to optimize a massive number of parameters if the model is to learn how to extract high-quality features. In recent years, due to the rapid development of internet technology, we have entered an era of information abundance characterized by massive amounts of available data. As a result, DL has attracted significant attention from researchers and has been rapidly developed. Compared with DL, however, researchers have a relatively low interest in AL. This is mainly because before the rise of DL, traditional machine learning requires relatively few labeled samples, meaning that early AL is rarely according the value it deserves. Although DL has made breakthroughs in various fields, most of this success is due to a large number of publicly available annotated datasets. However, the acquisition of a large number of high-quality annotated datasets consumes a lot of manpower, making it unfeasible in fields that require high levels of expertise (such as speech recognition, information extraction, medical images, etc.). Therefore, AL is gradually coming to receive the attention it is due.

It is therefore natural to investigate whether AL can be used to reduce the cost of sample annotation while retaining the powerful learning capabilities of DL. As a result of such investigations, deep active learning (DeepAL) has emerged. Although research on this topic is quite abundant, there has not yet been a comprehensive survey of DeepAL-related works; accordingly, this article aims to fill this gap. We provide a formal classification method for the existing work, along with a comprehensive and systematic overview. In addition, we also analyze and summarize the development of DeepAL from an application perspective. Finally, we discuss the confusion and problems associated with DeepAL and provide some possible development directions.

CCS Concepts: • **Computing methodologies** → **Machine learning algorithms**.

Additional Key Words and Phrases: Deep Learning, Active Learning, Deep Active Learning.

ACM Reference Format:
Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2021. A Survey of Deep Active Learning. 40 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Both deep learning (DL) and active learning (AL) are a subfield of machine learning. DL is also called representation learning [50]. It originates from the study of artificial neural networks and realizes the automatic extraction of data features. DL has strong learning capabilities due to its complex structure, but this also means that DL requires a large number of labeled samples to complete the corresponding training. With the release of a large number of large-scale data sets with annotations

*Both authors contributed equally to this research.
†Corresponding author.

Authors' addresses: Pengzhen Ren, pzhren@foxmail.com; Yun Xiao, yxiao@nwu.edu.cn, Northwest University; Xiaojun Chang, cxj273@gmail.com, RMIT University; Po-Yao Huang, Carnegie Mellon University; Zhihui Li, Qilu University of Technology (Shandong Academy of Sciences); Brij B. Gupta, National Institute of Technology Kurukshetra, India; Xiaojiang Chen; Xin Wang, Northwest University.

and the continuous improvement of computer computing power, DL-related research has ushered in large development opportunities. Compared with traditional machine learning algorithms, DL has an absolute advantage in performance in most application areas. AL focuses on the study of data sets, and it is also known as query learning [193]. AL assumes that different samples in the same data set have different values for the update of the current model, and tries to select the samples with the highest value to construct the training set. Then, the corresponding learning task is completed with the smallest annotation cost. Both DL and AL have important applications in the machine learning community. Due to their excellent characteristics, they have attracted widespread research interest in recent years. More specifically, DL has achieved unprecedented breakthroughs in various challenging tasks; however, this is largely due to the publication of massive labeled datasets [21, 120]. Therefore, DL is limited by the high cost of sample labeling in some professional fields that require rich knowledge. In comparison, an effective AL algorithm can theoretically achieve exponential acceleration in labeling efficiency [17]. This large potential saving in labeling costs is a fascinating development. However, the classic AL algorithm also finds it difficult to handle high-dimensional data [221]. Therefore, the combination of DL and AL, referred to as DeepAL, is expected to achieve superior results. DeepAL has been widely utilized in various fields, including image recognition [56, 72, 82, 98], text classification [190, 251], visual question answering [134] and object detection [4, 63, 170], etc. Although a rich variety of related work has been published, DeepAL still lacks a unified classification framework. To fill this gap, in this article, we will provide a comprehensive overview of the existing DeepAL related work [1], along with a formal classification method. The contributions of this survey are summarized as follows:

- As far as we know, this is the first comprehensive review work in the field of deep active learning.
- We analyze the challenges of combining active learning and deep learning, and systematically summarize and categorize existing DeepAL-related work for these challenges.
- We conduct a comprehensive and detailed analysis of DeepAL-related applications in various fields and future directions.

Next, we first briefly review the development status of DL and AL in their respective fields. Subsequently, in Section 2, the necessity and challenges of combining DL and AL are explicated. In Section 3, we conduct a comprehensive and systematic summary and discussion of the various strategies used in DeepAL. In Section 4, we review various applications of DeepAL in detail. In Section 5, we conduct a comprehensive discussion on the future direction of DeepAL. Finally, in Section 6, we make a summary and conclusion of this survey.

1.1 Deep Learning

DL attempts to build appropriate models by simulating the structure of the human brain. The McCulloch-Pitts (MCP) model proposed in 1943 by [65] is regarded as the beginning of modern DL. Subsequently, in 1986, [180] introduced backpropagation into the optimization of neural networks, which laid the foundation for the subsequent rapid development of DL. In the same year, Recurrent Neural Networks (RNNs) [105] were first proposed. In 1998, the LeNet [128] network made its first appearance, representing one of the earliest uses of deep neural networks (DNN). However, these pioneering early works were limited by the computing resources available at the time and did not

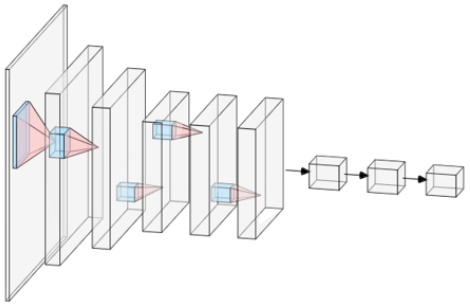
[1]We search about 270 related papers on DBLP using "deep active learning" as the keyword. We review the relevance of these papers to DeepAL one by one, eliminate irrelevant (just containing a few keywords) or information missing papers, and manually add some papers that do not contain these keywords but use DeepAL-related methods or relate to our current discussion. Finally, the survey references are constructed. The latest paper is updated to November 2020. The references include 103 conference papers, 153 journal papers, 3 books [62, 183, 221], 1 research report [193], and 1 dissertation [260]. There are 28 unpublished papers.

receive as much attention and investigation as they should have [126]. In 2006, Deep Belief Networks (DBNs) [91] were proposed and used to explore a deeper range of networks, which prompted the name of neural networks as DL. AlexNet [120] is considered the first CNN deep learning model, which greatly improves the image classification results on large-scale data sets (such as ImageNet). In the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)-2012 competition [49], the AlexNet [120] won the championship in the top-5 test error rate by nearly 10% ahead of the second place. AlexNet uses the ReLUs (Rectified Linear Units) [150] activation function to effectively suppress the gradient disappearance problem, while the use of multiple GPUs greatly improves the training speed of the model. Subsequently, DL began to win championships in various competitions and demonstrated very competitive results in many fields, such as visual data processing, natural language processing, speech processing, and many other well-known applications [240, 241]. From the perspective of automation, the emergence of DL has transformed the manual design of features [42, 139] in machine learning to facilitate automatic extraction [87, 203]. It is precisely because of this powerful automatic feature extraction capability that DL has demonstrated such unprecedented advantages in many fields. After decades of development, the research work related to DL is very rich. In Fig.1a, we present a standard deep learning model example: convolutional neural network (CNN) [127, 179]. Based on this approach, similar CNNs are applied to various image processing tasks. In addition, RNNs and GANs (Generative Adversarial Networks) [182] are also widely utilized. Beginning in 2017, DL gradually shifted from the initial feature extraction automation to the automation of model architecture design [16, 174, 262]; however, this still has a long way to go.

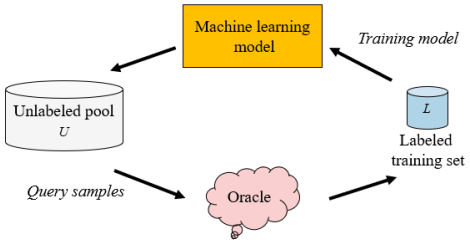
Thanks to the publication of a large number of existing annotation datasets [21, 120], in recent years, DL has made breakthroughs in various fields including machine translation [5, 18, 217, 231], speech recognition [152, 159, 169, 189], and image classification [89, 144, 158, 239]. However, this comes at the cost of a large number of manually labeled datasets, and DL has a strong greedy attribute to the data. While, in the real world, obtaining a large number of unlabeled datasets is relatively simple, the manual labeling of datasets comes at a high cost; this is particularly true for those fields where labeling requires a high degree of professional knowledge [94, 207]. For example, the labeling and description of lung lesion images of COVID-19 patients requires experienced clinicians to complete, and it is clearly impractical to demand that such professionals complete a large amount of medical image labeling. Similar fields also include speech recognition [1, 260], medical imaging [94, 129, 151, 243], recommender systems [2, 37], information extraction [22], satellite remote sensing [135] and robotics [8, 31, 32, 216, 258], machine translation [25, 164] and text classification [190, 251], etc. Therefore, a way of maximizing the performance gain of the model when annotating a small number of samples is urgently required.

1.2 Active Learning

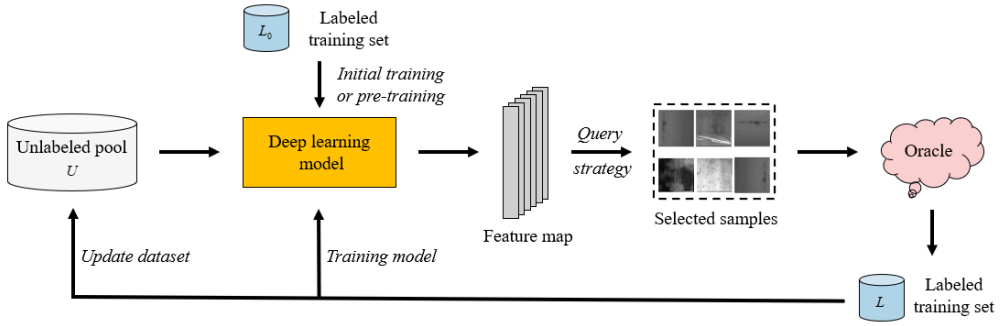
AL is just such a method dedicated to studying how to obtain as many performance gains as possible by labeling as few samples as possible. More specifically, it aims to select the most useful samples from the unlabeled dataset and hand it over to the oracle (e.g., human annotator) for labeling, to reduce the cost of labeling as much as possible while still maintaining performance. AL approaches can be divided into membership query synthesis [10, 113], stream-based selective sampling [41, 118] and pool-based [131] AL from application scenarios [193]. Membership query synthesis means that the learner can request to query the label of any unlabeled sample in the input space, including the sample generated by the learner. Moreover, the key difference between stream-based selective sampling and pool-based sampling is that the former makes an independent judgment on whether each sample in the data stream needs to query the labels of unlabeled samples, while the latter chooses the best query sample based on the evaluation and ranking of the entire dataset. Related research on stream-based selective sampling is mainly aimed at the application scenarios of small



(a) Structure diagram of convolutional neural network.



(b) The pool-based active learning cycle.



(c) A typical example of deep active learning.

Fig. 1. Comparison of typical architectures of DL, AL, and DeepAL. (a) A common DL model: Convolutional Neural Network. (b)The pool-based AL cycle: Use the query strategy to query the sample in the unlabeled pool U and hand it over to the oracle for labeling, then add the queried sample to the labeled training dataset L and train, and then use the newly learned knowledge for the next round of querying. Repeat this process until the label budget is exhausted or the pre-defined termination conditions are reached. (c) A typical example of DeepAL: The parameters θ of the DL model are initialized or pre-trained on the labeled training set L_0, and the samples of the unlabeled pool U are used to extract features through the DL model. Select samples based on the corresponding query strategy, and query the label in querying to form a new label training set L, then train the DL model on L, and update U at the same time. Repeat this process until the label budget is exhausted or the pre-defined termination conditions are reached (see Section 3.4 for stopping strategy details).

mobile devices that require timeliness, because these small devices often have limited storage and computing capabilities. The more common pool-based sampling strategy in the paper related to AL research is more suitable for large devices with sufficient computing and storage resources. In Fig.1b, we illustrate the framework diagram of the pool-based active learning cycle. In the initial state, we can randomly select one or more samples from the unlabeled pool U, give this sample to the oracle query label to get the labeled dataset L, and then train the model on L using supervised learning. Next, we use this new knowledge to select the next sample to be queried, add the newly queried sample to L, and then conduct training. This process is repeated until the label budget is exhausted or the pre-defined termination conditions are reached (see Section 3.4 for stopping strategy details).

It is different from DL by using manual or automatic methods to design models with high-performance feature extraction capabilities. AL starts with datasets, primarily through the design of elaborate query rules to select the best samples from unlabeled datasets and query their labels, in an attempt to reduce the labeling cost to the greatest extent possible. Therefore, the design of query rules is crucial to the performance of AL methods. Related research is also quite rich. For example, in a given set of unlabeled datasets, the main query strategies include the uncertainty-based approach [19, 106, 131, 173, 196, 222], diversity-based approach [23, 72, 84, 153] and expected model change [69, 177, 195]. In addition, many works have also studied hybrid query strategies [14, 200, 244, 255], taking into account the uncertainty and diversity of query samples, and attempting to find a balance between these two strategies. Because separate sampling based on uncertainty often results in sampling bias [26, 44], the currently selected sample is not representative of the distribution of unlabeled datasets. On the other hand, considering only strategies that promote diversity in sampling may lead to increased labeling costs, as may be a considerable number of samples with low information content will consequently be selected. More classic query strategies are examined in [194]. Although there is a substantial body of existing AL-related research, AL still faces the problem of expanding to high-dimensional data (e.g., images, text, and video, etc.) [221]; thus, most AL works tend to concentrate on low-dimensional problems [90, 221]. In addition, AL often queries high-value samples based on features extracted in advance and does not have the ability to extract features.

2 THE NECESSITY AND CHALLENGE OF COMBINING DL AND AL

DL has a strong learning capability in the context of high-dimensional data processing and automatic feature extraction, while AL has significant potential to effectively reduce labeling costs. Therefore, an obvious approach is to combine DL and AL, as this will greatly expand their application potential. This combined approach, referred to as DeepAL, was proposed by considering the complementary advantages of the two methods, and researchers have high expectations for the results of studies in this field. However, although AL-related research on query strategy is quite rich, it is still quite difficult to apply this strategy directly to DL. This is mainly due to:

- **Model uncertainty in Deep Learning.** The query strategy based on uncertainty is an important direction of AL research. In classification tasks, although DL can use the softmax layer to obtain the probability distribution of the label, the facts show that they are too confident. The SR (Softmax Response) [229] of the final output is unreliable as a measure of confidence, and the performance of this method will thus be even worse than that of random sampling [227].
- **Insufficient data for labeled samples.** AL often relies on a small amount of labeled sample data to learn and update the model, while DL is often very greedy for data [92]. The labeled training samples provided by the classic AL method thus insufficient to support the training of traditional DL. In addition, the one-by-one sample query method commonly used in AL is also not applicable in the DL context [255].
- **Processing pipeline inconsistency.** The processing pipelines of AL and DL are inconsistent. Most AL algorithms focus primarily on the training of classifiers, and the various query strategies utilized are largely based on fixed feature representations. In DL, however, feature learning and classifier training are jointly optimized. Only fine-tuning the DL models in the AL framework, or treating them as two separate problems, may thus cause divergent issues [229].

To address the first problem, some researchers have applied Bayesian deep learning [70] to deal with the high-dimensional mini-batch samples with fewer queries in the AL context [72, 115,

165, 223], thereby effectively alleviating the problem of the DL model being too confident about the output results. To solve the problem of insufficient labelled sample data, researchers have considered using generative networks for data augmentation [223] or assigning pseudo-labels to high-confidence samples to expand the labeled training set [229]. Some researchers have also used labeled and unlabeled datasets to combine supervised and semisupervised training across AL cycles [96, 202]. In addition, the empirical research in [192] shows that the previous heuristic-based AL [193] query strategy is invalid when it is applied to DL in batch settings; therefore, for the one-by-one query strategy in classic AL, many researchers focus on the improvement of the batch sample query strategy [14, 79, 115, 255], taking both the amount of information and the diversity of batch samples into account. Furthermore, to deal with the pipeline inconsistency problem, researchers have considered modifying the combined framework of AL and DL to make the proposed DeepAL model as general as possible, an approach that can be extended to various application fields. This is of great significance to the promotion of DeepAL. For example, [245] embeds the idea of AL into DL and consequently proposes a task-independent architecture design.

3 DEEP ACTIVE LEARNING

In this section, we will provide a comprehensive and systematic overview of DeepAL-related works. Fig.1c illustrates a typical example of DeepAL model architecture. The parameters θ of the deep learning model are initialized or pre-trained on the labeled training set L_0, while the samples of the unlabeled pool U are used to extract features through the deep learning model. The next steps are to select samples based on the corresponding query strategy, and query the label in the oracle to form a new label training set L, then train the deep learning model on L and update U at the same time. This process is repeated until the label budget is exhausted or the predefined termination conditions are reached (see Section 3.4 for stopping strategy details). From the DeepAL framework example in Fig.1c, we can roughly divide the DeepAL framework into two parts: namely, the AL query strategy on the unlabeled dataset and the DL model training method. These will be discussed and summarized in the following Section 3.1 and 3.2 respectively. Next, we will discuss the efforts made by DeepAL on the generalization of the model in Section 3.3. Finally, we briefly discuss the stopping strategy in DeepAL in Section 3.4.

3.1 Query Strategy Optimization in DeepAL

In the pool-based method, we define $U^n = \{\mathcal{X}, \mathcal{Y}\}$ as an unlabeled dataset with n samples; here, $\mathcal{X}$ is the sample space, $\mathcal{Y}$ is the label space, and $P(x, y)$ is a potential distribution, where $x \in \mathcal{X}, y \in \mathcal{Y}$. $L^m = \{X, Y\}$ is the current labeled training set with m samples, where $\mathrm{x} \in X, \mathrm{y} \in Y$. Under the standard supervision environment of DeepAL, our main goal is to design a query strategy Q, $U^n \xrightarrow{Q} L^m$, using the deep model $f \in \mathcal{F}, f : \mathcal{X} \rightarrow \mathcal{Y}$. The optimization problem of DeepAL in a supervised environment can be expressed as follows:

$$\arg\min_{L^m \subseteq U^n, (\mathrm{x},\mathrm{y}) \in L^m, (x,y) \in U^n} \mathbb{E}_{(x,y)}[\ell(f(\mathrm{x}), \mathrm{y})], \tag{1}$$

where $\ell(\cdot) \in \mathcal{R}^+$ is the given loss equation, and we expect that $m \ll n$. Our goal is to make m as small as possible while ensuring a predetermined level of accuracy. Therefore, the query strategy Q in DeepAL is crucial to reduce the labeling cost. Next, we will conduct a comprehensive and systematic review of DeepAL's query strategy from the following five aspects.

- *Batch Mode DeepAL (BMDAL).* The batch-based query strategy is the foundation of DeepAL. The one-by-one sample query strategy in traditional AL is inefficient and not applicable to DeepAL, so it is replaced by batch-based query strategy.

- *Uncertainty-based and Hybrid Query Strategies.* Uncertainty-based query strategy refers to the model based on sample uncertainty ranking to select the sample to be queried. The greater the uncertainty of the sample, the easier it is to be selected. However, this is likely to ignore the relationship between samples. Therefore, the method that considers multiple sample attributes is called the hybrid query strategy.
- *Deep Bayesian Active Learning (DBAL).* Active learning based on Bayesian convolutional neural network [70] is called deep Bayesian active learning.
- *Density-based Methods.* The density-based method is a query strategy that attempts to find a core subset [161] representing the distribution of the entire dataset from the perspective of the dataset to reduce the cost of annotation.
- *Automated Design of DeepAL.* Automated design of DeepAL refers to a method that uses automated methods to design AL query strategies or DL models that have an important impact on DeepAL performance.

3.1.1 Batch Mode DeepAL (BMDAL). The main difference between DeepAL and classical AL is that DeepAL uses batch-based sample querying. In traditional AL, most algorithms use a one-by-one query method, which leads to frequent training of the learning model but little change in the training data. The training set obtained by this query method is not only inefficient in the training of the DL model, but can also easily lead to overfitting. Therefore, it is necessary to investigate BMDAL in more depth. In the context of BMDAL, at each acquisition step, we score a batch of candidate unlabeled data samples $\mathcal{B} = \{x_1, x_2, ..., x_b\} \subseteq U$ based on the acquisition function a used and the deep model $f_\theta(L)$ trained on L, to select a new batch of data samples $\mathcal{B}^* = \{x_1^*, x_2^*, ..., x_b^*\}$. This problem can be formulated as follows:

$$\mathcal{B}^* = \underset{\mathcal{B} \subseteq U}{\arg\max}\, a_{batch}(\mathcal{B}, f_\theta(L)), \tag{2}$$

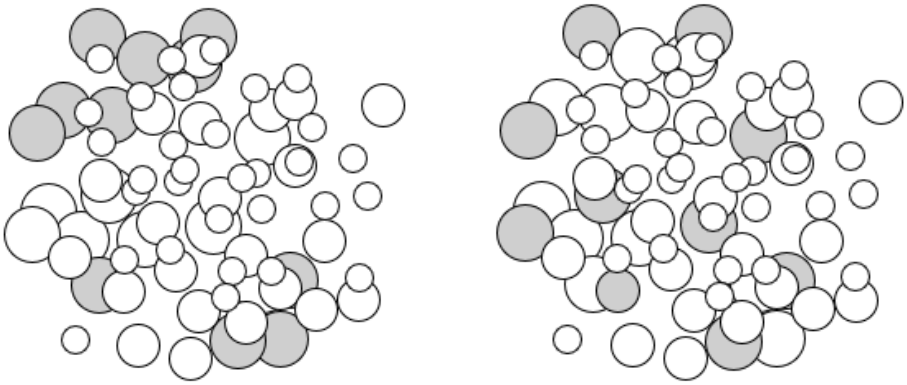
where L is labeled training set. In order to facilitate understanding, we also use D_{train} to represent the labeled training set.

A naive approach would be to continuously query a batch of samples based on the one-by-one strategy. For example, [71, 103] adopts the method of batch acquisition and chooses BALD (Bayesian Active Learning by Disagreement) [97] to query top-K samples with the highest scores. The acquisition function a_{BALD} of this idea is expressed as follows:

$$\begin{aligned} &a_{\text{BALD}}\left(\{x_1, \ldots, x_b\}, \mathcal{P}\left(\omega \mid D_{train}\right)\right) = \sum_{i=1}^{b} \mathbb{I}\left(y_i; \omega \mid x_i, D_{train}\right), \\ &\mathbb{I}\left(y; \boldsymbol{\omega} \mid x, D_{train}\right) = \mathbb{H}\left(y \mid x, D_{train}\right) - \mathbb{E}_{\mathcal{P}(\boldsymbol{\omega} \mid D_{train})}\left[\mathbb{H}\left(y \mid x, \boldsymbol{\omega}, D_{train}\right)\right], \end{aligned} \tag{3}$$

where $\mathbb{I}\left(y; \boldsymbol{\omega} \mid x, D_{train}\right)$ used in BALD is to estimate the mutual information between model parameters and model predictions. The larger the mutual information value $\mathbb{I}(*)$, the higher the uncertainty of the sample. The condition of $\boldsymbol{\omega}$ on D_{train} indicates that the model has been trained with D_{train}. And $\omega \sim \mathcal{P}\left(\omega \mid D_{train}\right)$ represents the model parameters of the current Bayesian model. $\mathbb{H}(*)$ represents the entropy of the model prediction. $\mathbb{E}[H(*)]$ is the expectation of the entropy of the model prediction over the posterior of the model parameters. Equation (3) considers each sample independently and selects samples to construct a batch query dataset in a one-by-one way.

Clearly, however, this method is not feasible, as it is very likely to choose a set of information-rich but similar samples. The information provided to the model by such similar samples is essentially the same, which not only wastes labeling resources, but also makes it difficult for the model to learn genuinely useful information. In addition, this query method that considers each sample



(a) Batch query strategy considering only the amount of information.

(b) Batch query strategy considering both information volume and diversity.

Fig. 2. A comparison diagram of two batch query strategies, one that only considers the amount of information and one that considers both the amount and diversity of information. The size of the dots indicates the amount of information in the samples, while the distance between the dots represents the similarity between the samples. The points shaded in gray indicate the sample points to be queried in a batch.

independently also ignores the correlation between samples. This is likely to lead to local decisions that make the batch sample set of queries insufficiently optimized. Therefore, how to simultaneously consider the correlation between different query samples is the primary problem for BMDAL. To solve the above problems, BatchBALD [115] expands BALD, which considers the correlation between data points by estimating the joint mutual information between multiple data points and model parameters. The acquisition function of BatchBALD can be expressed as follows:

$$\begin{aligned} &a_{\text{BatchBALD}}\left(\{x_1,\ldots,x_b\},\mathcal{P}\left(\omega \mid D_{train}\right)\right)=\mathbb{I}\left(y_1,\ldots,y_b;\omega \mid x_1,\ldots,x_b,D_{train}\right),\\ &\mathbb{I}\left(y_{1:b};\boldsymbol{\omega} \mid x_{1:b},D_{train}\right)=\mathbb{H}\left(y_{1:b} \mid x_{1:b},D_{train}\right)-\mathbb{E}_{\mathcal{P}(\boldsymbol{\omega}\mid D_{train})}\mathbb{H}\left(y_{1:b} \mid x_{1:b},\boldsymbol{\omega},D_{train}\right), \end{aligned} \tag{4}$$

where $x_1,\ldots,x_b$ and $y_1,\ldots,y_b$ are represented by joint random variables $x_{1:b}$ and $y_{1:b}$ in a product probability space, and $\mathbb{I}\left(y_{1:b};\boldsymbol{\omega} \mid x_{1:b},D_{train}\right)$ denotes the mutual information between these two random variables. BatchBALD considers the correlation between different query samples by designing an explicit joint mutual information mechanism to obtain a better query batch sample set.

The batch-based query strategy forms the basis of the combination of AL and DL, and related research on this topic is also very rich. We will provide a detailed overview and discussion of BMDAL query strategies in the following sections.

3.1.2 Uncertainty-based and Hybrid Query Strategies. Because the uncertainty-based approach is simple in form and has low computational complexity, it is a very popular query strategy in AL. This query strategy is mainly used in certain shallow models (eg, SVM [222] or KNN [102]). This is mainly because the uncertainty of these models can be accurately obtained by traditional uncertainty sampling methods. In uncertainty-based sampling, learners try to select the most uncertain samples to form a batch query set. For example, in the margin sampling [188], margin M is defined as the difference between the predicted highest probability and the predicted second highest probability of an sample as follows: $M=P\left(y_1 \mid x\right)-P\left(y_2 \mid x\right)$, where y_1 and y_2 are the first and second most probable labels predicted for the sample x under the current model. The

smaller the margin M, the greater the uncertainty of the sample x. The AL algorithm selects the top-K samples with the smallest margin M as the batch query set by calculating the margin M of all unlabeled samples. Information entropy [193] is also a commonly used uncertainty measurement standard. For a k-class task, the information entropy $\mathbb{E}(x)$ of sample x can be defined as follows:

$$\mathbb{E}(x) = -\sum_{i=1}^{k} P(y_i \mid x) \cdot \log\left(P(y_i \mid x)\right), \tag{5}$$

where $P(y_i \mid x)$ is the probability that the current sample x is predicted to be class y_i. The greater the entropy of the sample, the greater its uncertainty. Therefore, the top-K samples with the largest information entropy should be selected. More query strategies based on uncertainty can be found in [3].

There are many DeepAL [13, 88, 156, 173] methods that directly utilize an uncertainty-based sampling strategy. However, DFAL (DeepFool Active Learning) [57] contends that these methods are easily fooled by adversarial examples; thus, it focuses on the study of examples near the decision boundary, and actively uses the information provided by these adversarial examples on the input spatial distribution in order to approximate their distance to the decision boundary. This adversarial query strategy can effectively improve the convergence speed of CNN training. Nevertheless, as analyzed in Section 3.1.1, this can easily lead to insufficient diversity of batch query samples (such that relevant knowledge regarding the data distribution is not fully utilized), which in turn leads to low or even invalid DL model training performance. A feasible strategy would thus be to use a hybrid query strategy in a batch query, taking into account both the information volume and diversity of samples in either an explicit or implicit manner.

The performance of early Batch Mode Active Learning (BMAL) [29, 107, 153, 218, 235, 238] algorithms are often excessively reliant on the measurement of similarity between samples. In addition, these algorithms are often only good at exploitation (learners tend to focus only on samples near the current decision boundary, corresponding to high-information query strategies), meaning that the samples in the query batch sample set cannot represent the true data distribution of the feature space (due to the insufficient diversity of batch sample sets). To address this issue, Exploration-P [244] uses a deep neural network to learn the feature representation of the samples, then explicitly calculates the similarity between the samples. At the same time, the processes of exploitation and exploration (in the early days of model training, learners used random sampling strategies for exploration purposes) are balanced to enable more accurate measurement of the similarity between samples. More specifically, Exploration-P uses the information entropy in Equation (5) to estimate the uncertainty of sample x under the current model. The uncertainty of the selected sample set S can be expressed as $E(S) = \sum_{x_i \in S} \mathbb{E}(x_i)$. Furthermore, to measure the redundancy between samples in the selected sample set S, Exploration-P uses $R(S)$ to represent the redundancy of selected sample set S:

$$R(S) = \sum_{x_i \in S} \sum_{x_j \in S} Sim(x_i, x_j), \quad Sim(x_i, x_j) = f(x_i)\mathcal{M}f(x_j), \tag{6}$$

where $f(x)$ represents the feature of sample x extracted by deep learning model f, $Sim(x_i, x_j)$ measures the similarity between two samples, and $\mathcal{M}$ is a similarity matrix (when $\mathcal{M}$ is the identity matrix, the similarity of two samples is the product of their feature vectors. In addition, $\mathcal{M}$ can also be learned as a parameter of f). Therefore, the selected sample set S is expected to have the largest uncertainty and the smallest redundancy. For this reason, Exploration-P considers these two strategies, and the final goal equation is defined as:

$$\mathrm{I}(S) = E(S) - \frac{\alpha}{|S|} R(S), \tag{7}$$

where, α is used to balance the weight of the hybrid query strategies, uncertainty and redundancy.

Moreover, DMBAL (Diverse Mini-Batch Active Learning) [255] adds informativeness to the optimization goal of K-means by weight, and further presents an in-depth study of a hybrid query strategy that considers the sample information volume and diversity under the mini-batch sample query setting. DMBAL [255] can easily achieve expansion from the generalized linear model to DL; this not only increases the scalability of DMBAL [255] but also increases the diversity of active query samples in the mini-batch. Fig.2 illustrates a schematic diagram of this idea. This hybrid query strategy is quite popular. For example, WI-DL (Weighted Incremental Dictionary Learning) [135] mainly considers the two stages of DBN. In the unsupervised feature learning stage, the key consideration is the representativeness of the data, while in the supervised fine-tuning stage, the uncertainty of the data is considered; these two indicators are then integrated, and finally optimized using the proposed weighted incremental dictionary learning algorithm.

Although the above improvements have resulted in a good performance, there is still a hidden danger that must be addressed: namely, that, diversity-based strategies are not appropriate for all datasets. More specifically, the richer the category content of the dataset, the larger the batch size, and the better the effect of diversity-based methods; by contrast, an uncertainty-based query strategy will perform better with smaller batch sizes and less rich content. These characteristics depend on the statistical characteristics of the dataset. The BMAL context, whether the data are unfamiliar and potentially unstructured, makes it impossible to determine which AL query strategy is more appropriate. In light of this, BADGE (Batch Active learning by Diverse Gradient Embeddings) [14] samples point groups that are disparate and high magnitude when represented in a hallucinated gradient space, meaning that both the prediction uncertainty of the model and the diversity of the samples in a batch are considered simultaneously. Most importantly, BADGE can achieve an automatic balance between forecast uncertainty and sample diversity without the need for manual hyperparameter adjustments. Moreover, while BADGE [14] considers this hybrid query strategy in an implicit way, WAAL (Wasserstein Adversarial Active Learning) [200] proposes a hybrid query strategy that explicitly balances uncertainty and diversity. In addition, WAAL [200] uses Wasserstein distance to model the interactive procedure in AL as a distribution matching problem, derives losses from it, and then decomposes WAAL [200] into two stages: DNN parameter optimization and query batch selection. TA-VAAL (Task-Aware Variational Adversarial Active Learning) [112] also explores the balance of this hybrid query strategy. The assumption underpinning TA-VAAL is that the uncertainty-based method does not make good use of the overall data distribution, while the data distribution-based method often ignores the structure of the task. Consequently, TA-VAAL proposes to integrate the loss prediction module [245] and the concept of RankCGAN [185] into VAAL (Variational Adversarial Active Learning) [204], enabling both the data distribution and the model uncertainty to be considered. TA-VAAL has achieved good performance on various balanced and unbalanced benchmark datasets. The structure diagram of TA-VAAL and VAAL is presented in Fig.3.

Notably, although the hybrid query strategy achieves superior performance, the uncertainty-based AL query strategy is more convenient to combine with the output of the softmax layer of DL. Thus, the query strategy based on uncertainty is still widely used.

3.1.3 Deep Bayesian Active Learning (DBAL). As noted in Section 2, which analyzes the challenge of combining DL and AL, the acquisition function based on uncertainty is an important research direction of many classic AL algorithms. Moreover, traditional DL methods rarely represent such model uncertainty.

To solve the above problems, Deep Bayesian Active Learning appears. In the given input set X and the output Y belonging to class c, the probabilistic neural network model can be defined as

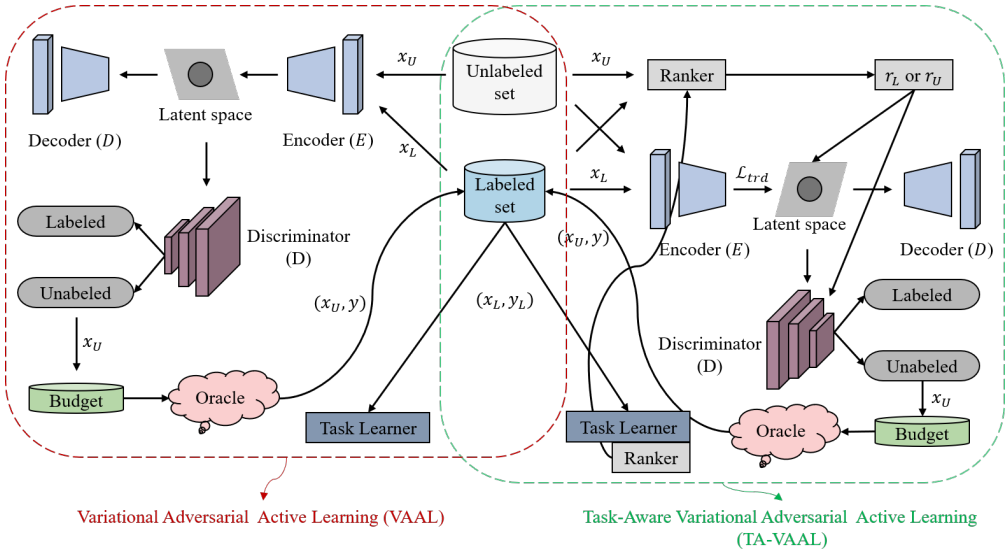


Fig. 3. Structure comparison chart of VAAL [204] and TA-VAAL [112]. 1) VAAL uses labeled data and unlabeled data in a semi-supervised way to learn the latent representation space of the data, then selects the unlabeled data with the largest amount of information according to the latent space for labeling. 2) TA-VAAL expands VAAL and integrates the loss prediction module [245] and RankCGAN [185] into VAAL in order to consider data distribution and model uncertainty simultaneously.

$f(\mathrm{x};\theta)$, $p(\theta)$ is a prior on the parameter space θ (usually Gaussian), and the likelihood $p(\mathrm{y}=c|\mathrm{x},\theta)$ is usually given by $softmax(f(\mathrm{x};\theta))$. Our goal is to obtain the posterior distribution over θ, as follows:

$$p(\theta|X,Y) = \frac{p(Y|X,\theta)p(\theta)}{p(Y|X)}. \tag{8}$$

For a given new data point x^*, $\hat{\mathrm{y}}$ is predicted by:

$$p\left(\hat{\mathrm{y}}|\mathrm{x}^*,X,Y\right) = \int p\left(\hat{\mathrm{y}}|\mathrm{x},\theta\right)p(\theta|X,Y)d\theta = \mathbb{E}_{\theta\sim p(\theta|X,Y)}[f(\mathrm{x};\theta)]. \tag{9}$$

DBAL [72] combines BCNNs (Bayesian Convolutional Neural Networks) [70] with AL methods to adapt BALD [97] to the deep learning environment, thereby developing a new AL framework for high-dimensional data. This approach adopts the above method to first perform Gaussian prior modeling on the weights of a CNN, and then uses variational inference to obtain the posterior distribution of network prediction. In addition, in practice, researchers often also use a powerful and low-cost MC-dropout (Monte-Carlo dropout) [212] stochastic regularization technique to obtain posterior samples, consequently attaining good performance on real-world datasets [111, 130]. Moreover, this regularization technique has been proven to be equivalent to variational inference [71]. However, a core-set approach [192] points out that DBAL [72] is unsuitable for large datasets due to the need for batch sampling. It should be noted here that while DBAL [72] allows the use of dropout in testing for better confidence estimation, the analysis presented in [79] contends that the performance of this method is similar to the performance of using neural network SR [229] as uncertainty sampling, which requires vigilance. In addition, DEBAL (Deep Ensemble Bayesian Active Learning) [165] argues that the pattern collapse phenomenon [211] in the variational inference method leads to the overconfident prediction characteristic of the DBAL method. For

this reason, DEBAL combines the expressive power of ensemble methods with MC-dropout to obtain better uncertainty in the absence of trading representativeness. For its part, BatchBALD [115] opts to expand BALD [97] to the batch query context; this approach no longer calculates the mutual information between a single sample and model parameters but rather recalculates the mutual information between the batch samples and the model parameters to jointly score the batch of samples. This enables BatchBALD to more accurately evaluate the joint mutual information. Inspired by the latest research on Bayesian core sets [33, 99], ACS-FW (Active Bayesian CoreSets with Frank-Wolfe optimization) [162] reconstructed the batch structure to optimize the sparse subset approximation of the log-posterior induced by the entire dataset. Using this similarity, ACS-FW then employs the Frank-Wolfe [68] algorithm to enable effective Bayesian AL at scale, while its use of random projection has made it still more popular. Compared with other query strategies (e.g., maximizing the predictive entropy (MAXENT) [72, 192] and BALD [97]), ACS-FW achieves better coverage across the entire data manifold. DPEs (Deep Probabilistic Ensembles) [39] introduces an expandable DPEs technology, which uses a regularized ensemble to approximate the deep BNN, and then evaluates the classification effect of these DPEs in a series of large-scale visual AL experiments.

ActiveLink (Deep Active Learning for Link Prediction in Knowledge Graphs) [156] is inspired by the latest advances in Bayesian deep learning [71, 236]. Adopting the Bayesian view of the existing neural link predictors, it expands the uncertainty sampling method by using the basic structure of the knowledge graph, thereby creating a novel DeepAL method. ActiveLink further noted that although AL can sample efficiently, the model needs to be retrained from scratch for each iteration in the AL process, which is unacceptable in the DL model training context. A simple solution would be to use newly selected data to train the model incrementally, or to combine it with existing training data [199]; however, this would cause the model to be biased either towards a small amount of newly selected data or towards data selected early in the process. In order to solve this bias problem, ActiveLink adopts a principled and unbiased incremental training method based on meta-learning. More specifically, in each AL iteration, ActiveLink uses the newly selected samples to update the model parameters, then approximates the meta-objective of the model's future prediction by generalizing the model based on the samples selected in the previous iteration. This enables ActiveLink to strike a balance between the importance of the newly and previously selected data, and thereby to achieve an unbiased estimation of the model parameters.

In addition to the above-mentioned DBAL work, due to the lesser parameter of BNN and the uncertainty sampling strategy being similar to traditional AL, the research on DBAL is quite extensive, and there are many works related to this topic [83, 143, 176, 201, 242, 247].

3.1.4 Density-based Methods. The term, density-based method, mainly refers to the selection of samples from the perspective of the set (core set [161]). The construction of the core set is a representative query strategy. This idea is mainly inspired by the compression idea of the core set dataset and attempts to use the core set to represent the distribution of the feature space of the entire original dataset, thereby reducing the labeling cost of AL.

FF-Active (Farthest First Active Learning) [75] is based on this idea and uses the farthest-first traversal in the space of neural activation over a representation layer to query consecutive points from the pool. It is worth noting here that FF-Active [75] and Exploration-P [244] resemble the way in which random queries are used in the early stages of AL to enhance AL's exploration ability, which prevents AL from falling into the trap of insufficient sample diversity. Similarly, to solve the sampling bias problem in batch querying, the diversity of batch query samples is increased. The Core-set approach [192] attempts to solve this problem by constructing a core subset. A further attempt was made to solve the k-Center problem [62] by building a core subset so that the model

learned on the selected core set will be more competitive than the rest of the data. However, the Core-set approach requires a large distance matrix to be built on the unlabeled dataset, meaning that this search process is computationally expensive; this disadvantage will become more apparent on large-scale unlabeled datasets [14].

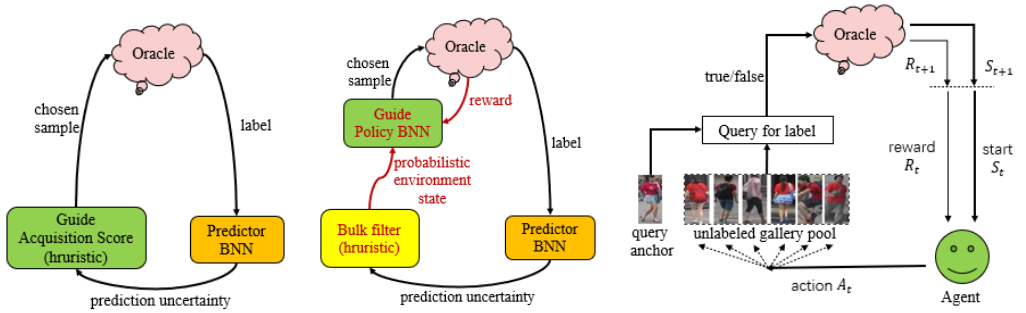
Active Palmprint Recognition [56] applies DeepAL to high-dimensional and complex palmprint recognition data. Similar to the core set concept, [56] regards AL as a binary classification task. It is expected that the labeled and unlabeled sample sets will have the same data distribution, making the two difficult to distinguish; that is, the goal is to find a labeled core subset with the same distribution as the original dataset. More specifically, due to the heuristic generative model simulation data distribution being difficult to train and unsuitable for high-dimensional and complex data such as palm prints, the author considers whether the sample can be positively distinguished from the unlabeled or labeled dataset with a high degree of confidence. Those samples that can be clearly distinguished are obviously different from the data distribution of the core annotation subset. These samples will then be added to the annotation dataset for the next round of training. Previous core-set-based methods [75, 192] often simply try to query data points as far as possible to cover all points of the data manifold without considering the density, which results in the queried data points overly representing sample points from manifold sparse areas. Similar to [56], DAL (Discriminative Active Learning) [79] also regards AL as a binary classification task and further aims to make the queried labeled dataset indistinguishable from the unlabeled dataset. The key advantage of DAL [79] is that it can sample from the unlabeled dataset in proportion to the data density, without biasing the sample points in the sparse popular domain. Moreover, the method proposed by DAL [79] is not limited to classification tasks, which are conceptually easy to transfer to other new tasks.

In addition to the corresponding query strategy, some researchers have also considered the impact of batch query size on query performance. For example, [14, 115, 162, 255] focus primarily on the optimization of query strategies in smaller batches, while [38] recommended expanding the query scale of AL for large-scale sampling (10k or 500k samples at a time). Moreover, by integrating hundreds of models and reusing intermediate checkpoints, the distributed searching of training data on large-scale labeled datasets can be efficiently realized with a small computational cost. [38] also proved that the performance of using the entire dataset for training is not the upper limit of performance, as well as that AL based on subsets specifically may yield better performance.

Furthermore, the attributes of the dataset itself also have an important impact on the performance of DeepAL. With this in mind, GA (Gradient Analysis) [225] assesses the relative importance of image data in common datasets and proposes a general data analysis tool design to facilitate a better understanding of the diversity of training examples in the dataset. GA [225] finds that not all datasets can be trained on a small sub-sample set because the relative difference of sample importance in some datasets is almost negligible; therefore, it is not advisable to blindly use smaller sub-datasets in the AL context. In addition, [19] finds that compared with the Bayesian deep learning approach (Monte-Carlo dropout [72]) and density-based [191] methods, ensemble-based AL can effectively offset the imbalance of categories in the dataset during the acquisition process, resulting in more calibration prediction uncertainty, and thus better performance.

In general, density-based methods primarily consider the selection of core subsets from the perspective of data distribution. There are relatively few related research methods, which suggests a new possible direction for sample querying.

3.1.5 Automated Design of DeepAL. DeepAL is composed of two parts: deep learning and active learning. Manually designing these two parts requires a lot of energy and their performance is severely limited by the experience of researchers. Therefore, it has important significance to



(a) Active learning pipeline. (b) Reinforced Active Learning (RAL) [86]. (c) Deep Reinforcement Active Learning (DRAL) [136].

Fig. 4. Comparison of standard AL, RAL [86] and DRAL [136] pipelines.

consider how to automate the design of deep learning models and active learning query strategies in DeepAL.

To this end, [61] redefines the heuristic AL algorithm as a reinforcement learning problem and introduces a new description through a clear selection strategy. In addition, some researchers have also noted that, in traditional AL workflows, the acquisition function is often regarded as a fixed known prior, and that it will not be known whether this acquisition function is appropriate until the label budget is exhausted. This makes it impossible to flexibly and quickly tune the acquisition function. Accordingly, one good option may be to use reinforcement learning to dynamically tune the acquisition function. RAL (Reinforced Active Learning) [86] proposes to use BNN as a learning predictor for acquisition functions. As such, all probability information provided by the BNN predictor will be combined to obtain a comprehensive probability distribution; subsequently, the probability distribution is sent to a BNN probabilistic policy network, which performs reinforcement learning in each labeling round based on the oracle feedback. This feedback will fine-tune the acquisition function, thereby continuously improving its quality. DRAL (Deep Reinforcement Active Learning) [136] adopts a similar idea and designs a deep reinforcement active learning framework for the person Re-ID task. This approach uses the idea of reinforcement learning to dynamically adjust the acquisition function so as to obtain high-quality query samples. Fig.4 presents a comparison between traditional AL, RAL and DRAL pipelines. The pipeline of AL is shown in Fig.4a. The standard AL pipeline usually consists of three parts. The oracle provides a set of labeled data; the predictor (here, BNN) is used to learn these data and provides predictable uncertainty for the guide. The guide is usually a fixed, hard-coded acquisition function that picks the next sample for the oracle to restart the cycle. The pipeline of RAL (Reinforced Active Learning) [86] is shown in Fig.4b. RAL replaces the fixed acquisition function with the policy BNN. The policy BNN learns in a probabilistic manner, obtains feedback from the oracle, and learns how to select the next optimal sample point (new parts in red) in a reinforcement learning-based manner. Therefore, RAL can adjust the acquisition function more flexibly to adapt to the existing dataset. The pipeline of DRAL (Deep Reinforcement Active Learning) [136] is shown in Fig.4c. DRAL utilizes a deep reinforcement active learning framework for the person Re-ID task. For each query anchor (probe), the agent (reinforcement active learner) will select sequential instances from the gallery pool during the active learning process and hand it to the oracle to obtain manual annotation with

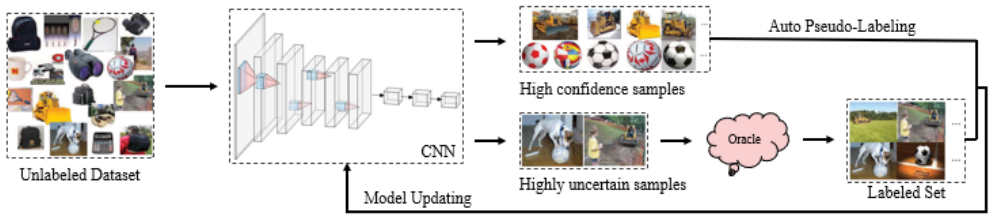


Fig. 5. In CEAL [229], the overall framework of DeepAL is utilized. CEAL [229] gradually feeds the samples from the unlabeled dataset to the initialized CNN, after which the CNN classifier outputs two types of samples: a small number of uncertain samples and a large number of samples with high prediction confidence. A small number of uncertain samples are labeled through the oracle, and the CNN classifier is used to automatically assign pseudo-labels to a large number of high-prediction confidence samples. These two types of samples are then used to fine-tune the CNN, and the updated process is repeated.

binary feedback (positive/negative). The state evaluates the similarity relationships between all instances and calculates rewards based on oracle feedback to adjust agent queries.

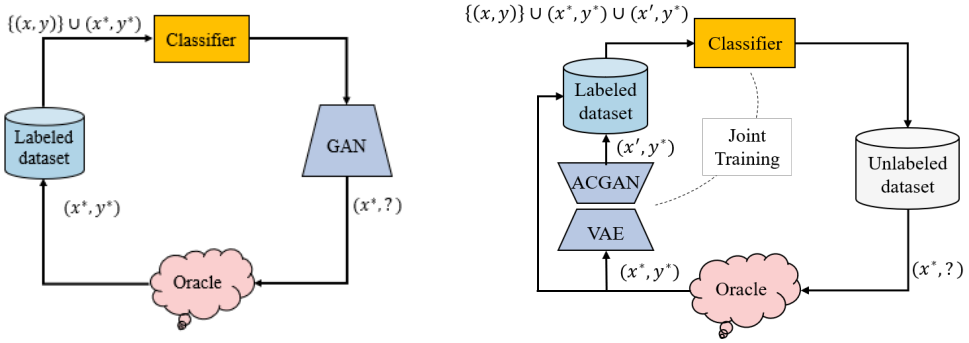
On the other hand, Active-iNAS (Active Learning with incremental Neural Architecture Search) [76] notices that most previous DeepAL methods [4, 6, 123] assume that a suitable DL model has been designed for the current task, meaning that their primary focus is on how to design an effective query mechanism; however, the existing DL model is not necessarily optimal for the current DeepAL task. Active-iNAS [76] accordingly challenges this assumption and uses NAS (neural architecture search) [174] technology to dynamically search for the most effective model architectures while conducting active learning. There is also some work devoted to providing a convenient performance comparison platform for DeepAL; for example, [148] discusses and studies the robustness and reproducibility of the DeepAL method in detail, and presents many useful suggestions.

In general, these query strategies are not independent of each other but are rather interrelated. Batch-based BMDAL provides the basis for the update training of AL query samples on the DL model. Although the query strategies in DeepAL are rich and complex, they are largely designed to take the diversity and uncertainty of query batches in BMDAL into account. Previous uncertainty-based methods often ignore the diversity in the batch and can thus be roughly divided into two categories: those that design a mechanism that explicitly encourages batch diversity in the input or learning representation space, and those that directly measure the mutual information (MI) of the entire batch.

3.2 Data Expansion of Labeled Samples in DeepAL

AL often requires only a small amount of labeled sample data to realize learning and model updating, while DL requires a large amount of labeled data for effective training. Therefore, the combination of AL and DL requires as much as possible to use the data strategy without consuming too much human resources to achieve DeepAL model training. Most previous DeepAL methods [253] often only train on the labeled sample set sampled by the query strategy. However, this ignores the existence of existing unlabeled datasets, meaning that the corresponding data expansion and training strategies are not fully utilized. These strategies help to improve the problem of insufficient labeled data in DeepAL training without adding to the manual labeling costs. Therefore, the study of these strategies is also quite meaningful.

For example, CEAL (Cost-Effective Active Learning) [229] enriches the training set by assigning pseudo-labels to samples with high confidence in model prediction in addition to the labeled dataset



(a) Generative adversarial active learning (GAAL).

(b) Bayesian generative active deep learning (BGADL).

Fig. 6. Structure comparison chart of GAAL [259] and BGADL [223]. For more details, please see [223].

sampled by the query strategy. This expanded training set is then also used in the training of the DL model. This strategy is shown in Fig.5. Another very popular strategy involves performing unsupervised training on labeled and unlabeled datasets and incorporating other strategies to train the entire network structure. For example, WI-DL [135] notes that full DBN training requires a large number of training samples, and it is impractical to apply DBN to a limited training set in an AL context. Therefore, in order to improve the training efficiency of DBN, WI-DL employs a combination of unsupervised feature learning on all datasets and supervised fine-tuning on labeled datasets.

At the same time, some researchers have considered using GAN (Generative Adversarial Networks) for data augmentation. For example, GAAL (Generative Adversarial Active Learning) [259] introduced the GAN to the AL query method for the first time. GAAL aims to use generative learning to generate samples with more information than the original dataset. However, random data augmentation does not guarantee that the generated samples will have more information than those contained in the original data, and could thus represent a waste of computing resources. Accordingly, BGADL (Bayesian Generative Active Deep Learning) [223] expands the idea of GAAL [259] and proposes a Bayesian generative active deep learning method. More specifically, BGADL combines the generative adversarial active learning [259], Bayesian data augmentation [224], ACGAN (Auxiliary-Classifier Generative Adversarial Networks) [155] and VAE (Variational Autoencoder) [114] methods, with the aim of generating samples of disagreement regions [194] belonging to different categories. Structure comparison between GAAL and BGADL is presented in Fig.6.

Subsequently, VAAL [204] and ARAL (Adversarial Representation Active Learning) [146] borrowed from several previous methods [135, 223, 259] not only to train the network using labeled and unlabeled datasets but also to introduce generative adversarial learning into the network architecture for data augmentation purposes, thereby further improving the learning ability of the network. In more detail, VAAL [204] noticed that the batch-based query strategy based on uncertainty not only readily leads to insufficient sample diversity, but is also highly susceptible to interference from outliers. In addition, density-based methods [192] are susceptible to p-norm limitations when applied to high-dimensional data, resulting in calculation distances that are too concentrated [54]. To this end, VAAL [204] proposes to use the adversarial learning representation method to distinguish between the potential spatial coding features of labeled and unlabeled data,

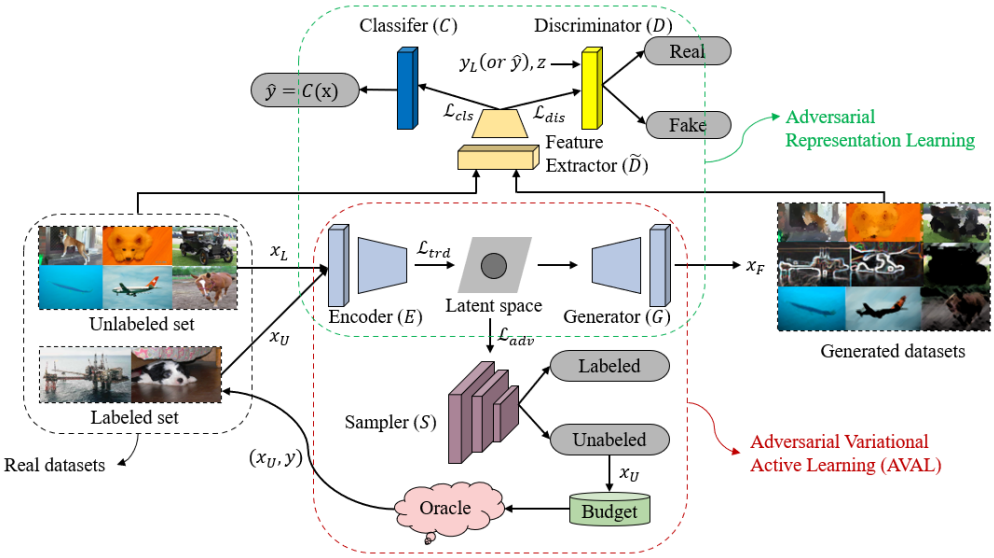


Fig. 7. The overall structure of ARAL [146]. ARAL uses not only real datasets (both labeled and unlabeled), but also generated datasets to jointly train the network. The whole network consists of an encoder (E), generator (G), discriminator (D), classifier (C) and sampler (S), and all parts of the model are trained together.

thus reducing interference from outliers. VAAL [204] also uses labeled and unlabeled data to jointly train a VAE [114, 208] in a semi-supervised manner; the goal here is to deceive the adversarial network [80] into predicting that all data points come from the labeled pool, in order to solve the problem of distance concentration. VAAL [204] can learn an effective low-dimensional latent representation on a large-scale dataset, and further provides an effective sampling method by jointly learning the representation form and uncertainty.

Subsequently, ARAL [146] expanded VAAL [204], aiming to use as few manual annotation samples as possible while still making full use of the existing or generated data information in order to improve the model's learning ability. In addition to using labeled and unlabeled datasets, ARAL [146] also uses samples produced by deep production networks to jointly train the entire model. ARAL [146] comprises both VAAL [204] and adversarial representation learning [53]. By using VAAL [204] to learn the potential feature representation space of the labeled and unlabeled data, the unlabeled samples with the largest amount of information can be selected accordingly. At the same time, both real and generated data are used to enhance the model's learning ability through confrontational representation learning [53]. Similarly, TA-VAAL [112] also extends VAAL by using the global data structure from VAAL and local task-related information from the learning loss for sample querying purposes. We present the framework of ARAL [146] in Fig.7.

Unlike ARAL [146] and VAAL [204], which use labeled and unlabeled datasets for adversarial representation learning, SSAL (Semi-Supervised Active Learning) [202] implements a new training method. More specifically, SSAL [202] uses unsupervised, supervised, and semi-supervised learning methods across AL cycles, and makes full use of existing information for training without increasing the cost of labeling as much as possible. In more detail, the process is as follows: before the AL starts, first use labeled and unlabeled data for unsupervised pretraining. In each AL learning cycle, first, perform supervised training on the labeled dataset, then perform semi-supervised training on all datasets. This represents an attempt to devise a wholly new training method. The author finds

that, compared with the difference between the sampling strategies, this model training method yields a surprising performance improvement.

As analyzed above, this kind of exploration of training methods and data utilization skills is also essential; in fact, the resultant performance gains may even exceed those generated by changing the query strategy. Applying these techniques enables the full use of existing data without any associated increase in labeling costs, which helps in resolving the issue of the number of AL query samples being insufficient to support the updating of the DL model.

3.3 DeepAL Generic Framework

As mentioned in Section 2, a processing pipeline inconsistency exists between AL and DL; thus, only fine-tuning the DL model in the AL framework, or simply combining AL and DL to treat them as two separate problems, may cause divergence. For example, [13] first conducts offline supervised training of the DL model on two different types of session datasets to grant basic conversational capabilities to the backbone network, then enables the online AL stage to interact with human users, enabling the model to be improved in an open way based on user feedback. AL-DL [227] proposes an AL method for DL models with DBNs, while ADN [257] further proposes an active deep network architecture for sentiment classification. [213] proposes an AL algorithm using CNN for captcha recognition. However, generally speaking, the above methods first perform routine supervised training on this depth model on the labeled dataset, then actively sample based on the output of the depth model. There are many similar related works [63, 198] that adopt this split-and-splitting approach that treats the training of AL and deep models as two independent problems and consequently increases the possibility, which the two problems will diverge. Although this method achieved some success at the time, a general framework that closely combines the two tasks of DL and AL would play a vital role in the performance improvement and promotion of DeepAL.

CEAL [229] is one of the first works to combine AL and DL in order to solve the problem of depth image classification. CEAL [229] merges deep convolutional neural networks into AL, and consequently proposes a novel DeepAL framework. It sends samples from the unlabeled dataset to the CNN step by step, after which the CNN classifier outputs two types of samples: a small number of uncertain samples and a large number of samples with high prediction confidence. A small number of uncertain samples are labeled by the oracle, and the CNN classifier is used to automatically assign pseudo-labels to a large number of high-prediction-confidence samples. Then, these two types of samples are used to fine-tune the CNN and the update process is repeated. In Fig.5, we present the overall framework of CEAL. Moreover, HDAL (Heuristic Deep Active Learning) [132] uses a similar framework for face recognition tasks: it combines AL with a deep CNN model to integrate feature learning and AL query model training.

In addition, Fig.1c illustrates a widespread general framework for DeepAL tasks. Related works include [56, 88, 140, 243, 254] , among others. More specifically, [243] proposes a framework that uses an FCN (Fully Convolutional Network) [137] and AL to solve the medical image segmentation problem using a small number of annotations. It first trains FCN on a small number of labeled datasets, then extracts the features of the unlabeled datasets through FCN, using these features to estimate the uncertainty and similarity of unlabeled samples. This strategy, which is similar to that described in Section 3.1.2, helps to select highly uncertain and diverse samples to be added to the labeled dataset in order to start the next stage of training. Active Palmprint Recognition [56] proposes a similar DeepAL framework as that for the palmprint recognition task. The difference is that inspired by domain adaptation [20], Active Palmprint Recognition [56] regards AL as a binary classification task: it is expected that the labeled and unlabeled sample sets have the same data distribution, making the two difficult to distinguish. Supervision training can be performed directly

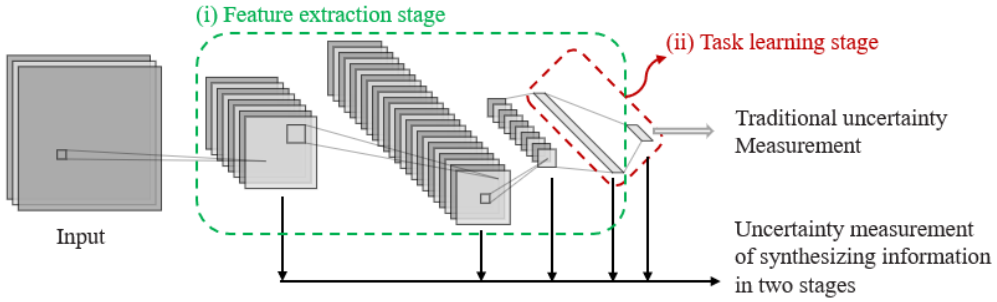


Fig. 8. Taking a common CNN as an example, this figure presents a comparison between the traditional uncertainty measurement method [56, 140, 243] and the uncertainty measurement method of synthesizing information in two stages [88, 245, 254] (i.e., the feature extraction stage and task learning stage).

on a small number of labeled datasets, which reduces the burden associated with labeling. [140] proposes a DeepAL framework for defect detection. This approach performs uncertainty sampling based on the output features of the detection model to generate a list of candidate samples for annotation. In order to further take the diversity of defect categories in the samples into account, [140] designs an average margin method to control the sampling ratio of each defect category.

Different from the above methods, it is common for the final output of the DL model to be used as the basis for determining the uncertainty or diversity of the sample (Active Palmprint Recognition [56] uses the output of the first fully connected layer). [88, 245, 254] also used the output of the DL model's middle hidden layer. As analyzed in Section 3.1.2 and Section 2, due to the difference in learning paradigms between the deep and shallow models, the traditional uncertainty-based query strategy cannot be directly applied to the DL model. In addition, unlike the shallow model, the deep model can be regarded as composed of two stages, namely the feature extraction stage and the task learning stage. It is inaccurate to use only the output of the last layer of the DL model as the basis for evaluating the sample prediction uncertainty; this is because the uncertainty of the DL model is in fact composed of the uncertainty of these two stages. A schematic diagram of this concept is presented in Fig.8. To this end, AL-MV (Active Learning with Multiple Views) [88] treats the features from different hidden layers in the middle of CNN as multiview data, taking the uncertainty of both stages into account, and the AL-MV algorithm is designed to implement adaptive weighting of the uncertainty of each layer, to enable more accurate measurement of the sampling uncertainty. LLAL (Learning Loss for Active Learning) [245] also used a similar idea. More specifically, LLAL designs a small parameter module of the loss prediction module to attach to the target network, using the output of multiple hidden layers of the target network as the input of the loss prediction module. The loss prediction module is learned to predict the target loss of the unlabeled dataset, while the top-K strategy is used to select the query samples. LLAL achieves task-agnostic AL framework design at a small parameter cost and further achieves competitive performance on a variety of mainstream visual tasks (namely, image classification, target detection, and human pose estimation). Similarly, [254] uses a similar strategy to implement a DeepAL framework for finger bone segmentation tasks. [254] uses Deeply Supervised U-Net [175] as the segmentation network, then subsequently uses the output of the multilevel segmentation hidden layer and the output of the last layer as the input of AL; this input information is then integrated to form the basis for the evaluation of the sample information size. We take LLAL [245] as an example to explicate the overall network structure of this idea in Fig.9.

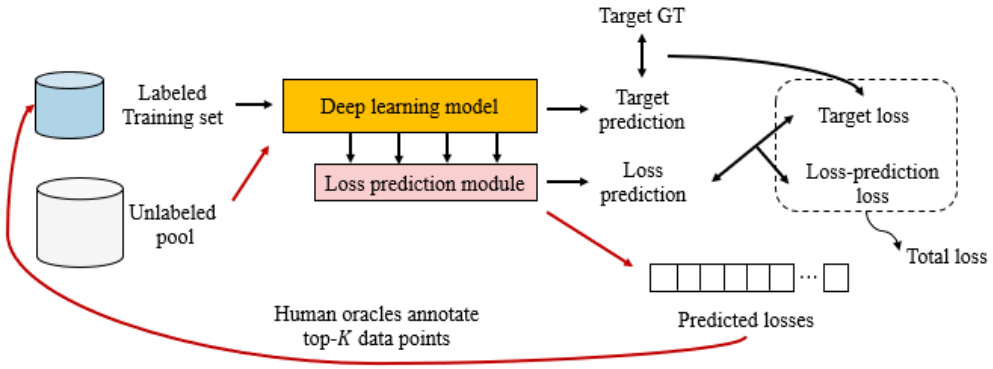


Fig. 9. The overall framework of LLAL [245]. The black line represents the stage of training model parameters, optimizing the overall loss composed of target loss and loss-prediction loss. The red line represents the sample query phase of AL. The output of the multiple hidden layers of the DL model is used as the input of the loss prediction module, while the top-K unlabeled data points are selected according to the predicted losses and assigned labels by the oracle.

The research on the general framework is highly beneficial to the development and promotion of DeepAL, as this task-independent framework can be conveniently transplanted to other fields. In the current fusion of DL and AL, DL is primarily responsible for feature extraction, while AL is mainly responsible for sample querying; thus, a deeper and tighter fusion will help DeepAL achieve better performance. Of course, this will require additional exploration and effort on the part of researchers. Finally, the challenges of combining DL and AL and related work on the corresponding solutions are summarized in Table 1.

3.4 DeepAL Stopping Strategy

In addition to querying strategies and training methods, an appropriate stopping strategy has an important impact on DeepAL performance. At present, most DeepALs [30, 66, 135, 141, 190] often use the predefined stopping criterion, and when the criterion is satisfied, they stop querying labels from the oracle. These predefined stopping criteria include the maximum number of iterations, the minimum threshold for changing classification accuracy, the minimum number of labeled samples, and the expected accuracy value, etc.

Although these stopping criteria are simple, these predefined stopping criteria are likely to cause DeepAL to fail to achieve optimal performance. This is because the premature termination of AL annotation querying leads to large performance losses in the model, and excessive annotation behavior wastes a lot of annotation budget. Therefore, Stabilizing Predictions (SP) [27] makes a comprehensive review of AL stopping strategies and proposes an AL stopping strategy based on stability prediction. Specifically, the SP predivides a part of the samples from the unlabeled dataset to form a stop set (the stop set does not need to be labeled), and the SP checks the prediction stability on the stop set in each iteration. When the prediction performance of the model on the stop set stabilizes, the iteration is stopped. A well-trained model often has a stable predictive ability, and SP takes advantage of this feature. The predivided stop set does not require specific labeling information, which avoids additional labeling costs contrary to the purpose of AL. Although SP is a stopping strategy proposed mainly for AL, it also is relevant for DeepAL.

Table 1. The challenges of combining DL and AL, as well as a summary of related work on the corresponding solutions.

Challenges	Solutions	Foundation	Category	Publications
Model uncertainty in Deep Learning	Query strategy optimization	Batch Mode DeepAL (BMDAL)	Uncertainty-based and Hybrid Query Strategies	[13, 14, 57, 88, 112, 135, 156, 173, 200, 244, 255]
			Deep Bayesian Active Learning (DBAL)	[39, 72, 115, 156, 162, 165, 176, 192, 201, 242] [83, 143, 247]
			Density-based Methods	[38, 56, 75, 79, 192, 244]
			Automated Design of DeepAL	[61, 76, 86, 136]
Insufficient data for labeled samples	Data expansion of labeled samples	-		[112, 135, 146, 202, 204, 223, 229, 259]
Processing pipeline inconsistency	Common framework DeepAL	-		[13, 63, 88, 132, 198, 213, 227, 229, 243, 257] [56, 140, 225, 245, 254]

4 APPLICATION OF DEEPAL IN FIELDS SUCH AS VISION AND NLP

Today, DeepAL has been applied to areas including but not limited to visual data processing (such as object detection, semantic segmentation, etc.), NLP (such as machine translation, text classification, semantic analysis, etc.), speech and audio processing, social network analysis, medical image processing, wildlife protection, industrial robotics, and disaster analysis, among other fields. In this section, we provide a systematic and detailed overview of existing DeepAL-related work from an application perspective.

4.1 Visual Data Processing

Just as DL is widely used in the computer vision field, the first field in which DeepAL is expected to reach its potential is that of computer vision. In this section, we mainly discuss DeepAL-related research in the field of visual data processing.

4.1.1 Image classification and recognition. As with DL, the classification and recognition of images in DeepAL form the basis for research into other vision tasks. One of the most important problems that DeepAL faces in the field of image vision tasks is that of how to efficiently query samples of high-dimensional data (an area in which traditional AL performs poorly) and obtain satisfactory performance at the smallest possible labeling cost.

To solve this problem, CEAL [229] assigns pseudo-labels to samples with high confidence and adds them to the highly uncertain sample set queried using the uncertainty-based AL method, then uses the expanded training set to train the DeepAL model image classifier. [173] first integrated the criteria of AL into the deep belief network and subsequently conducted extensive research on classification tasks on a variety of real uni-modal and multi-modal datasets. WI-DL [135] uses the DeepAL method to simultaneously consider the two selection criteria of maximizing representativeness and uncertainty on hyperspectral image (HSI) datasets for remote sensing classification tasks. Similarly, [48, 133] also studied the classification of HSI. [133] introduces AL to initialize HSI and then performs transfer learning. This work also recommends constructing and connecting higher-level features to source and target HSI data in order to further overcome the cross-domain disparity. [48] proposes a unified deep network combined with active transfer learning, thereby training the HSI classification well while using less labeled training data.

Medical image analysis is also an important application. For example, [66] explores the use of AL rather than random learning to train convolutional neural networks for tissue (e.g., stroma,

lymphocytes, tumor, mucosa, keratin pearls, blood, and background/adipose) classification tasks. [30] conducted a comprehensive review of DeepAL-related methods in the field of medical image analysis. As discussed above, since the annotation of medical images requires strong professional knowledge, it is usually both very difficult and very expensive to find well-trained experts willing to perform annotations. In addition, DL has achieved impressive performance on various image feature tasks. Therefore, a large number of works continue to focus on combining DL and AL in order to apply DeepAL to the field of medical image analysis [36, 55, 123, 181, 186, 187, 205, 206]. The DeepAL method is also used to classify in situ plankton [28] and perform the automatic counting of cells [6].

In addition, DeepAL also has a wide range of applications in our daily life. For example, [213] proposes an AL algorithm that uses CNN for verification code recognition. It can use the ability to obtain labeled data for free to avoid human intervention and greatly improve the recognition accuracy when less labeled data is used. HDAL [132] combines the excellent feature extraction capabilities of deep CNN and the ability to save on AL labeling costs to design a heuristic deep active learning framework for face recognition tasks.

4.1.2 Object detection and semantic segmentation. Object detection and semantic segmentation have important applications in various fields, including autonomous driving, medical image processing, and wildlife protection. However, these fields are also limited by the higher sample labeling cost. Thus, the lower labeling cost of DeepAL is expected to accelerate the application of the corresponding DL models in certain real-world areas where labeling is more difficult.

[178] designs a DeepAL framework for object detection, which uses the layered architecture where labeling is more difficult as an example of "query by committee" to select the image set to be queried, while at the same time introducing a similar exploration/exploitation trade-off strategy to [244]. DeepAL is also widely used in natural biological fields and industrial applications. For example, [154] uses deep neural networks to quickly transferable and automatically extract information, and further combines transfer learning and AL to design a DeepAL framework for species identification and counting in camera trap images. [110] uses unmanned aerial vehicles (UAV) to obtain images for wildlife detection purposes; moreover, to enable this wildlife detector to be reused, [110] uses AL and introduces transfer sampling (TS) to find the corresponding area between the source and target datasets, thereby facilitating the transfer of data to the target domain. [63] proposes a DeepAL framework for deep object detection in autonomous driving to train LiDAR 3D object detectors. [140] proposes the adaptation of a widespread DeepAL framework to defect detection in real industries, along with an uncertainty sampling method for use in generating candidate label categories. This work uses the average margin method to set the sampling scale of each defect category and is thus able to obtain the required performance with less labeled data.

In addition, DeepAL also has important applications in the area of medical image segmentation. For example, [74] proposes an AL-based transfer learning mechanism for medical image segmentation, which can effectively improve the image segmentation performance on a limited labeled dataset. [243] combines FCN and AL to create a DeepAL framework for biological-image segmentation. This work uses the uncertainty and similarity information provided by the FCN to extend the maximum set cover problem, significantly reducing the required labeling workload by pointing out the most effective labeling areas. DASL (Deep Active Self-paced Learning) [233] proposes a deep region-based network, Nodules R-CNN, for pulmonary nodule segmentation tasks. This work generates segmentation masks for use as examples, and at the same time, combines AL and SPL (Self-Paced Learning) [121] to propose a new deep active self-paced learning strategy that reduces the labeling workload. [232] proposes a Nodule-plus Region-based CNN for pulmonary nodule detection and segmentation in 3D thoracic Computed Tomography (CT). This work combines AL

and SPL strategies to create a new deep self-paced active learning (DSAL) strategy, which reduces the annotation workload and makes effective use of unannotated data. [254] further proposes a new deep-supervised active learning method for finger bone segmentation tasks. This model can be fine-tuned in an iterative and incremental learning manner and uses the output of the intermediate hidden layer as the basis for sample selection. Compared with the complete markup, [254] achieved comparable segmentation results using fewer samples.

4.1.3 Video processing. Compared with the image task that only needs to process information in the spatial dimension, the video task also needs to process the information in the temporal dimension. This makes the task of annotating the video more expensive, which also means that the need to introduce AL has become more urgent. DeepAL also has broader application scenarios in this field.

For example, [100] proposes to use imitation learning to perform navigation tasks. The visual environment and actions taken by the teacher viewed from a first-person perspective are used as the training set. Through training, it is hoped that students will become able to predict and execute corresponding actions in their own environment. When performing tasks, students use deep convolutional neural networks for feature extraction, learn imitation strategies, and further use the AL method to select samples with insufficient confidence, which are added to the training set to update the action strategy. [100] significantly improves the initial strategy using fewer samples. DeActive [95] proposes a DeepAL activity recognition model. Compared with the traditional DL activity recognition model, DeActive requires fewer labeled samples, consumes fewer resources, and achieves high recognition accuracy. [230] minimizes the annotation cost of the video-based person Re-ID dataset by integrating AL into the DL framework. Similarly, [136] proposes a deep reinforcement active learning method for person Re-ID, using oracle feedback to guide the agent (i.e. the model in the reinforcement learning process) in selecting the next uncertainty sample. The agent selection mechanism is continuously optimized through alternately refined reinforcement learning strategies. [4] further proposes an active learning object detection method based on convolutional neural networks for pedestrian target detection in video and static images.

4.2 Natural Language Processing (NLP)

NLP has always been a very challenging task. The goals of NLP are to make computers understand complex human language and to help humans deal with various natural language-related tasks. Insufficient data labeling is also a key challenge in the NLP context. Below, we introduce some of the most famous DeepAL methods in the NLP field.

4.2.1 Machine translation. Machine translation has very important application value, but it usually requires a large number of parallel corpora as a training set. For many low-resource language pairs, building such a corpus requires a very high cost.

For this reason, [248] proposes to use the AL framework to select information source sentences to construct a parallel corpus. It proposes two effective sentence selection methods for AL: selection based on semantic similarity and decoder probability. Compared with traditional methods, the two proposed sentence selection methods show considerable advantages. [164] proposes a curriculum learning framework related to AL for machine translation tasks. It can decide which training samples to show to the model during different periods of training based on the estimated difficulty of a sample and the current competence of the model. This method not only effectively improves the training efficiency but also obtains a good accuracy improvement. This kind of thinking is also very valuable for DeepAL's sample selection strategy.

4.2.2 Text classification. Text classification tasks also face the challenge of excessive labeling costs, such as patent classification [60, 125] and clinical text classification [64, 73, 160]. These labeling tasks often need to be completed by experts, and the number of datasets and texts in each document is often very large, which makes it difficult for human experts to complete the corresponding labeling tasks.

[251] claims to be the first AL method for text classification with CNNs. [251] focuses on selecting those samples that have the greatest impact on the embedding space. It proposes a method for sentence classification that selects instances containing words whose embeddings are likely to be updated with the greatest magnitude, thereby rapidly learning discriminative, task-specific embeddings. They also extend this method to text classification tasks, which outperformed the baseline AL method in sentence and text classification tasks. [7] also proposes a new DeepAL framework for text classification tasks. It uses RNN as the acquisition function in AL. The method proposed by [7] can effectively reduce the number of label instances required for deep learning while saving training time without reducing model accuracy. [166] focuses on the problem of sampling bias in deep active classification and apply active text classification on the large-scale text corpora of [250]. These methods generally show better performance than that of the traditional AL-based baseline methods, and more relevant DeepAL-based text classification applications can be found in [190].

4.2.3 Semantic analysis. In this typical NLP task, the aim is to make the computer understand a natural language description. The relevant application scenarios are numerous and varied, including but not limited to sentiment classification, news identification, etc.

More specifically, for example, [257] uses restricted Boltzmann machines (RBM) to construct an active deep network (ADN), then conduct unsupervised training on the labeled and unlabeled datasets. ADN uses a large number of unlabeled datasets to improve the model's generalization ability, and further employs AL in a semi-supervised learning framework, unifying the selection of labeled data and classifiers in a semi-supervised classification framework; this approach obtains competitive results on sentiment classification tasks. [22] proposes a human-computer collaborative learning system for news accuracy detection tasks (that is, identifying misleading and false information in news) that utilizes only a limited number of annotation samples. This system is a deep AL-based model that uses 1-2 orders of magnitude fewer annotation samples than fully supervised learning. Such a reduction in the number of samples greatly accelerates the convergence speed of the model and results in an astonishing 25% average performance gains in detection performance.

4.2.4 Information extraction. Information extraction aims to extract and simplify the most important information from large texts, which is an important basis for correlation analysis between different concepts.

[168] uses relevant tweets from disaster-stricken areas to extract information that facilitates the identification of infrastructure damage during earthquakes. For this reason, [168] combines RNN and GRU-based models with AL, using AL-based methods to pre-train the model so that it will retrieve tweets featuring infrastructure damage in different regions, thereby significantly reducing the manual labeling workload. In addition, entity resolution (ER) is the task of recognizing the same real entities with different representations across databases and represents a key step in knowledge base creation and text mining. [34, 197, 199] uses the combination of DL and AL to determine how the technical level of NER (Named Entity Recognition) can be improved in the case of a small training set. [109] developed a DL-based ER method that combines transfer learning and AL to design an architecture that allows for the learning of a model that is transferable from high-resource environments to low-resource environments. [141] proposes a novel ALPNN (Active Learning Policy Neural Network) design to recognize the concepts and relationships in large EEG

(electroencephalogram) reports; this approach can help humans extract available clinical knowledge from a large number of such reports.

4.2.5 Question-answering. Intelligent question-answering is also a common processing task in the NLP context, and DL has achieved impressive results in these areas. However, the performance of these applications still relies on the availability of massive labeled datasets; AL is expected to bring new hope to this challenge.

The automatic question-answering system has a very wide range of applications in the industry, and DeepAL is also highly valuable in this field. For example, [13] uses the online AL strategy combined with the DL model to achieve an open domain dialogue by interacting with real users and learning incrementally from user feedback in each round of dialogue. [104] finds that AL strategies designed for specific tasks (e.g., classification) often have only one correct answer and that these uncertainty-based measurements are often calculated based on the output of the model. Many real-world vision tasks often have multiple correct answers, which leads to the overestimation of uncertainty measures and sometimes even worse performance than random sampling baselines. For this reason, [104] proposes to estimate the uncertainty in the hidden space within the model rather than the uncertainty in the output space of the model in the Visual Question Answer (VQA) generation, thus overcoming the paraphrasing nature of language.

4.3 Other Applications

The emergence of DeepAL is exciting, as it is expected to reduce the annotation costs by orders of magnitude while maintaining performance levels. For this reason, DeepAL is also widely used in other fields.

These applications include, but are not limited to, gene expression, robotics, wearable device data analysis, social networking, ECG signal analysis, etc. For some more specific examples, MLFS (Multi-Level Feature Selection) [101] combines DL and AL to select genes/miRNAs based on expression profiles and proposes a novel multi-level feature selection method. MLFS also considers the biological relationship between miRNAs and genes and applies this method to miRNA expansion tasks. Moreover, the failure risk of real-world robots is expensive. [8] proposes a risk-aware resampling technique; this approach uses AL together with existing solvers and DL to optimize the robot's trajectory, enabling it to effectively deal with the collision problem in scenes with moving obstacles, and verify the effectiveness of the DeepAL method on a real nano-quadcopter. [258] further proposes an active trajectory generation framework for the inverse dynamics model of the robot control algorithm, which enables the systematic design of the information trajectory used to train the DNN inverse dynamics module.

In addition, [83, 96] uses sensors installed in wearable devices or mobile terminals to collect user movement information for human activity recognition purposes. [96] proposes a DeepAL framework for activity recognition with context-aware annotator selection. ActiveHARNet (Active Learning for Human Activity Recognition) [83] proposes a resource-efficient deep ensembled model that supports incremental learning and inference on the device, utilizes the approximation in the BNN to represent the uncertainty of the model, and further proves the feasibility of ActiveHARNet deployment and incremental learning on two public datasets. For its part, DALAUP (Deep Active Learning for Anchor User Prediction) [37] designs a DeepAL framework for anchor user prediction in social networks that reduces the cost of annotating anchor users and improves the prediction accuracy. DeepAL is also using in the classification of electrocardiogram (ECG) signals. For example, [171] proposes an active DL-based ECG signal classification method. [85] proposed an AL-based ECG classification method using eigenvalues and DL. The use of the AL method enables the cost of marking ECG signals by medical experts to be effectively reduced. Furthermore, the cost of label

annotation in the speech and audio fields is also relatively high. [1] finds that a model trained on a corpus composed of thousands of recordings collected by a small number of speakers is unable to be generalized to new domains; therefore, [1] developed a practical scheme that involves using AL to train deep neural networks for speech emotion recognition tasks when label resources are limited.

In general, the current applications of DeepAL are mainly focused on visual image processing tasks, although there are also applications in NLP and other fields. Compared with DL and AL, DeepAL is still in the preliminary stage of research, meaning that the corresponding classic works are relatively few; however, it still has the same broad application scenarios and practical value as DL. In addition, in order to facilitate readers' access to specific applications of DeepAL in related fields, we have classified and summarized all application scenarios and datasets used by survey-related work in Section 4 in detail. The specific information is shown in Table 2.

5 DISCUSSION AND FUTURE DIRECTIONS

DeepAL combines the common advantages of DL and AL: it inherits not only DL's ability to process high-dimensional image data and conduct automatic feature extraction but also AL's potential to effectively reduce annotation costs. DeepAL, therefore, has fascinating potential especially in areas where labels require high levels of expertise and are difficult to obtain.

Most recent work reveals that DeepAL has been successful in many common tasks. DeepAL has attracted the interest of a large number of researchers by reducing the cost of annotation and its ability to implement the powerful feature extraction capabilities of DL; consequently, the related research work is also extremely rich. However, there are still a large number of unanswered questions on this subject. As [148] discovered, the results reported on the random sampling baseline (RSB) differ significantly between different studies. For example, under the same settings, using 20% of the label data of CIFAR 10, the RSB performance reported by [245] is 13% higher than that in [223]. Secondly, the same DeepAL method may yield different results in different studies. For example, using 40% of the label data of CIFAR 100 [119] and VGG16 [203] as the extraction network, the reported results of [192] and [204] differ by 8%. Furthermore, the latest DeepAL research also exhibits some inconsistencies. For example, [192] and [57] point out that diversity-based methods have always been better than uncertainty-based methods, and that uncertainty-based methods perform worse than RSB; however, the latest research of [245] shows that this is not the case.

Compared with AL's strategic selection of high-value samples, RSB has been regarded as a strong baseline [192, 245]. However, the above problems reveal an urgent need to design a general performance evaluation platform for DeepAL work, as well as to determine a unified high-performance RSB. Secondly, the reproducibility of different DeepAL methods is also an important issue. The highly reproducible DeepAL method helps to evaluate the performance of different DALs. A common evaluation platform should be used for experiments under consistent settings, and snapshots of experimental settings should be shared. In addition, multiple repetitive experiments with different initializations under the same experimental conditions should be implemented, as this could effectively avoid misleading conclusions caused by experimental setup problems. Researchers should pay sufficient attention to these inconsistent studies to enable them to clarify the principles involved. On the other hand, adequate ablation experiments and transfer experiments are also necessary. The former will make it easier for us to determine which improvements bring about performance gains, while the latter can help to ensure that the AL selection strategy does indeed enable the indiscriminate selection of high-value samples for the dataset.

The current research directions regarding DeepAL methods focus primarily on the improvement of AL selection strategies, the optimization of training methods, and the improvement of task-independent models. As noted in Section 3.1, the improvement of AL selection strategy is

Table 2. DeepAL's research examples in Vision, NLP and other fields.

Field	Task	Publications	Datasets	Scenes
Vision	Image classification and recognition	[173, 213, 229]	CACD [35], Caltech-256 [81], VidTIMIT [183], CK [108], MNIST [128], CIFAR 10 [119], emoFBVP [172], MindReading [58] Cool PHP CAPTCHA [213]	Handwritten numbers, face, CAPTCHA recognition, etc.
		[48, 133, 135]	PaviaC, PaviaU, Botswana [135], Salinas Valley, Indian Pines [48], Washington DC Mall, Urban [133]	Hyperspectral image
		[30, 55, 66, 186] [123, 187, 206] [36, 181, 205]	Erie County [66], EEG [9], BreaKHis [210], SVEB, SVDB [186]	Biomedical
	Object detection	[178]	VOC [59], Kitti [77]	–
		[110, 154]	SS [215], eMML [67], NACTI[1], CCT[2], UAV[3]	Biodiversity survey
		[63]	KITTI [78]	Autonomous driving
		[140]	NEU-DET [209]	Defect detection
	Semantic segmentation	[74, 232, 233, 243]	SPIM [46], Confocal[47], LIDC-IDRI [12], MICCAI, Lymph node [252]	Bio-medical image
	Video processing	[100]	Mash-simulator[4]	Autonomous navigation
		[95]	OPPORTUNITY [95], WISDM [122], SenseBox [219], Skoda Daphnet [15],CASAS [40]	Smart home
		[4, 230]	PRID [93], MARS [256], BDD100K [246], DukeMTMC-VideoReID [237], CityPersons [249], Caltech Pedestrian[52]	Person Re-id
NLP	Machine translation	[164, 248]	OPUS [220], UNPC [261], IWSLT, WMT [163]	Ind-En, Ch-En, En-Vi, Fr-En, En-De, etc.
	Text classification	[7, 166, 190, 251]	CR[5], Subj, MR[6], MuR[7], DR [226] AGN, DBP, AMZP, AMZF, YRF [250]	–
	Semantic analysis	[257]	MOV [157], BOO, DVDs, ELE, KIT [24, 45]	Sentiment classification
		[22]	KDnugget's Fake News,[8] Harvard Dataverse [124], Liar [234]	News veracity detection
	Information extraction	[168]	Italy, Iran-Iraq, Mexico earthquake dataset	Disaster assessment
		[141]	Temple University Hospital[10]	Electroencephalography (EEG) reports
		[34, 109, 197, 199]	CoNLL [184], NCBI [51], MedMentions [149], OntoNotes [167], DBLP, FZ, AG [147], Cora [228]	Named entity recognition (NER)
	Question answering	[13]	CMDC [43], JabberWacky's chatlogs[9]	Dialogue generation
		[104]	Visual Genome [117], VQA [11]	Visual question answer (VQA)
Other	–	[101]	BC, HCC, Lung	Gene expression
		[8, 258]	EATG [258], Crazyflie 2.0[11]	Robotics
		[83, 96]	HHAR [214], NWFD [145]	Smart device
		[37]	Foursquare, Twitter [116]	Social network
		[85, 171]	MIT-BIH [142], INCART, SVDB [171]	Electrocardiogram (ECG) signal classification
		[1]	MSP-Podcast [138]	Speech emotion recognition

[1] http://lila.science/datasets/nacti
[2] http://lila.science/datasets/caltech-camera-traps
[3] http://kuzikus-namibia.de/xe_index.html
[4] https://github.com/idiap/mash-simulator
[5] www.cs.uic.edu/liub/FBS/sentiment-analysis.html
[6] Subj and MR datasets are available at: http://www.cs.cornell.edu/people/pabo/movie-review-data/
[7] http://www.cs.jhu.edu/~mdredze/datasets/sentiment/
[8] https://github.com/GeorgeMcIntire/fake_real_news_dataset
[9] http://www.jabberwacky.com/j2conversations. JabberWacky is an in-browser, open-domain, retrieval-based bot.
[10] https://www.isip.piconepress.com/projects/tuh_eeg/
[11] https://www.bitcraze.io/
[–] Non-specific application scenarios

currently centered around taking into account the query strategy based on uncertainty and diversity explicitly or implicitly. Moreover, hybrid selection strategies are increasingly favored by researchers. Moreover, the optimization of training methods mainly focuses on labeled datasets, unlabeled datasets, or the use of methods such as GAN to expand data, as well as the hybrid training method of unsupervised, semi-supervised, and supervised learning across the AL cycle. This training method promises to deliver even more performance improvements than are thought to be achievable through changes to the selection strategy. In fact, this makes up for the issues of the DL model requiring a large number of labeled training samples and the AL selecting a limited number of labeled samples. In addition, the use of unlabeled or generated datasets is also conducive to making full use of existing information without adding to the annotation costs. Furthermore, the incremental training method is also an important research direction. From a computing resource perspective, it is unacceptable to train a deep model from scratch in each cycle. While simple incremental training will cause the deviation of model parameters, the huge potential savings on resources are quite attractive. Although related research remains quite scarce, this is still a very promising research direction.

Task independence is also an important research direction, as it helps to make DeepAL models more directly and widely extensible to other tasks. However, the related research remains insufficient, and the corresponding DeepAL methods tend to focus only on the uncertainty-based selection method. Because DL itself is easier to integrate with the uncertainty-based AL selection strategy, we believe that uncertainty-based methods will continue to dominate research directions not related to these tasks in the future. On the other hand, it may also be advisable to explicitly take the diversity-based selection strategy into account; of course, this will also give rise to great challenges. In addition, it should be pointed out that blindly pursuing the idea of training models on smaller subsets would be unwise, as the relative difference in sample importance in some datasets with a large variety of content and a large number of samples can almost be ignored.

There is no conflict between the above-mentioned improvement directions; thus, a mixed improvement strategy is an important development direction for the future. In general, DeepAL research has significant practical application value in terms of both labeling costs and application scenarios; however, DeepAL research remains in its infancy at present, and there is still a long way to go in the future.

6 SUMMARY AND CONCLUSIONS

For the first time, the necessity and challenges of combining traditional active learning and deep learning have been comprehensively analyzed and summarized. In response to these challenges, we analyze and compare existing work from three perspectives: query strategy optimization, labeled sample data expansion, and model generality. In addition, we also summarize the stopping strategy of DeepAL. Then, we review the related work of DeepAL from the perspective of the application. Finally, we conduct a comprehensive discussion on the future direction of DeepAL. As far as we know, this is the first comprehensive and systematic review in the field of deep active learning.

ACKNOWLEDGMENTS

This work was partially supported by the NSFC under Grant (No.61972315 and No.62072372) and the Shaanxi Science and Technology Innovation Team Support Project under grant agreement (No.2018TD-026) and the Australian Research Council Discovery Early Career Researcher Award (No.DE190100626).

REFERENCES

[1] Mohammed Abdel-Wahab and Carlos Busso. 2019. Active Learning for Speech Emotion Recognition Using Deep Neural Network. In *8th International Conference on Affective Computing and Intelligent Interaction, ACII 2019, Cambridge, United Kingdom, September 3-6, 2019*. IEEE, 1–7.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749.

[3] Charu C. Aggarwal, Xiangnan Kong, Quanquan Gu, Jiawei Han, and Philip S. Yu. 2014. Active Learning: A Survey. In *Data Classification: Algorithms and Applications*. CRC Press, 571–606.

[4] Hamed Habibi Aghdam, Abel Gonzalez-Garcia, Antonio M. López, and Joost van de Weijer. 2019. Active Learning for Deep Detection Neural Networks. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 3671–3679.

[5] Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively Multilingual Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 3874–3884.

[6] Saeed S. Alahmari, Dmitry B. Goldgof, Lawrence O. Hall, and Peter R. Mouton. 2019. Automatic Cell Counting using Active Deep Learning and Unbiased Stereology. In *2019 IEEE International Conference on Systems, Man and Cybernetics, SMC 2019, Bari, Italy, October 6-9, 2019*. IEEE, 1708–1713.

[7] Bang An, Wenjun Wu, and Huimin Han. 2018. Deep Active Learning for Text Classification. In *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing, ICVISP 2018, Las Vegas, NV, USA, August 27-29, 2018*. ACM, 22:1–22:6.

[8] Olov Andersson, Mariusz Wzorek, and Patrick Doherty. 2017. Deep Learning Quadcopter Control via Risk-Aware Active Learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*. AAAI Press, 3812–3818.

[9] Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. 2001. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E* 64, 6 (2001), 061907.

[10] Dana Angluin. 1988. Queries and Concept Learning. *Machine Learning* 2, 4 (1988), 319–342.

[11] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*. IEEE Computer Society, 2425–2433.

[12] Samuel G Armato III, Geoffrey McLennan, Luc Bidaut, Michael F McNitt-Gray, Charles R Meyer, Anthony P Reeves, Binsheng Zhao, Denise R Aberle, Claudia I Henschke, Eric A Hoffman, et al. 2011. The lung image database consortium (LIDC) and image database resource initiative (IDRI): a completed reference database of lung nodules on CT scans. *Medical physics* 38, 2 (2011), 915–931.

[13] Nabiha Asghar, Pascal Poupart, Xin Jiang, and Hang Li. 2017. Deep Active Learning for Dialogue Generation. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics, *SEM @ACM 2017, Vancouver, Canada, August 3-4, 2017*. Association for Computational Linguistics, 78–83.

[14] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[15] Marc Bachlin, Meir Plotnik, Daniel Roggen, Inbal Maidan, Jeffrey M Hausdorff, Nir Giladi, and Gerhard Troster. 2009. Wearable assistant for Parkinson's disease patients with the freezing of gait symptom. *IEEE Transactions on Information Technology in Biomedicine* 14, 2 (2009), 436–446.

[16] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. 2017. Designing Neural Network Architectures using Reinforcement Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

[17] Mariaflorina Balcan, Alina Beygelzimer, and John Langford. 2009. Agnostic active learning. *J. Comput. System Sci.* 75, 1 (2009), 78–89.

[18] Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James R. Glass. 2019. Identifying and Controlling Important Neurons in Neural Machine Translation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[19] William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. 2018. The Power of Ensembles for Active Learning in Image Classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 9368–9377.

[20] Shai Bendavid, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. 2010. A theory of learning from different domains. *Machine Learning* 79, 1 (2010), 151–175.

[21] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. 2006. Greedy Layer-Wise Training of Deep Networks. (2006), 153–160.

[22] Sreyasee Das Bhattacharjee, Ashit Talukder, and Bala Venkatram Balantrapu. 2017. Active learning based news veracity detection with feature weighting and deep-shallow fusion. (2017), 556–565.

[23] Mustafa Bilgic and Lise Getoor. 2009. Link-based active learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*.

[24] John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

[25] Michael Bloodgood and Chris Callison-Burch. 2014. Bucking the Trend: Large-Scale Cost-Focused Active Learning for Statistical Machine Translation. *CoRR* abs/1410.5877 (2014).

[26] Michael Bloodgood and K. Vijay-Shanker. 2009. Taking into Account the Differences between Actively and Passively Acquired Data: The Case of Active Learning with Support Vector Machines for Imbalanced Datasets. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, May 31 - June 5, 2009, Boulder, Colorado, USA, Short Papers*. The Association for Computational Linguistics, 137–140.

[27] Michael Bloodgood and K. Vijay-Shanker. 2014. A Method for Stopping Active Learning Based on Stabilizing Predictions and the Need for User-Adjustable Stopping. *CoRR* abs/1409.5165 (2014).

[28] Erik Bochinski, Ghassen Bacha, Volker Eiselein, Tim J. W. Walles, Jens C. Nejstgaard, and Thomas Sikora. 2018. Deep Active Learning for In Situ Plankton Classification. In *Pattern Recognition and Information Forensics - ICPR 2018 International Workshops, CVAUI, IWCF, and MIPPSNA, Beijing, China, August 20-24, 2018, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 11188)*. Springer, 5–15.

[29] Klaus Brinker. 2003. Incorporating Diversity in Active Learning with Support Vector Machines. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*. AAAI Press, 59–66.

[30] Samuel Budd, Emma C. Robinson, and Bernhard Kainz. 2019. A Survey on Active Learning and Human-in-the-Loop Deep Learning for Medical Image Analysis. *CoRR* abs/1910.02923 (2019).

[31] Alex Burka and Katherine J. Kuchenbecker. 2017. How Much Haptic Surface Data Is Enough?. In *2017 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 27-29, 2017*. AAAI Press.

[32] Sylvain Calinon, Florent Guenter, and Aude Billard. 2007. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. *IEEE Trans. Syst. Man Cybern. Part B* 37, 2 (2007), 286–298.

[33] Trevor Campbell and Tamara Broderick. 2019. Automated Scalable Bayesian Inference via Hilbert Coresets. *Journal of Machine Learning Research* 20, 15 (2019), 1–38.

[34] Haw-Shiuan Chang, Shankar Vembu, Sunil Mohan, Rheeya Uppaal, and Andrew McCallum. 2020. Using error decay prediction to overcome practical issues of deep active learning for named entity recognition. *Mach. Learn.* 109, 9-10 (2020), 1749–1778.

[35] Bor-Chun Chen, Chu-Song Chen, and Winston H. Hsu. 2014. Cross-Age Reference Coding for Age-Invariant Face Recognition and Retrieval. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 8694)*. Springer, 768–783.

[36] Xuhui Chen, Jinlong Ji, Tianxi Ji, and Pan Li. 2018. Cost-Sensitive Deep Active Learning for Epileptic Seizure Detection. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, BCB 2018, Washington, DC, USA, August 29 - September 01, 2018*. ACM, 226–235.

[37] Anfeng Cheng, Chuan Zhou, Hong Yang, Jia Wu, Lei Li, Jianlong Tan, and Li Guo. 2019. Deep Active Learning for Anchor User Prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 2151–2157.

[38] Kashyap Chitta, Jose M Alvarez, Elmar Haussmann, and Clement Farabet. 2019. Training Data Distribution Search with Ensemble Active Learning. *arXiv preprint arXiv:1905.12737* (2019).

[39] Kashyap Chitta, Jose M. Alvarez, and Adam Lesnikowski. 2018. Large-Scale Visual Active Learning with Deep Probabilistic Ensembles. *CoRR* abs/1811.03575 (2018).

[40] Diane J Cook and Maureen Schmitter-Edgecombe. 2009. Assessing the quality of activities in a smart environment. *Methods of information in medicine* 48, 5 (2009), 480.

[41] Ido Dagan and Sean P. Engelson. 1995. Committee-Based Sampling For Training Probabilistic Classifiers. In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*. Morgan Kaufmann, 150–157.

[42] Navneet Dalal and Bill Triggs. 2005. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*. IEEE Computer Society, 886–893.

[43] Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in Imagined Conversations: A New Approach to Understanding Coordination of Linguistic Style in Dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics, CMCL@ACL 2011, Portland, Oregon, USA, June 23, 2011*. Association for Computational Linguistics, 76–87.

[44] Sanjoy Dasgupta. 2011. Two faces of active learning. *Theoretical Computer Science* 412, 19 (2011), 1767–1781.

[45] Sajib Dasgupta and Vincent Ng. 2009. Mine the Easy, Classify the Hard: A Semi-Supervised Approach to Automatic Sentiment Classification. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*, Keh-Yih Su, Jian Su, and Janyce Wiebe (Eds.). The Association for Computer Linguistics, 701–709. https://www.aclweb.org/anthology/P09-1079/

[46] Diana L. Delibaltov, Utkarsh Gaur, Jennifer Kim, Matthew Kourakis, Erin Newman-Smith, William Smith, Samuel A. Belteton, Daniel Szymanski, and B. S. Manjunath. 2016. CellECT: cell evolution capturing tool. *BMC Bioinform.* 17 (2016), 88.

[47] Diana L. Delibaltov, Pratim Ghosh, Volkan Rodoplu, Michael Veeman, William Smith, and B. S. Manjunath. 2013. A Linear Program Formulation for the Segmentation of Ciona Membrane Volumes. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013 - 16th International Conference, Nagoya, Japan, September 22-26, 2013, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 8149)*. Springer, 444–451.

[48] Cheng Deng, Yumeng Xue, Xianglong Liu, Chao Li, and Dacheng Tao. 2019. Active Transfer Learning Network: A Unified Deep Joint Spectral–Spatial Feature Learning Model for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* 57, 3 (2019), 1741–1754.

[49] Jia Deng, Alex Berg, Sanjeev Satheesh, Hao Su, Aditya Khosla, and Fei-Fei Li. 2012. Large scale visual recognition challenge. *www. image-net. org/challenges/LSVRC/2012* 1 (2012).

[50] Li Deng. 2014. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing* 3 (2014).

[51] Rezarta Islamaj Dogan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Informatics* 47 (2014), 1–10.

[52] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. 2012. Pedestrian Detection: An Evaluation of the State of the Art. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 4 (2012), 743–761.

[53] Jeff Donahue and Karen Simonyan. 2019. Large Scale Adversarial Representation Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 10541–10551.

[54] David L Donoho et al. 2000. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture* 1, 2000 (2000), 32.

[55] Baolin Du, Qi Qi, Han Zheng, Yue Huang, and Xinghao Ding. 2018. Breast Cancer Histopathological Image Classification via Deep Active Learning and Confidence Boosting. In *Artificial Neural Networks and Machine Learning - ICANN 2018 - 27th International Conference on Artificial Neural Networks, Rhodes, Greece, October 4-7, 2018, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 11140)*. Springer, 109–116.

[56] Xuefeng Du, Dexing Zhong, and Huikai Shao. 2019. Building an Active Palmprint Recognition System. In *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*. IEEE, 1685–1689.

[57] Melanie Ducoffe and Frédéric Precioso. 2018. Adversarial Active Learning for Deep Networks: a Margin Based Approach. *CoRR* abs/1802.09841 (2018).

[58] Rana El Kaliouby and Peter Robinson. 2004. Mind reading machines: Automated inference of cognitive mental states from video. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, Vol. 1. IEEE, 682–688.

[59] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision* 88, 2 (2010), 303–338.

[60] Caspar J. Fall, A. Törcsvári, K. Benzineb, and G. Karetka. 2003. Automated categorization in the international patent classification. *SIGIR Forum* 37, 1 (2003), 10–25.

[61] Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to Active Learn: A Deep Reinforcement Learning Approach. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Association for Computational Linguistics, 595–605.

[62] Reza Zanjirani Farahani and Masoud Hekmatfar. 2009. *Facility location: concepts, models, algorithms and case studies*. Springer.

[63] Di Feng, Xiao Wei, Lars Rosenbaum, Atsuto Maki, and Klaus Dietmayer. 2019. Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector. In *2019 IEEE Intelligent Vehicles Symposium, IV 2019, Paris, France, June 9-12, 2019*. IEEE, 667–674.

[64] Rosa L. Figueroa, Qing Zeng-Treitler, Long H. Ngo, Sergey Goryachev, and Eduardo P. Wiechmann. 2012. Active learning for clinical text classification: is it better than random sampling? *J. Am. Medical Informatics Assoc.* 19, 5 (2012), 809–816.

[65] Frederic Brenton Fitch. 1944. McCulloch Warren S. and Pitts Walter. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics , vol. 5 (1943), pp. 115–133. *Journal of Symbolic Logic* 9 (1944), 49–50.

[66] Jonathan Folmsbee, Xulei Liu, Margaret Brandwein-Weber, and Scott Doyle. 2018. Active deep learning: Improved training efficiency of convolutional neural networks for tissue classification in oral cavity cancer. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, 770–773.

[67] Tavis Forrester, William J McShea, RW Keys, Robert Costello, Megan Baker, and Arielle Parsons. 2013. eMammal–citizen science camera trapping as a solution for broad-scale, long-term monitoring of wildlife populations. *Sustainable Pathways: Learning from the Past and Shaping the Future* (2013).

[68] Marguerite Frank, Philip Wolfe, et al. 1956. An algorithm for quadratic programming. *Naval research logistics quarterly* 3, 1-2 (1956), 95–110.

[69] Alexander Freytag, Erik Rodner, and Joachim Denzler. 2014. Selecting Influential Examples: Active Learning with Expected Model Output Changes. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV (Lecture Notes in Computer Science, Vol. 8692)*. Springer, 562–577.

[70] Yarin Gal and Zoubin Ghahramani. 2015. Bayesian Convolutional Neural Networks with Bernoulli Approximate Variational Inference. *CoRR* abs/1506.02158 (2015).

[71] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*. JMLR.org, 1050–1059.

[72] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian Active Learning with Image Data. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*. PMLR, 1183–1192.

[73] Vijay Garla, Caroline Taylor, and Cynthia Brandt. 2013. Semi-supervised clinical text classification with Laplacian SVMs: An application to cancer case management. *J. Biomed. Informatics* 46, 5 (2013), 869–875.

[74] Utkarsh Gaur, Matthew Kourakis, Erin Newman-Smith, William Smith, and B. S. Manjunath. 2016. Membrane segmentation via active learning with deep networks. In *2016 IEEE International Conference on Image Processing, ICIP 2016, Phoenix, AZ, USA, September 25-28, 2016*. IEEE, 1943–1947.

[75] Yonatan Geifman and Ran El-Yaniv. 2017. Deep Active Learning over the Long Tail. *CoRR* abs/1711.00941 (2017).

[76] Yonatan Geifman and Ran El-Yaniv. 2019. Deep Active Learning with a Neural Architecture Search. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 5974–5984.

[77] Andreas Geiger, Philip Lenz, and Raquel Urtasun. [n.d.]. Are we ready for autonomous driving. In *Proc. CVPR*. 3354–3361.

[78] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*. IEEE Computer Society, 3354–3361.

[79] Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative Active Learning. *CoRR* abs/1907.06347 (2019).

[80] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2672–2680.

[81] Gregory Griffin, Alex Holub, and Pietro Perona. 2007. Caltech-256 object category dataset. (2007).

[82] Denis A. Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. 2020. Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 9038–9046.

[83] Gautham Krishna Gudur, Prahalathan Sundaramoorthy, and Venkatesh Umaashankar. 2019. ActiveHARNet: Towards On-Device Deep Bayesian Active Learning for Human Activity Recognition. *CoRR* abs/1906.00108.

[84] Yuhong Guo. 2010. Active Instance Sampling via Matrix Partition. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*. Curran Associates, Inc., 802–810.

[85] Kazim Hanbay. 2019. Deep Neural Network Based Approach for ECG Classification Using Hybrid Differential Features and Active Learning. *Iet Signal Processing* 13, 2 (2019), 165–175.
[86] Manuel Haußmann, Fred A. Hamprecht, and Melih Kandemir. 2019. Deep Active Learning with Adaptive Acquisition. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 2470–2476.
[87] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 770–778.
[88] Tao He, Xiaoming Jin, Guiguang Ding, Lan Yi, and Chenggang Yan. 2019. Towards Better Uncertainty Sampling: Active Learning with Multiple Views for Deep Convolutional Neural Network. In *IEEE International Conference on Multimedia and Expo, ICME 2019, Shanghai, China, July 8-12, 2019*. IEEE, 1360–1365.
[89] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. 2019. Bag of Tricks for Image Classification with Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 558–567.
[90] José Miguel Hernández-Lobato and Ryan P. Adams. 2015. Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015 (JMLR Workshop and Conference Proceedings, Vol. 37)*. JMLR.org, 1861–1869.
[91] Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. 2006. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* 18, 7 (2006), 1527–1554.
[92] Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580 (2012).
[93] Martin Hirzer, Csaba Beleznai, Peter M Roth, and Horst Bischof. 2011. Person re-identification by descriptive and discriminative classification. In *Scandinavian conference on Image analysis*. Springer, 91–102.
[94] Steven C. H. Hoi, Rong Jin, Jianke Zhu, and Michael R. Lyu. 2006. Batch mode active learning and its application to medical image classification. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006 (ACM International Conference Proceeding Series, Vol. 148)*. ACM, 417–424.
[95] H. M. Sajjad Hossain, M. D. Abdullah Al Haiz Khan, and Nirmalya Roy. 2018. DeActive: Scaling Activity Recognition with Active Deep Learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 2 (2018), 66:1–66:23.
[96] H. M. Sajjad Hossain and Nirmalya Roy. 2019. Active Deep Learning for Activity Recognition with Context Aware Annotator Selection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. ACM, 1862–1870.
[97] Neil Houlsby, Ferenc Huszar, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian Active Learning for Classification and Preference Learning. *CoRR* abs/1112.5745 (2011).
[98] Yue Huang, Zhenwei Liu, Minghui Jiang, Xian Yu, and Xinghao Ding. 2020. Cost-Effective Vehicle Type Recognition in Surveillance Images With Deep Active Learning and Web Data. *IEEE Transactions on Intelligent Transportation Systems* 21, 1 (2020), 79–86.
[99] Jonathan H Huggins, Trevor Campbell, and Tamara Broderick. 2016. Coresets for Scalable Bayesian Logistic Regression. (2016), 4080–4088.
[100] Ahmed Hussein, Mohamed Medhat Gaber, and Eyad Elyan. 2016. Deep Active Learning for Autonomous Navigation. In *Engineering Applications of Neural Networks - 17th International Conference, EANN 2016, Aberdeen, UK, September 2-5, 2016, Proceedings (Communications in Computer and Information Science, Vol. 629)*. Springer, 3–17.
[101] Rania Ibrahim, Noha A Yousri, Mohamed A Ismail, and Nagwa M El-Makky. 2014. Multi-level gene/MiRNA feature selection using deep belief nets and active learning. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 3957–3960.
[102] Prateek Jain and Ashish Kapoor. 2009. Active learning for large multi-class problems. (2009), 762–769.
[103] David Janz, Jos van der Westhuizen, and José Miguel Hernández-Lobato. 2017. Actively Learning what makes a Discrete Sequence Valid. *CoRR* abs/1708.04465 (2017).
[104] Khaled Jedoui, Ranjay Krishna, Michael Bernstein, and Fei-Fei Li. 2019. Deep Bayesian Active Learning for Multiple Correct Outputs. *CoRR* abs/1912.01119 (2019).
[105] Michael I. Jordan. 1986. Serial Order: A Parallel Distributed Processing Approach. *Advances in psychology* 121 (1986), 471–495.
[106] Ajay Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. 2009. Multi-class active learning for image classification. (2009), 2372–2379.
[107] J. Ajay Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. 2010. Multi-class batch-mode active learning for image classification. *Robotics and Automation* (2010), 1873–1878.

[108] Takeo Kanade, Jeffrey F Cohn, and Yingli Tian. 2000. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*. IEEE, 46–53.

[109] Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource Deep Entity Resolution with Transfer and Active Learning. (2019), 5851–5861.

[110] Benjamin Kellenberger, Diego Marcos, Sylvain Lobry, and Devis Tuia. 2019. Half a Percent of Labels is Enough: Efficient Animal Detection in UAV Imagery Using Deep CNNs and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing* 57, 12 (2019), 9524–9533.

[111] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. 2017. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. In *British Machine Vision Conference 2017, BMVC 2017, London, UK, September 4-7, 2017*. BMVA Press.

[112] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. 2020. Task-Aware Variational Adversarial Active Learning. *arXiv: Learning* (2020).

[113] Ross D King, Kenneth E Whelan, Ffion M Jones, Philip G K Reiser, Christopher H Bryant, Stephen Muggleton, Douglas B Kell, and Stephen G Oliver. 2004. Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 427, 6971 (2004), 247–252.

[114] Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

[115] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. 2019. BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 7024–7035.

[116] Xiangnan Kong, Jiawei Zhang, and Philip S. Yu. 2013. Inferring anchor links across multiple heterogeneous social networks. In *22nd ACM International Conference on Information and Knowledge Management, CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013*. ACM, 179–188.

[117] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *Int. J. Comput. Vis.* 123, 1 (2017), 32–73.

[118] Vikram Krishnamurthy. 2002. Algorithms for optimal scheduling and management of hidden Markov model sensors. *IEEE Transactions on Signal Processing* 50, 6 (2002), 1382–1397.

[119] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. Citeseer.

[120] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. (2012), 1097–1105.

[121] M P Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models. (2010), 1189–1197.

[122] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel Moore. 2010. Activity recognition using cell phone accelerometers. *SIGKDD Explor.* 12, 2 (2010), 74–82.

[123] Bogdan Kwolek, Michal Koziarski, Andrzej Bukala, Zbigniew Antosz, Boguslaw Olborski, Pawel Wąsowicz, Jakub Swadźba, and Boguslaw Cyganek. 2019. Breast Cancer Classification on Histopathological Images Affected by Data Imbalance Using Active Learning and Deep Convolutional Neural Network. (2019), 299–312.

[124] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor detection over varying time windows. *PloS one* 12, 1 (2017), e0168344.

[125] Leah S. Larkey. 1999. A Patent Search and Classification System. In *Proceedings of the Fourth ACM conference on Digital Libraries, August 11-14, 1999, Berkeley, CA, USA*. ACM, 179–187.

[126] Yann Lecun, Yoshua Bengio, and Geoffrey E Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[127] Y. LeCun, B. Boser, S. J. Denker, D. Henderson, E. R. Howard, W. Hubbard, and D. L. Jackel. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation* (1989), 541–551.

[128] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[129] Byungjae Lee and Kyunghyun Paeng. 2018. A Robust and Effective Approach Towards Accurate Metastasis Detection and pN-stage Classification in Breast Cancer. (2018), 841–850.

[130] Christian Leibig, Vaneeda Allken, Murat Seckin Ayhan, Philipp Berens, and Siegfried Wahl. 2017. Leveraging uncertainty information from deep neural networks for disease detection. *Scientific Reports* 7, 1 (2017), 17816–17816.

[131] David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. (1994), 3–12.

[132] Ya Li, Keze Wang, Lin Nie, and Qing Wang. 2017. Face Recognition via Heuristic Deep Active Learning. (2017), 97–107.

[133] Jianzhe Lin, Liang Zhao, Shuying Li, Rabab K Ward, and Z Jane Wang. 2018. Active-Learning-Incorporated Deep Transfer Learning for Hyperspectral Image Classification. *IEEE Journal of Selected Topics in Applied Earth Observations*

and Remote Sensing 11, 11 (2018), 4048–4062.
[134] Xiao Lin and Devi Parikh. 2017. Active Learning for Visual Question Answering: An Empirical Study. *CoRR* abs/1711.01732 (2017).
[135] Peng Liu, Hui Zhang, and Kie B Eom. 2017. Active Deep Learning for Classification of Hyperspectral Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10, 2 (2017), 712–724.
[136] Zimo Liu, Jingya Wang, Shaogang Gong, Huchuan Lu, and Dacheng Tao. 2019. Deep Reinforcement Active Learning for Human-in-the-Loop Person Re-Identification. (2019), 6122–6131.
[137] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. (2015), 3431–3440.
[138] Reza Lotfian and Carlos Busso. 2017. Building naturalistic emotionally balanced speech corpus by retrieving emotional speech from existing podcast recordings. *IEEE Transactions on Affective Computing* (2017).
[139] David G. Lowe. 1999. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision, Kerkyra, Corfu, Greece, September 20-25, 1999*. 1150–1157.
[140] Xiaoming Lv, Fajie Duan, Jiajia Jiang, Xiao Fu, and Lin Gan. 2020. Deep Active Learning for Surface Defect Detection. *Sensors* 20, 6 (2020), 1650.
[141] Ramon Maldonado and Sanda M Harabagiu. 2019. Active deep learning for the identification of concepts and relations in electroencephalography reports. *Journal of Biomedical Informatics* 98 (2019), 103265.
[142] RG Mark, PS Schluter, G Moody, P Devlin, and D Chernoff. 1982. An annotated ECG database for evaluating arrhythmia detectors. In *IEEE Transactions on Biomedical Engineering*, Vol. 29. IEEE-INST ELECTRICAL ELECTRONICS ENGINEERS INC 345 E 47TH ST, NEW YORK, NY ..., 600–600.
[143] Giovanna Martínez-Arellano and Svetan M. Ratchev. 2019. Towards An Active Learning Approach To Tool Condition Monitoring With Bayesian Deep Learning. In *Proceedings of the 33rd International ECMS Conference on Modelling and Simulation, ECMS 2019 Caserta, Italy, June 11-14, 2019*. European Council for Modeling and Simulation, 223–229.
[144] Muhammad Mateen, Junhao Wen, Nasrullah, Sun Song, and Zhouping Huang. 2019. Fundus Image Classification Using VGG-19 Architecture with PCA and SVD. *Symmetry* 11 (2019), 1.
[145] Taylor R. Mauldin, Marc E. Canby, Vangelis Metsis, Anne H. H. Ngu, and Coralys Cubero Rivera. 2018. SmartFall: A Smartwatch-Based Fall Detection System Using Deep Learning. *Sensors* 18, 10 (2018), 3363.
[146] Ali Mottaghi and Serena Yeung. 2019. Adversarial Representation Active Learning. *CoRR* abs/1912.09720 (2019).
[147] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. In *Proceedings of the 2018 International Conference on Management of Data, SIGMOD Conference 2018, Houston, TX, USA, June 10-15, 2018*. ACM, 19–34.
[148] Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. 2020. Towards Robust and Reproducible Active Learning Using Neural Networks. *arXiv* (2020), arXiv–2002.
[149] Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical Losses and New Resources for Fine-grained Entity Typing and Linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Association for Computational Linguistics, 97–109.
[150] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *ICML*.
[151] Ju Gang Nam, Sunggyun Park, Eui Jin Hwang, Jong Hyuk Lee, Kwang Nam Jin, Kun Young Lim, Thienkai Huy Vu, Jae Ho Sohn, Sangheum Hwang, Jin Mo Goo, et al. 2019. Development and Validation of Deep Learning–based Automatic Detection Algorithm for Malignant Pulmonary Nodules on Chest Radiographs. *Radiology* 290, 1 (2019), 218–228.
[152] Ali Bou Nassif, Ismail Shahin, Imtinan B. Attili, Mohammad Azzeh, and Khaled Shaalan. 2019. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* 7 (2019), 19143–19165.
[153] T. Hieu Nguyen and Arnold Smeulders. 2004. Active learning using pre-clustering. *ICML* (2004), 79–79.
[154] Mohammad Sadegh Norouzzadeh, Dan Morris, Sara Beery, Neel Joshi, Nebojsa Jojic, and Jeff Clune. 2019. A deep active learning system for species identification and counting in camera trap images. *CoRR* abs/1910.09716 (2019).
[155] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional Image Synthesis With Auxiliary Classifier GANs. (2017), 2642–2651.
[156] Natalia Ostapuk, Jie Yang, and Philippe Cudre-Mauroux. 2019. ActiveLink: Deep Active Learning for Link Prediction in Knowledge Graphs. *WWW '19: The Web Conference on The World Wide Web Conference WWW 2019* (2019), 1398–1408.
[157] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing, EMNLP 2002, Philadelphia, PA, USA, July 6-7, 2002*. 79–86.

[158] Mercedes Eugenia Paoletti, Juan Mario Haut, Rubén Fernández-Beltran, Javier Plaza, Antonio J. Plaza, Jun Yu Li, and Filiberto Pla. 2019. Capsule Networks for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* 57 (2019), 2145–2160.

[159] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*. ISCA, 2613–2617.

[160] John P. Pestian, Chris Brew, Pawel Matykiewicz, D. J. Hovermale, Neil Johnson, K. Bretonnel Cohen, and Wlodzislaw Duch. 2007. A shared task involving multi-label classification of clinical free text. In *Biological, translational, and clinical language processing, BioNLP@ACL 2007, Prague, Czech Republic, June 29, 2007*. Association for Computational Linguistics, 97–104.

[161] Jeff M Phillips. 2016. Coresets and sketches. *arXiv preprint arXiv:1601.00617* (2016).

[162] Robert Pinsler, Jonathan Gordon, Eric Nalisnick, and Jose Miguel Hernandezlobato. 2019. Bayesian Batch Active Learning as Sparse Subset Approximation. (2019), 6356–6367.

[163] Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom M. Mitchell. 2018. Contextual Parameter Generation for Universal Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 425–435.

[164] Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848* (2019).

[165] Remus Pop and Patric Fulop. 2018. Deep Ensemble Bayesian Active Learning : Addressing the Mode Collapse issue in Monte Carlo dropout via Ensembles. *CoRR* abs/1811.03897 (2018).

[166] Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling Bias in Deep Active Classification: An Empirical Study. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*. Association for Computational Linguistics, 4056–4066.

[167] Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards Robust Linguistic Analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*. ACL, 143–152.

[168] Shalini Priya, Saharsh Singh, Sourav Kumar Dandapat, Kripabandhu Ghosh, and Joydeep Chandra. 2019. Identifying infrastructure damage during earthquake using deep active learning. (2019), 551–552.

[169] Yao Qin, Nicholas Carlini, Ian J. Goodfellow, Garrison W. Cottrell, and Colin Raffel. 2019. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. *ArXiv* abs/1903.10346 (2019).

[170] Zhenshen Qu, Jingda Du, Yong Cao, Qiuyu Guan, and Pengbo Zhao. 2020. Deep Active Learning for Remote Sensing Object Detection. *CoRR* abs/2003.08793 (2020).

[171] M M Al Rahhal, Yakoub Bazi, Haikel Alhichri, Naif Alajlan, Farid Melgani, and Ronald R Yager. 2016. Deep learning approach for active classification of electrocardiogram signals. *Information Sciences* 345, 345 (2016), 340–354.

[172] Hiranmayi Ranganathan, Shayok Chakraborty, and Sethuraman Panchanathan. 2016. Multimodal emotion recognition using deep learning architectures. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 1–9.

[173] Hiranmayi Ranganathan, Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. 2017. Deep active learning for image classification. (2017), 3934–3938.

[174] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Xiaojiang Chen, and Xin Wang. 2020. A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions. *CoRR* abs/2006.02903 (2020).

[175] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. (2015), 234–241.

[176] Matthias Rottmann, Karsten Kahl, and Hanno Gottschalk. 2018. Deep Bayesian Active Semi-Supervised Learning. In *17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, Orlando, FL, USA, December 17-20, 2018*. IEEE, 158–164.

[177] Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown* (2001), 441–448.

[178] Soumya Roy, Asim Unmesh, and Vinay P. Namboodiri. 2018. Deep active learning for object detection. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 91.

[179] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning internal representations by error propagation. *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1* (1986), 318–362.

[180] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536.

[181] Ario Sadafi, Niklas Koehler, Asya Makhro, Anna Bogdanova, Nassir Navab, Carsten Marr, and Tingying Peng. 2019. Multiclass Deep Active Learning for Detecting Red Blood Cell Subtypes in Brightfield Microscopy. (2019), 685–693.
[182] Mathew Salvaris, Danielle Dean, and Wee Hyong Tok. 2018. Generative Adversarial Networks. *arXiv: Machine Learning* (2018), 187–208.
[183] Conrad Sanderson. 2008. *Biometric person recognition: Face, speech and fusion*. Vol. 4. VDM Publishing.
[184] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*. ACL, 142–147.
[185] Yassir Saquil, Kwang In Kim, and Peter Hall. 2018. Ranking CGANs: Subjective Control over Semantic Image Attributes. (2018), 131.
[186] G Sayantan, P T Kien, and K V Kadambari. 2018. Classification of ECG beats using deep belief network and active learning. *Medical & Biological Engineering & Computing* 56, 10 (2018), 1887–1898.
[187] Melanie Lubrano Di Scandalea, Christian S Perone, Mathieu Boudreau, and Julien Cohenadad. 2019. Deep Active Learning for Axon-Myelin Segmentation on Histology Data. *arXiv: Computer Vision and Pattern Recognition* (2019).
[188] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active Hidden Markov Models for Information Extraction. In *Advances in Intelligent Data Analysis, 4th International Conference, IDA 2001, Cascais, Portugal, September 13-15, 2001, Proceedings (Lecture Notes in Computer Science, Vol. 2189)*. Springer, 309–318.
[189] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised Pre-training for Speech Recognition. *ArXiv* abs/1904.05862 (2019).
[190] Christopher Schröder and Andreas Niekler. 2020. A Survey of Active Learning for Text Classification using Deep Neural Networks. *arXiv preprint arXiv:2008.07267* (2020).
[191] Ozan Sener and Silvio Savarese. 2017. A geometric approach to active learning for convolutional neural networks. *arXiv preprint arXiv:1708.00489* 7 (2017).
[192] Ozan Sener and Silvio Savarese. 2018. Active Learning for Convolutional Neural Networks: A Core-Set Approach. *international conference on learning representations* (2018).
[193] Burr Settles. 2009. *Active learning literature survey*. Technical Report. University of Wisconsin-Madison Department of Computer Sciences.
[194] Burr Settles. 2012. Active Learning, volume 6 of Synthesis Lectures on Artificial Intelligence and Machine Learning. *Morgan & Claypool* (2012).
[195] Burr Settles, Mark Craven, and Soumya Ray. 2007. Multiple-Instance Active Learning. (2007), 1289–1296.
[196] H S Seung, M Opper, and H Sompolinsky. 1992. Query by committee. (1992), 287–294.
[197] Matthew Shardlow, Meizhi Ju, Maolin Li, Christian O'Reilly, Elisabetta Iavarone, John McNaught, and Sophia Ananiadou. 2019. A text mining pipeline using active and deep learning aimed at curating information in computational neuroscience. *Neuroinformatics* 17, 3 (2019), 391–406.
[198] Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V. Dylov. 2019. Active Learning with Deep Pre-trained Models for Sequence Tagging of Clinical and Biomedical Texts. In *2019 IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2019, San Diego, CA, USA, November 18-21, 2019*. IEEE, 482–489.
[199] Yanyao Shen, Hyokun Yun, Zachary C Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. *arXiv preprint arXiv:1707.05928* (2017).
[200] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. 2020. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 1308–1318.
[201] Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian Active Learning for Natural Language Processing: Results of a Large-Scale Empirical Study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Association for Computational Linguistics, 2904–2909.
[202] Oriane Simeoni, Mateusz Budnik, Yannis Avrithis, and Guillaume Gravier. 2019. Rethinking deep active learning: Using unlabeled data at model training. *CoRR* abs/1911.08177 (2019).
[203] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
[204] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. 2019. Variational Adversarial Active Learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*. IEEE, 5971–5980.
[205] Asim Smailagic, Pedro Costa, Alex Gaudio, Kartik Khandelwal, Mostafa Mirshekari, Jonathon Fagert, Devesh Walawalkar, Susu Xu, Adrian Galdran, Pei Zhang, et al. 2020. O-MedAL: Online active deep learning for medical image analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 10, 4 (2020), e1353.

[206] Asim Smailagic, Pedro Costa, Hae Young Noh, Devesh Walawalkar, Kartik Khandelwal, Adrian Galdran, Mostafa Mirshekari, Jonathon Fagert, Susu Xu, Pei Zhang, et al. 2018. MedAL: Accurate and Robust Deep Active Learning for Medical Image Analysis. (2018), 481–488.

[207] Justin S Smith, Benjamin Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E Roitberg. 2018. Less is more: Sampling chemical space with active learning. *Journal of Chemical Physics* 148, 24 (2018), 241733.

[208] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. 2015. Learning structured output representation using deep conditional generative models. (2015), 3483–3491.

[209] Kechen Song and Yunhui Yan. 2013. A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science* 285 (2013), 858–864.

[210] Fabio A. Spanhol, Luiz S. Oliveira, Caroline Petitjean, and Laurent Heutte. 2016. A Dataset for Breast Cancer Histopathological Image Classification. *IEEE Trans. Biomed. Eng.* 63, 7 (2016), 1455–1462.

[211] Akash Srivastava, Lazar Valkoz, Chris Russell, U. Michael Gutmann, and A. Charles Sutton. 2017. VEEGAN: Reducing Mode Collapse in GANs using Implicit Variational Learning. *neural information processing systems* (2017), 3310–3320.

[212] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[213] Fabian Stark, Caner Hazırbas, Rudolph Triebel, and Daniel Cremers. 2015. Captcha recognition with active deep learning. In *Workshop new challenges in neural computation*, Vol. 2015. Citeseer, 94.

[214] Allan Stisen, Henrik Blunck, Sourav Bhattacharya, Thor Siiger Prentow, Mikkel Baun Kjærgaard, Anind Dey, Tobias Sonne, and Mads Møller Jensen. 2015. Smart devices are different: Assessing and mitigatingmobile sensing heterogeneities for activity recognition. In *Proceedings of the 13th ACM conference on embedded networked sensor systems*. 127–140.

[215] Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. 2015. Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. *Scientific data* 2, 1 (2015), 1–14.

[216] Kuniyuki Takahashi, Tetsuya Ogata, Jun Nakanishi, Gordon Cheng, and Shigeki Sugano. 2017. Dynamic motion learning for multi-DOF flexible-joint robots using active–passive motor babbling through deep learning. *Advanced Robotics* 31, 18 (2017), 1002–1015.

[217] Xu Tan, Yi Ren, Di He, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2019. Multilingual Neural Machine Translation with Knowledge Distillation. *ArXiv* abs/1902.10461 (2019).

[218] Yao Tan, Liu Yang, Qinghua Hu, and Zhibin Du. 2019. Batch Mode Active Learning for Semantic Segmentation Based on Multi-Clue Sample Selection. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 831–840.

[219] Joseph Taylor, H. M. Sajjad Hossain, Mohammad Arif Ul Alam, Md Abdullah Al Hafiz Khan, Nirmalya Roy, Elizabeth Galik, and Aryya Gangopadhyay. 2017. SenseBox: A low-cost smart home system. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017, Kona, Big Island, HI, USA, March 13-17, 2017*. IEEE, 60–62.

[220] Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*. European Language Resources Association (ELRA), 2214–2218.

[221] Simon Tong. 2001. *Active learning: theory and applications*. Vol. 1. Stanford University USA.

[222] Simon Tong and Daphne Koller. 2002. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research* 2, 1 (2002), 45–66.

[223] Toan Tran, Thanh-Toan Do, Ian D. Reid, and Gustavo Carneiro. 2019. Bayesian Generative Active Deep Learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA (Proceedings of Machine Learning Research, Vol. 97)*. PMLR, 6295–6304.

[224] Toan Tran, Trung Pham, Gustavo Carneiro, Lyle Palmer, and Ian Reid. 2017. A bayesian data augmentation approach for learning deep models. In *Advances in neural information processing systems*. 2797–2806.

[225] Kailas Vodrahalli, Ke Li, and Jitendra Malik. 2018. Are All Training Examples Created Equal? An Empirical Study. *CoRR* abs/1811.12569 (2018).

[226] Byron C. Wallace, Michael J. Paul, Urmimala Sarkar, Thomas A. Trikalinos, and Mark Dredze. 2014. A large-scale quantitative analysis of latent factors and sentiment in online doctor reviews. *J. Am. Medical Informatics Assoc.* 21, 6 (2014), 1098–1103.

[227] Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. (2014), 112–119.

[228] Jiannan Wang, Guoliang Li, Jeffrey Xu Yu, and Jianhua Feng. 2011. Entity Matching: How Similar Is Similar. *Proc. VLDB Endow.* 4, 10 (2011), 622–633.

[229] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin. 2017. Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology* 27, 12 (2017), 2591–2600.

[230] Menglin Wang, Baisheng Lai, Zhongming Jin, Xiaojin Gong, Jianqiang Huang, and Xiansheng Hua. 2018. Deep active learning for video-based person re-identification. *arXiv preprint arXiv:1812.05785* (2018).

[231] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. 2019. Learning Deep Transformer Models for Machine Translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. Association for Computational Linguistics, 1810–1822.

[232] Wenzhe Wang, Ruiwei Feng, Jintai Chen, Yifei Lu, Tingting Chen, Hongyun Yu, Danny Z Chen, and Jian Wu. 2019. Nodule-Plus R-CNN and Deep Self-Paced Active Learning for 3D Instance Segmentation of Pulmonary Nodules. *IEEE Access* 7 (2019), 128796–128805.

[233] Wenzhe Wang, Yifei Lu, Bian Wu, Tingting Chen, Danny Z Chen, and Jian Wu. 2018. Deep Active Self-paced Learning for Accurate Pulmonary Nodule Segmentation. (2018), 723–731.

[234] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, Regina Barzilay and Min-Yen Kan (Eds.). Association for Computational Linguistics, 422–426.

[235] Zengmao Wang, Bo Du, Lefei Zhang, and Liangpei Zhang. 2016. A batch-mode active learning framework by querying discriminative and representative samples for hyperspectral image classification. *Neurocomputing* 179 (2016), 88–100.

[236] Max Welling and Whye Yee Teh. 2011. Bayesian Learning via Stochastic Gradient Langevin Dynamics. *ICML* (2011), 681–688.

[237] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wanli Ouyang, and Yi Yang. 2018. Exploit the Unknown Gradually: One-Shot Video-Based Person Re-Identification by Stepwise Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. IEEE Computer Society, 5177–5186.

[238] Xide Xia, Pavlos Protopapas, and Finale Doshivelez. 2016. Cost-Sensitive Batch Mode Active Learning: Designing Astronomical Observation by Optimizing Telescope Time and Telescope Choice. (2016), 477–485.

[239] Ismet Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Kumar Mahajan. 2019. Billion-scale semi-supervised learning for image classification. *ArXiv* abs/1905.00546 (2019).

[240] Yilin Yan, Min Chen, Saad Sadiq, and Mei-Ling Shyu. 2017. Efficient imbalanced multimedia concept retrieval by deep learning on spark clusters. *International Journal of Multimedia Data Engineering and Management (IJMDEM)* 8, 1 (2017), 1–20.

[241] Yilin Yan, Min Chen, Mei-Ling Shyu, and Shu-Ching Chen. 2015. Deep learning for imbalanced multimedia data classification. In *2015 IEEE international symposium on multimedia (ISM)*. IEEE, 483–488.

[242] Jie Yang, Thomas Drake, Andreas Damianou, and Yoelle Maarek. 2018. Leveraging Crowdsourcing Data For Deep Active Learning – An Application: Learning Intents in Alexa. (2018), 23–32.

[243] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. 2017. Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation. (2017), 399–407.

[244] C. Yin, B. Qian, S. Cao, X. Li, J. Wei, Q. Zheng, and I. Davidson. 2017. Deep Similarity-Based Batch Mode Active Learning with Exploration-Exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*. 575–584.

[245] Donggeun Yoo and In So Kweon. 2019. Learning Loss for Active Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. Computer Vision Foundation / IEEE, 93–102.

[246] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. 2018. BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling. *CoRR* abs/1805.04687 (2018).

[247] Jiaming Zeng, Adam Lesnikowski, and Jose M. Alvarez. 2018. The Relevance of Bayesian Layer Positioning to Model Uncertainty in Deep Bayesian Active Learning. *CoRR* abs/1811.12535 (2018).

[248] Pei Zhang, Xueying Xu, and Deyi Xiong. 2018. Active Learning for Neural Machine Translation. In *2018 International Conference on Asian Language Processing, IALP 2018, Bandung, Indonesia, November 15-17, 2018*. IEEE, 153–158.

[249] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. 2017. CityPersons: A Diverse Dataset for Pedestrian Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 4457–4465.

[250] Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 649–657.

[251] Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active Discriminative Text Representation Learning. (2017), 3386–3392.

[252] Yizhe Zhang, Michael T. C. Ying, Lin Yang, Anil T. Ahuja, and Danny Z. Chen. 2016. Coarse-to-Fine Stacked Fully Convolutional Nets for lymph node segmentation in ultrasound images. In *IEEE International Conference on Bioinformatics and Biomedicine, BIBM 2016, Shenzhen, China, December 15-18, 2016*. IEEE Computer Society, 443–448. https://doi.org/10.1109/BIBM.2016.7822557

[253] Wencang Zhao, Yu Kong, Zhengming Ding, and Yun Fu. 2017. Deep Active Learning Through Cognitive Information Parcels. (2017), 952–960.

[254] Ziyuan Zhao, Xiaoyan Yang, Bharadwaj Veeravalli, and Zeng Zeng. 2020. Deeply Supervised Active Learning for Finger Bones Segmentation. *arXiv preprint arXiv:2005.03225* (2020).

[255] Fedor Zhdanov. 2019. Diverse mini-batch Active Learning. *CoRR* abs/1901.05954 (2019).

[256] Liang Zheng, Zhi Bie, Yifan Sun, Jingdong Wang, Chi Su, Shengjin Wang, and Qi Tian. 2016. MARS: A Video Benchmark for Large-Scale Person Re-Identification. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI (Lecture Notes in Computer Science, Vol. 9910)*. Springer, 868–884.

[257] Shusen Zhou, Qingcai Chen, and Xiaolong Wang. 2010. Active Deep Networks for Semi-Supervised Sentiment Classification. (2010), 1515–1523.

[258] Siqi Zhou and Angela P Schoellig. 2019. Active Training Trajectory Generation for Inverse Dynamics Model Learning with Deep Neural Networks. In *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 1784–1790.

[259] Jia-Jie Zhu and José Bento. 2017. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956* (2017).

[260] Xiaojin Zhu, John Lafferty, and Ronald Rosenfeld. 2005. *Semi-supervised learning with graphs*. Ph.D. Dissertation. Carnegie Mellon University, language technologies institute, school of ....

[261] Michal Ziemski, Marcin Junczys-Dowmunt, and Bruno Pouliquen. 2016. The United Nations Parallel Corpus v1.0. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

[262] Barret Zoph and Quoc V. Le. 2017. Neural Architecture Search with Reinforcement Learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

State-Relabeling Adversarial Active Learning

Beichen Zhang[1], Liang Li[2]*, Shijie Yang[1, 2], Shuhui Wang[2], Zheng-Jun Zha[3], Qingming Huang[1, 2, 4]
[1]University of Chinese Academy of Sciences, Beijing, China
[2]Key Lab of Intell. Info. Process., Inst. of Comput. Tech., Chinese Academy of Sciences, China
[3]University of Science and Technology of China, China, [4]Peng Cheng Laboratory, Shenzhen, China,
{beichen.zhang, shijie.yang}@vipl.ict.ac.cn, {liang.li, wangshuhui}@ict.ac.cn,
zhazj@ustc.edu.cn, qmhuang@ucas.ac.cn

Abstract

Active learning is to design label-efficient algorithms by sampling the most representative samples to be labeled by an oracle. In this paper, we propose a state relabeling adversarial active learning model (SRAAL), that leverages both the annotation and the labeled/unlabeled state information for deriving the most informative unlabeled samples. The SRAAL consists of a representation generator and a state discriminator. The generator uses the complementary annotation information with traditional reconstruction information to generate the unified representation of samples, which embeds the semantic into the whole data representation. Then, we design an online uncertainty indicator in the discriminator, which endues unlabeled samples with different importance. As a result, we can select the most informative samples based on the discriminator's predicted state. We also design an algorithm to initialize the labeled pool, which makes subsequent sampling more efficient. The experiments conducted on various datasets show that our model outperforms the previous state-of-art active learning methods and our initially sampling algorithm achieves better performance.

1. Introduction

Although deep neural network models have made great success in many areas, they still heavily rely on large-scale labeled data to train large number of parameters. Unfortunately, it is very difficult, time-consuming, or expensive to obtain labeled samples, which becomes the main bottleneck for deep learning methods [10]. To reduce the demand of labeled data, learning methods like unsupervised learning [6, 35], semi-supervised learning [19, 30], weakly supervised learning [46, 28] and active learning have attracted a lot of attention. Unsupervised and semi-supervised methods aim to fully utilize the unlabeled samples while active learning is to select as few samples to be labeled as possible for efficient training. This paper focuses on active learning, which is widely used in computer vision tasks such as classification [37, 2] and segmentation [42, 17].

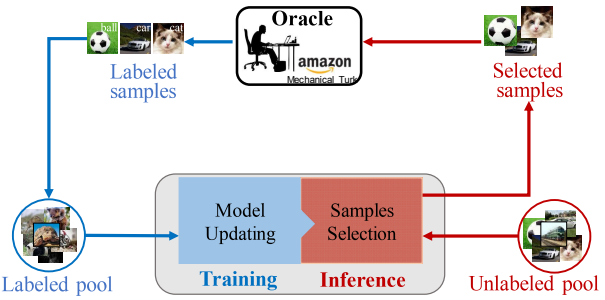


Figure 1. A traditional pool-based active learning cycle. At each iteration, the sampling model is trained with labeled data. After training, a subset of unlabeled samples is selected based on the model inference and then labeled by an oracle. The active learning system will repeat this iteration until the model performance meets user's requirements or the label budget runs out.

In active learning, this means the selected samples should be the most informative ones. As shown in Fig. 1, active learning algorithm is typically an iterative process in which a set of samples is selected to be labeled from an unlabeled pool at each iteration. These selected unlabeled subsets are labeled by an oracle, integrated into the labeled data pool. How to select the most informative samples from the unlabeled pool is the key problem in active learning.

To solve this problem, previous works have made full use of the annotation information of labeled data from the oracle. Many methods [2, 37, 14, 42] built deep learning models, trained them under the supervision of the labeled

*Corresponding author

samples, and then implemented the inference for sampling. A recent work [7] (LL4AL) designed a deep network with an auxiliary loss prediction for labeled data, then selected samples based on the predicted loss. As the supervised information for network training, both the size and quality of labeled instances determines the performance of models. Further, the above approaches usually require a certain amount of labeled samples to achieve a high accuracy. However, in the early iterations of sampling, the labeled pool is usually small, so that it restricts the ability to choose samples with high quality.

Beside the above annotation information, some recent works focused on utilizing the state information of samples which indicate a sample is labeled (0) or unlabeled (1). As the key of active learning is to select unlabeled samples and label them, this state information can be directly used as the supervised information in this process. Some recent works [8, 39] regarded the state information as a kind of adversarial label. They built a discriminator to map the data point to a binary label which is 1 if the sample is unlabeled and is 0, otherwise. The above works consider all the unlabeled samples of the same quality. In fact, different samples in unlabeled pool have different importance for target task, and an unlabeled sample has lower priority to be labeled if it is more similar to samples in labeled pool. Thus, this state information should be deeply explored to leverage the sample selection.

In this paper, we propose a state relabeling adversarial active learning model (SRAAL) that considers both the annotation and the state information for deriving most informative unlabeled samples. Our model consists of a unified representation generator and a labeled/unlabeled state discriminator. For the generator, we first build an unsupervised image reconstructor based on VAE architecture to learn the rich representation. Secondly, we design a supervised target learner to predict annotations for labeled samples, where the annotation information is embedded into the representation. Then we concatenate the above representations. For the state discriminator, both labeled state and relabeled unlabeled state are used to optimize the discriminator. We propose an online uncertainty indicator to do the state relabeling for unlabeled data, which endues the unlabeled samples with different importance. Specifically, the indicator calculates the uncertainty score for each unlabeled sample as its new state label. State relabeling helps the discriminator to select more instructive samples.

It is notable that most previous works mainly concentrated on the selection strategy in later iterations while the initialization is usually random. However, the initialization of labeled pool has a large influence for subsequent sample selection and performance of active learning. Here we introduce the k-center [15] approach to initialize the labeled pool, where selected samples is a diverse cover for the dataset under the minimax distance.

Experiments on four datasets at image classification and segmentation tasks show that our method outperforms previous state-of-the-art methods. Further, we implement the ablation study to evaluate the contribution of online uncertainty indicator, supervised target learner and our initial sampling algorithm in SRAAL.

The main contributions of this paper are summarized as:

(i) This paper proposes a state relabeling adversarial active learning model to select most informative unlabeled samples. An online uncertainty indicator is designed to relabel the state of unlabeled data with different importance.

(ii) We build an unsupervised image recontructor and a supervised target learner to generate a unified representation of image, where the annotation information is embedded iteratively.

(iii) We propose the initially sampling algorithm based on the k-center approach, which makes subsequent sampling more efficient.

2. Related work

Active learning has been widely studied for decades and most of the classical methods can be grouped into three scenarios: membership query synthesis [1], stream-based selective sampling [5, 23] and pool-based sampling. As the acquirement of abundant unlabeled samples becomes easy, most of recent works [14, 37, 2, 7, 39] focus on the last scenarios. Current active learning methods can be divided into two categories: pool-based approaches and synthesizing approaches.

Instead of querying most informative instances from an unlabeled pool, the synthesizing approaches [31, 32, 47] use generative models to produce new synthetic samples that are informative for the current model. These methods typically introduce various GAN models [16, 33] or VAE models [22, 40] into their algorithm to generate informative data with high quality. However, the synthesizing approaches still has some disadvantages to overcome, such as high computational complexity and instability of performance [47]. For this reason, this paper mainly focuses on research of the pool-based approaches.

The pool-based approaches can be categorized as distribution-based and uncertainty-based methods. The distribution approach chooses data points that will increase the diversity of labeled pool. To do so, the model should extract the representation of the data and calculate the distribution based on it. Previous works [26, 41, 43] have provided various method to learning the representation. Some active learning models [11, 44] optimize the data selection in a discrete space, and [34] clusters the informative data

points to be selected. Some works [3, 18, 29] focus on how to map the distance of distributions to the informativeness of a data point. Besides, some works estimate the distribution diversity by observing gradient [38], future errors [36] or output changes of trained model [13, 20]. Sener et al. [37] introduce core-set technique into active learning. This method calculates the core-set distance by intermediate features rather than the task-specific outputs, which makes the method applicable to any task and network. The core-set technique has a good performance on datasets with small number of classes. However, the core-set method performs ineffective when the number of classes is big or the data points are in high-dimensions [39].

Uncertainty-based approaches do selection by estimating the uncertainties of samples and sampling top-K data points at each iteration. For Bayesian frameworks, [21, 36] estimate uncertainty by Gaussian processes and [9] adopts Bayesian neural networks. [45] propose a novel active learning approach based on the optimum experimental design criteria in statistics. These traditional methods perform well in some specific tasks but do not scale to deep learning network and large-scale datasets. The ensemble model method was proposed by [2] and applied to some specific tasks [42]. [14] introduces Monte Carlo Dropout to build multiple forward passes, which is a general method for various tasks. However, both the ensemble method and dropout method are computationally inefficient for current deep network and large-scale datasets. Yoo et al. [7] propose a Learning-Loss method and has shown the state-of-the-art performance. Their model consists of a task module and a loss prediction module that predicts the loss of the task module. The two modules learn together and the target loss of task module is regarded as a ground-truth loss for the loss prediction module. This method only utilizes the annotation information in labeled samples and the loss prediction accuracy is affected by the performance of task module. If the task module is inaccurate, the predicted loss cannot reflect how informative the sample is.

Some recent works combine uncertainty and distribution to select data points using a two-step process. Distribution of data points can represent the labeled or unlabeled pool and uncertainty estimation based on the distribution can be more generalized and accurate. A two-step model calculating uncertainty based on information density was proposed in [27]. DFAL [8] and VAAL [39] introduces adversarial learning into their models and build a module to learn the representation of data points. The former method extract representation by learning labeled sample's annotation, while the VAAL method builds a latent space by a VAE that learns together with the discriminator. Both of these models map the representation to labeled/unlabeled in a brute force way and the labeled/unlabeled information are not equivalent to informativeness. For this reason, the results of this method may be unreliable.

3. Method

3.1. Overview

In this section, we formally define the scenario of active learning (AL) and set up the notations for the rest of the paper. In the AL, we have a target task and a target model Θ for the task. At the initial stage, there exists a large unlabeled data pool from which we can randomly select $\mathcal{M}$ samples and obtain annotations of them via an oracle. Let us denote the initial unlabeled pool by D_U and the initial labeled pool by D_L. (x_U) denotes that a data point in unlabeled pool and (x_L, y_L) denotes a data point and its annotation in labeled pool.

The key of the AL algorithm is to select the most informative samples from the unlabeled pool D_U. Once the labeled and unlabeled data pools are initialized, a fixed number of samples will be iteratively selected, labeled and transferred from the unlabeled pool to the labeled pool. Then the unlabeled and labeled pool are updated. As illustrated in Fig. 1, this procedure will repeat until the performance of the target model meets user's requirements, or the budget for annotation runs out.

Fig. 2 shows our state relabeling adversarial active learning model (SRAAL), which uses the annotation and labeled/unlabeled state information for selecting the most informative samples. The SRAAL consists of a unified representation generator (Section 3.2) and a labeled/unlabeled state discriminator (Section 3.3). The former learns the annotation-embedded image feature, and the latter selects more representative samples to be labeled with the help of the online uncertainty indicator. Sampling strategy based on the generator and discriminator is introduced in Section 3.4, and the proposed initially sampling algorithm with k-center is detailed in Section 3.5.

3.2. Unified representation generator

The image representation learning is in the charge of the unified representation generator which consists of the unsupervised image reconstructor (UIR) and the supervised target learner (STL). The image encoder consists of a CNN and two FC modules. The CNN extracts image feature and then FC individually learns the two latent variables for STL and UIR. The UIR module is a variational autoencoder (VAE) in which a low dimensional latent space is learned based on a Gaussian prior. As this process does not require annotations and the reconstruction target is the image itself, samples from both the labeled pool and unlabeled pool contribute to this module. As it's a VAE, the objective function

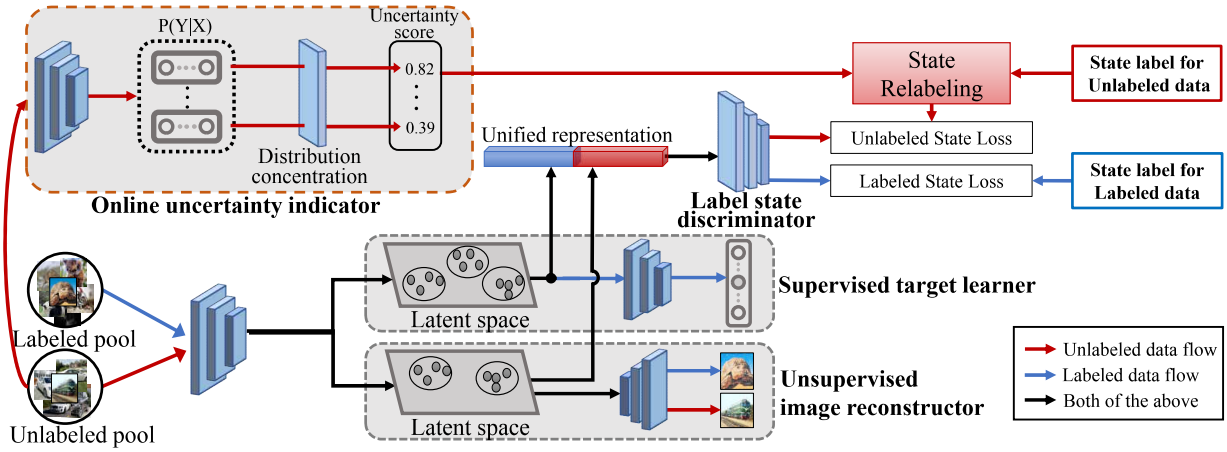


Figure 2. Network architecture of our proposed SRAAL. It consists of a unified representation generator and a labeled/unlabeled state discriminator. The generator embeds the annotation information into the final image features via the supervised target learner and unsupervised image reconstructor. Online uncertainty indicator is introduced to relabel the state of unlabeled samples and endues them with different importance. Finally, the state discriminator is updated through the labeled and unlabeled state losses, and helps select the more informative samples.

of this module can be formulated as,

$$\begin{aligned}\mathcal{L}^{UIR} &= \mathcal{L}_U^{UIR} + \mathcal{L}_L^{UIR} \\ \mathcal{L}_U^{UIR} &= E[log[p_\phi(x_U \mid z_U)] - D_{KL}(q_\theta(z_U \mid x_U) \parallel p(z))] \\ \mathcal{L}_L^{UIR} &= E[log[p_\phi(x_L \mid z_L)] - D_{KL}(q_\theta(z_L \mid x_L) \parallel p(z))]\end{aligned} \tag{1}$$

where $\mathcal{L}_U^{UIR}$ is the objective function for unlabeled data points and L_L^{UIR} is for labeled ones, z is the latent variables, ϕ parametrizes the decoder p_ϕ and θ parametrizes the encoder q_θ. The UIR module finally learns the rich representation by reconstructing both the labeled and unlabeled samples.

To embed the annotation information into the representation, we build a supervised target learner to predict annotations of the samples based on the representation in latent space. The STL is also a VAE network and its decoder does not decode the representation for reconstruction. The decoder of STL varies with different tasks. For example, the decoder is a classifier for image classification, or a segmentation model for semantic segmentation. The STL is similar to VAE but only labeled samples can provide loss for STL's training. We can formulate the objective function for STL as follows,

$$\mathcal{L}_L^{STL} = E[log[p_\phi(y_L \mid z_L)]] - D_{KL}(q_\theta(z_L \mid x_L) \parallel p(z))] \tag{2}$$

where z_L is the latent variables from the latent space for labeled data, ϕ parametrizes the decoder p_ϕ and θ parametrizes the encoder q_θ. The STL module will embed the annotation information into the representation.

The two above representations are concatenated together as the unified image representation.

3.3. State discriminator and state relabeling

To make full use of the state information, we introduce the adversarial learning into SRAAL, where a discriminator is built to model the state of samples. The previous works utilize the binary state information, where the state of unlabeled samples is set to 1 and that of labeled samples is set to 0. In fact, different samples in unlabeled pool have different contribution for target task, and an unlabeled sample has lower priority to be labeled if it is more similar to samples in labeled pool. To better use the state information, we propose the online uncertainty indicator (OUI) to calculate an uncertainty score to relabel the state of unlabeled data. The uncertainty score measures the distribution concentration of the unlabeled data and is bound to [0,1]. After the state relabeling, the state of unlabeled samples changes from the fixed binary label 1 to a new continuous state.

The OUI calculates the uncertainty score based on the prediction vector of the target model(such as image classifier, semantic segmentation model). Before each iteration, the target model is trained with labeled data and then produce a prediction vector for each unlabeled sample. Specifically, for image classification, the prediction is a probability vector for each category. For segmentation, each pixel has a probability vector and the prediction vector is the mean of each probability vector. Assume that the number of classes is C and the samples are labeled with $y_i \in \mathcal{R}^C$. The calcu-

lation of the uncertainty score is formulated as,

$$Indicator(x_U) = 1 - \frac{MINVar(V)}{Var(V)} \times max(V) \quad (3)$$

where x_U is an unlabeled sample, $V = p(x_U|D_L)$ is the probability vector of x_U based on the target model trained with current labeled pool D_L.

The $MINVar(V)$ can be formulated as,

$$\begin{aligned} &MINVar(V) = Var(V') \\ &= \frac{1}{C}((\frac{1}{C} - max(V))^2 + (C-1)(\frac{1}{C} - \frac{1-max(V)}{1-C})^2) \end{aligned} \quad (4)$$

$MINVar(V)$ is the variance of the vector V', whose maximum element is the same with the V's and other elements have the same value $\frac{1-max(V)}{C-1}$. $MINVar(V)$ is the smallest variance among vectors whose maximum are same with V's, so that $\frac{MINVar(V)}{Var(V)}$ measures the concentration of the probabilities distribution. According to Eq. 3, we can prove that the uncertainty score has three properties: (1) it has a boundary of [0,1); (2) it has a negative correlation with the value of maximum probability; (3) it has a positive correlation with the concentration of the probabilities distribution. Due to these properties, the uncertainty score has a good response to the informativeness of samples.

The objective function of the discriminator is defined as follows,

$$\begin{aligned} \mathcal{L}^D = &- E[log(D(q_\theta(z_L \mid x_L)))] \\ &- E[log(Indicator(x_U) - D(q_\theta(z_U \mid x_U)))] \end{aligned} \quad (5)$$

where the indicator relabels the unlabeled data's label.

As adversarial learning, the objective function of the unified representation generator in SRAAL is

$$\begin{aligned} \mathcal{L}^G_{adv} = &- E[log(D(q_\theta(z_L \mid x_L)))] \\ &- E[log(D(q_\theta(z_U \mid x_U)))] \end{aligned} \quad (6)$$

The total objective function combined with Eq. 1, Eq. 2 and Eq. 6 for the latent variable generator is also given as follows,

$$\mathcal{L}^G = \lambda_1 \mathcal{L}^{UIR} + \lambda_2 \mathcal{L}^{STL}_L + \lambda_3 \mathcal{L}^G_{adv} \quad (7)$$

3.4. Sampling strategy in active learning

The algorithm for training the SRAAL at each iteration is shown in Fig. 2. In the sampling step, the generator generates the unified representation for each unlabeled sample. The discriminator predicts its state value, and the top-K samples are selected to be labeled by the oracle.

Algorithm 1 Initialization of labeled pool

Input: labeled data pool D_L, unlabeled pool D_U, the size of initial labeled pool $\mathcal{M}$, latent variables z for all the data points

Hyperparameters: Randomly select I ($I \ll \mathcal{M}$) data points in D_U and move them to D_L

repeat

$u = argmax_{x_U \in D_U}[min_{x_L \in D_L} Distance(x_U, x_L)]$

$D_L = D_L \cup \{u\}$

$D_U = D_U - \{u\}$

until $size(D_L) = \mathcal{M}$

return the initialized labeled pool D_L

3.5. Initially sampling algorithm

It is worth noting that most AL methods mainly study the selection strategy, while the initialization of labeled pool is usually random. However, the initialization of labeled pool can heavily affect subsequent sample selection and performance of active learning. Thus, we propose an initially sampling algorithm in which the problem of initially sampling is defined as a set cover problem. The goal cover problem is to find a subset of data points that the largest distance of any point to the subset is minimum. To measure the distance between samples, first we train the unsupervised image reconstructor so that it learns the latent variables of all the samples, then we apply a greedy k-center algorithm where the distance between two samples is measured by the Euclidean distance between their latent variables. The final output is a subset with $\mathcal{M}$ samples that is labeled by an oracle and sent to the labeled pool. Alg. 1 shows the detail of the algorithm.

4. Experiment

In this section, we evaluate SRAAL against state-of-the-art active learning approaches on image classification and segmentation task.

For both tasks, we initialize the labeled pool D^0_L by randomly sampling M = 10% samples from the entire dataset and the rest 90% samples make up the initial unlabeled pool D^0_U. The unlabeled pool contains the rest of the training set form which samples are selected to be annotated by the oracle. We iteratively train the current model and select K = 5% samples from the unlabeled pool until the portion of labeled samples reaches 40%. For each active learning method, we repeat the experiment 5 times with different initial labeled pool and report the mean performance. When we compare the performance with our methods, they both start with the same initial labeled pool.

To verify the performance of our initial sampling algorithm, we also set an experiment to compare the designed

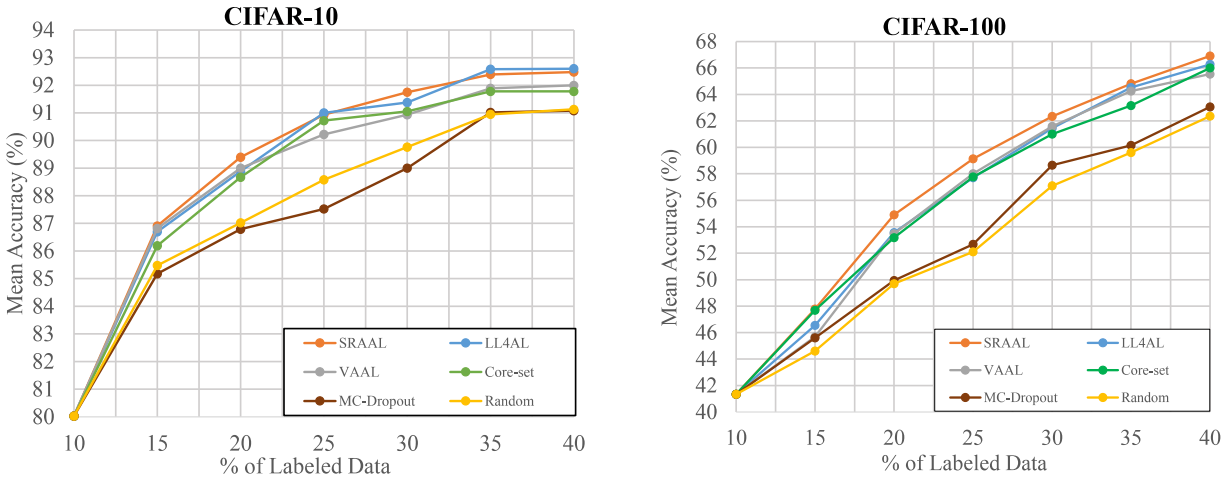


Figure 3. Active learning results of image classification over CIFAR-10 and CIFAR-100.

initialization with the random style. Besides these experiments, we also set an experiment for ablation study to evaluate some main modules in our model.

4.1. Active learning for image classification

Dataset. For the classification task, We choose CIFAR-10, CIFAR-100 [24] and Caltech-101 [12] as they are classical for image recognition and some recent works evaluate on them. Both of CIFAR-10 and CIFAR-100 have 60,000 images of 32×32×3 where 50000 is training images and 10000 is test images. The CIFAR-10 with 10 categories has 6000 images per class, while the CIFAR-100 has 100 classes containing 600 images each. The Caltech-101 consists of a total of 9,146 images, split between 101 different categories, as well as a background category, and each category in Caltech-101 has about 40 to 800 images. These datasets simulate different real-world situations for the number of images per class.

Compared methods. For image classification,we compare the performance of SRAAL with some recent state-of-the-art approaches, including Core-set [37], Monte-Carlo Dropout [14], VAAL [39] and LL4AL. We also introduce the random selection method as a baseline. We reproduce the results of these works with the official released codes and adopt the original hyperparameters. When evaluating, these methods are evaluated by the same target model.

Performance measurement. We evaluate the performances of these methods in image classification by measuring the average accuracy of 5 trials. In each trial, all the methods begin with a same initial labeled pool. The target model used to evaluate the accuracy is a 18-layer residual network (ResNet-18). We utilize a specified Resnet-18 model for CIFAR-10/100 and a classical Resnet-18 for Caltech-101. Besides, images from Caltech-101 are resized to 220×220 for convenience.

4.1.1 Performance on CIFAR-10

The left of Fig. 3 shows the performances on CIFAR-10. We can observe that, first, our method achieves an accuracy over 90% by using 25% of the data and the performance in last iteration reaches 92.48%. The highest accuracy of the Resnet-18 with full dataset reaches 93.5%, which is only 1.02% better than SRAAL with 40% samples. This shows that on CIFAR-10 SRAAL performs closely to the full-data trained model. Second, our method evidently outperforms MC-Dropout, Random sample, core-set and VAAL and is on par with the LL4AL. LL4AL outperforms our SRAAL at 25%, 35% and 40% with very slight lead, but underperforms our method at 15%, 20% and 30%. This also demonstrates that our SRAAL has selected more informative samples, and it benefits from the use of annotation and labeled/unlabeled state information.

4.1.2 Performance on CIFAR-100

CIFAR-10 dataset has 50000 training images categorized into 10 classes, while the CIFAR-100 has 50000 images in 100 classes. Thus, this dataset is much more challenging. The right of Fig. 3 shows the performances, we find that, first, all the AL methods have better results than random selection method. Second, on CIFAR-10, LL4AL performs continuously better than most methods. However, the LL4AL on CIFAR-10 becomes not as competitive as on CIFAR-10. Especially in first iteration, the core-set and LL4AL achieve better performance than LL4AL. The LL4AL trains its model only with the labeled samples. The inadequate labeled samples restrain the accuracy of its main module, which makes the sample selection inefficient. Third, for all iterations, our SRAAL achieves the better performance than the state-of-the-art methods, such as VAAL, LL4AL. Although the VAAL also uses the label state in-

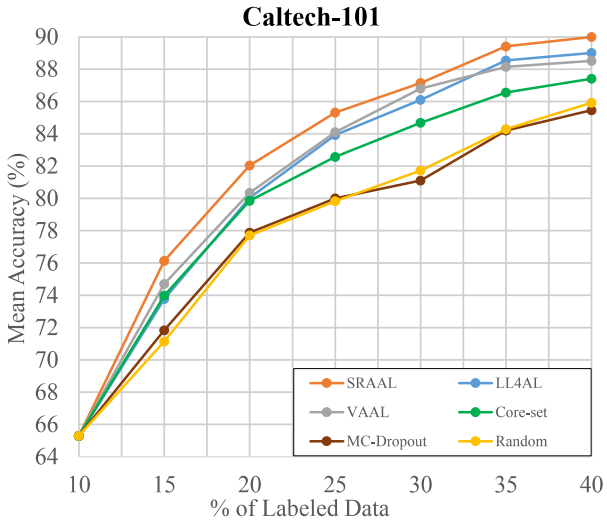


Figure 4. Active learning results of image classification over Caltech-101.

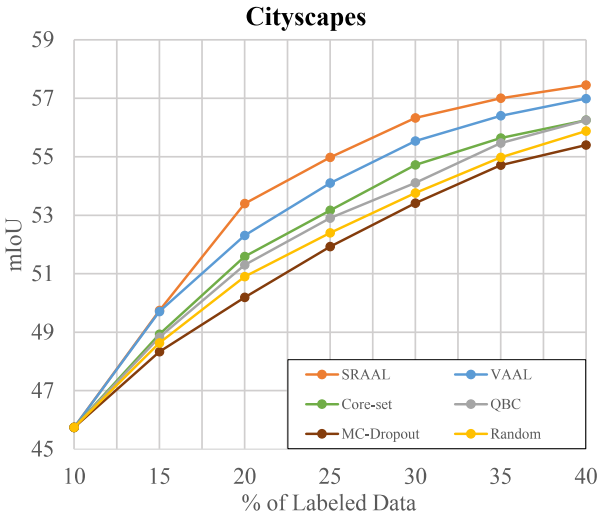


Figure 5. Active learning results of image semantic segmentation over Cityscapes.

formation, the state relabeling in the discriminator from our SRAAL has a better use of it. Besides, the annotation embedded unified representation from the generator provides the richer feature for samples.

4.1.3 Performance on Caltech-101

To further explore the influence of image amount per class to AL model, we conduct the comparison experiment on the Caltech-101 dataset. Fig. 4 shows the performance of these methods. Caltech-101 has less images per class and the image amount for each class is also different. We find that SRAAL outperforms all previous methods from the first iteration to last, and the gap between SRAAL and second-best method becomes larger than that over CIFAR100. This phenomenon provides a further proof that our method can better resist the impact from adequate labeled samples. Besides, the core-set method and LL4AL method perform worse than VAAL because they only utilize annotation information. This verifies that the label state information is useful to help sample the representative data again.

4.2. Active learning for semantic segmentation

Dataset. For semantic segmentation, it is a popular task to evaluate the active learning model. Semantic segmentation is more challenging than image classification, so that this experiment evaluates the performance of AL methods on a difficult task. Here we also conduct the comparison experiment on the Cityscapes dataset [4]. This dataset has 3475 frames with instance segmentation annotations recorded in street scenes. Following [39], we convert this dataset into 19 classes.

Compared methods. We evaluate our SRAAL against some active learning approaches for semantic segmentation. The compared works include Core-set [37], MC-Dropout [14], Query-By-Committee (QBC) [25], suggestive annotation (SA) [42] and VAAL [39].

Performance measurement. For this task, the target model is DRN, and the mean IoU is used to evaluate the performances. All the methods are evaluated with a same initial labeled pool and a same selection budget for each iteration.

Fig. 5 shows our results of semantic segmentation about different methods. We can observe that, first, the VAAL and our SRAAL obtain better performance than other methods, such as SA, QBC, MC-Dropout. This is because both VAAL and SRAAL introduce the label state information to model the sample selection. Second, our SRAAL outperforms the VAAL with a large margin. This benefits from the state relabeling of the proposed online uncertainty indicator. This relabeled state can better guide the discriminator to choose the most informative samples.

4.3. Initialization algorithm comparison

As mentioned in Section 3.5, we introduce the k-center approach to initialize the labeled pool. We evaluate the initialization algorithm on the CIFAR-10 dataset. The AL model is our proposed SRAAL and the target model is the ResNet-18 image classifier, which is the same with that in Section 4.1.

As shown in Fig. 6, we can observe that the mean accuracy of our initialization algorithm is significantly higher than the random initialization. The higher accuracy proves that our initial labeled samples are more informative than random selected ones. Further, the dotted lines show that the standard deviation of our initialization is also less than that of the random initialization. To sum up, our initializa-

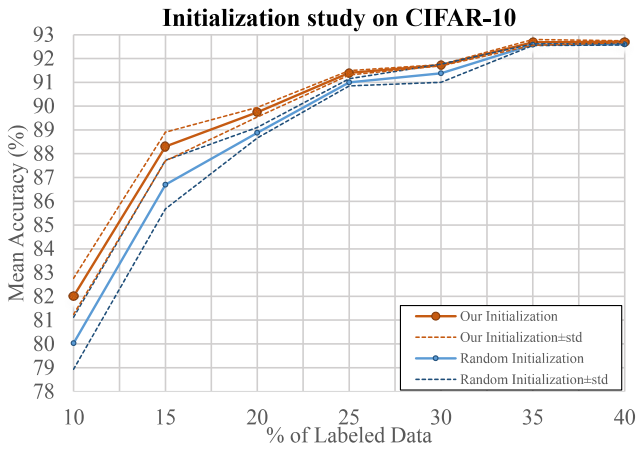


Figure 6. Experiment result for initial sampling algorithm.

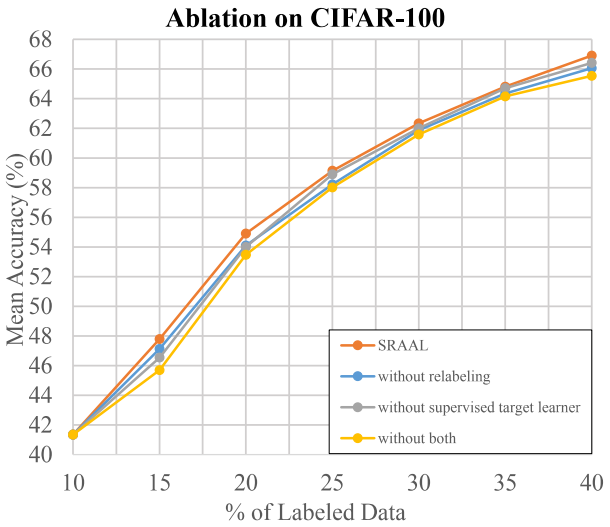


Figure 7. Experiment result for ablation study on CIFAR-100.

tion makes subsequent sampling more efficient.

4.4. Ablation study

To evaluate the contribution of different modules in our model, we conduct this experiment for ablation study on CIFAR-100 dataset. As the state relabeling is the key of our work, we first verify the role of it by eliminating the online uncertainty indicator and using the original binary label state. We also perform an ablation on the supervised target learner to explore the importance of the annotation embedded unified representation. Besides, an ablation to both two modules is performed as the control group.

Fig. 7 shows the results for the ablation study. The complete SRAAL consistently outperforms all the ablations, and the ablation to two modules performs yields lowest accuracy among the ablations. The above phenomenon illustrates that either the relabeling or the annotation-embedded representation helps to improve the AL performance.

4.5. Comparison on different uncertainty estimators

To accurately relabel the state of unlabeled data with different importance, we design an uncertainty score(Eq. 3) in the online uncertainty indicator module. To verify the superiority of our score and prove that our uncertainty score is more suitable for state relabeling, we compare some common uncertainty acquisition functions with ours by replacing our uncertainty score with them. The experiment result in Tab. 1 shows that our indicator outperforms these uncertainty acquisition functions under different sampling ratios, which verifies that our uncertainty score can better reflect the importance of unlabeled data.

(%)	20%	25%	30%	35%	40%
Ours	55.0	59.1	62.3	65.0	66.9
Entropy	54.0	58.2	61.1	64.5	65.7
SD	54.1	57.0	59.4	63.3	64.1

Table 1. Comparison with entropy and standard deviation(SD) under different sampling ratios.

5. Conclusion

In this paper, we study the active learning and propose a state-relabeling adversarial active learning model (SRAAL) that makes full use of both the annotation and the state information for deriving most informative unlabeled samples. The model consists of a unified representation generator that learns the annotation-embedded image feature, and a labeled/unlabeled state discriminator that selects most informative samples with the help of online updated indicator. Further, we introduce the k-center approach to initialize the labeled pool, which makes subsequent sampling more efficient. The experiments on image classification and segmentation demonstrate that our model outperforms previous state-of-the-art methods and the initially sampling algorithm significantly improve the performance of our model.

Acknowledgement. This work was supported in part by the National Key R&D Program of China under Grand:2018AAA0102003, in part by National Natural Science Foundation of China: 61771457, 61732007, 61772497, 61772494, 61931008, 61620106009, U1636214, 61622211, U19B2038, and in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013 and the Fundamental Research Funds for the Central Universities under Grant WK2100100030.

References

[1] Dana Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

[2] William H Beluch, Tim Genewein, Andreas Nürnberger, and Jan M Köhler. The power of ensembles for active learning in image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[3] Mustafa Bilgic and Lise Getoor. Link-based active learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Ido Dagan and Sean P Engelson. Committee-based sampling for training probabilistic classifiers. In *Machine Learning Proceedings 1995*, pages 150–157. Elsevier, 1995.

[6] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[7] In So Kweon Donggeun Yoo. Learning loss for active learning. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2019.

[8] Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

[9] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. *arXiv preprint arXiv:1906.02425*, 2019.

[10] Sayna Ebrahimi, Anna Rohrbach, and Trevor Darrell. Gradient-free policy architecture search and adaptation. *arXiv preprint arXiv:1710.05958*, 2017.

[11] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Shankar Sasrty. A convex optimization framework for active learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 209–216, 2013.

[12] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Pattern Recognition Workshop*, 2004.

[13] Alexander Freytag, Erik Rodner, and Joachim Denzler. Selecting influential examples: Active learning with expected model output changes. In *European Conference on Computer Vision*, pages 562–577. Springer, 2014.

[14] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

[15] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(none):293–306.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[17] Marc Gorriz, Axel Carlier, Emmanuel Faure, and Xavier Giro-i Nieto. Cost-effective active learning for melanoma segmentation. *arXiv preprint arXiv:1711.09168*, 2017.

[18] Mahmudul Hasan and Amit K Roy-Chowdhury. Context aware active learning of activity recognition models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4543–4551, 2015.

[19] Armand Joulin, Laurens van der Maaten, Allan Jabri, and Nicolas Vasilache. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*, pages 67–84. Springer, 2016.

[20] Christoph Käding, Erik Rodner, Alexander Freytag, and Joachim Denzler. Active and continuous exploration with deep neural networks and expected model output changes. *arXiv preprint arXiv:1612.06129*, 2016.

[21] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.

[22] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[23] Vikram Krishnamurthy. Algorithms for optimal scheduling and management of hidden markov model sensors. *IEEE Transactions on Signal Processing*, 50(6):1382–1397, 2002.

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[25] Weicheng Kuo, Christian Häne, Esther Yuh, Pratik Mukherjee, and Jitendra Malik. Cost-sensitive active learning for intracranial hemorrhage detection.

[26] Liang Li, Shuqiang Jiang, and Qingming Huang. Learning hierarchical semantic description via mixed-norm regularization for image understanding. *IEEE Transactions on Multimedia*, 14(5):1401–1413, 2012.

[27] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–866, 2013.

[28] Xuejing Liu, Liang Li, Shuhui Wang, Zheng-Jun Zha, Dechao Meng, and Qingming Huang. Adaptive reconstruction network for weakly supervised referring expression grounding. In *IEEE/CVF International Conference on Computer Vision, ICCV*, pages 2611–2620. IEEE, 2019.

[29] Oisin Mac Aodha, Neill DF Campbell, Jan Kautz, and Gabriel J Brostow. Hierarchical subquery evaluation for active learning on a graph. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 564–571, 2014.

[30] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018.

[31] Dwarikanath Mahapatra, Behzad Bozorgtabar, Jean-Philippe Thiran, and Mauricio Reyes. Efficient active learning for

image classification and segmentation using a sample selection and conditional generative adversarial network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 580–588. Springer, 2018.

[32] Christoph Mayer and Radu Timofte. Adversarial sampling for active learning. *arXiv preprint arXiv:1808.06671*, 2018.

[33] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[34] Hieu T Nguyen and Arnold Smeulders. Active learning using pre-clustering. In *ICML*, page 79. ACM, 2004.

[35] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5898–5906, 2017.

[36] Nicholas Roy and Andrew McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

[37] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[38] Burr Settles, Mark Craven, and Soumya Ray. Multiple-instance active learning. In *Advances in neural information processing systems*, pages 1289–1296, 2008.

[39] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. *arXiv preprint arXiv:1904.00370*, 2019.

[40] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[41] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 5693–5703, 2019.

[42] Lin Yang, Yizhe Zhang, Jianxu Chen, Siyuan Zhang, and Danny Z Chen. Suggestive annotation: A deep active learning framework for biomedical image segmentation. In *International conference on medical image computing and computer-assisted intervention*, pages 399–407. Springer, 2017.

[43] Shijie Yang, Liang Li, Shuhui Wang, Weigang Zhang, Qingming Huang, and Qi Tian. Skeletonnet: A hybrid network with a skeleton-embedding process for multi-view image representation learning. *IEEE Transactions on Multimedia*, 21(11):2916–2929, 2019.

[44] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.

[45] Zheng-Jun Zha, Meng Wang, Yan-Tao Zheng, Yi Yang, Richang Hong, and Tat-Seng Chua. Interactive video indexing with statistical active learning. *IEEE Trans. Multimedia*, 14(1):17–27, 2012.

[46] Zhi Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, v.5(1):48–57.

[47] Jia-Jie Zhu and José Bento. Generative adversarial active learning. *arXiv preprint arXiv:1702.07956*, 2017.

A DEEP ACTIVE LEARNING SYSTEM FOR SPECIES IDENTIFICATION AND COUNTING IN CAMERA TRAP IMAGES

A PREPRINT

Mohammad Sadegh Norouzzadeh[1,2], **Dan Morris**[1], **Sara Beery**[1,4], **Neel Joshi**[3], **Nebojsa Jojic**[3], and **Jeff Clune**[2,5]

[1]Microsoft AI for Earth, Redmond, WA
[2]Computer Science department, University of Wyoming, Laramie, WY
[3]Microsoft Research, Redmond, WA
[4]Computer Science Department, California Institute of Technology, Pasadena, CA
[5]Uber AI, San Francisco, CA

October 23, 2019

ABSTRACT

Biodiversity conservation depends on accurate, up-to-date information about wildlife population distributions. Motion-activated cameras, also known as camera traps, are a critical tool for population surveys, as they are cheap and non-intrusive. However, extracting useful information from camera trap images is a cumbersome process: a typical camera trap survey may produce millions of images that require slow, expensive manual review. Consequently, critical information is often lost due to resource limitations, and critical conservation questions may be answered too slowly to support decision-making. Computer vision is poised to dramatically increase efficiency in image-based biodiversity surveys, and recent studies have successfully harnessed deep learning techniques for automatic information extraction from camera trap images. However, the accuracy of results depends on the amount, quality, and diversity of the data available to train models, and the literature has focused on projects with millions of relevant, labeled training images. Many camera trap projects do not have a large set of labeled images, and hence cannot benefit from existing machine learning techniques. Furthermore, even projects that do have labeled data from similar ecosystems have struggled to adopt deep learning methods because image classification models overfit to specific image backgrounds (i.e., camera locations). In this paper, we focus not on *automating* the labeling of camera trap images, but on *accelerating* this process. We combine the power of machine intelligence and human intelligence to build a scalable, fast, and accurate active learning system to minimize the manual work required to identify and count animals in camera trap images. Our proposed scheme can match the state of the art accuracy on a 3.2 million image dataset with as few as 14,100 manual labels, which means decreasing manual labeling effort by over 99.5%.

***K*eywords** deep learning, deep neural networks, camera trap images, active learning, computer vision

1 Introduction

Wildlife population studies depend on tracking observations, i.e. occurrences of animals at recorded times and locations. This information facilitates the modeling of population sizes, distributions, and environmental interactions [1, 2, 3]. Motion-activated cameras, or camera traps, provide a non-intrusive and comparatively cheap method to collect observational data, and have transformed wildlife ecology and conservation in recent decades [4, 5]. Although camera trap networks can collect large volumes of images, turning raw images into actionable information is done manually, i.e. human annotators view and label each image [6]. The burden of manual review is the main disadvantage of camera trap surveys and limits the use of camera traps for large-scale studies.

Fortunately, recent advances in artificial intelligence have significantly accelerated information extraction. Loosely inspired by animal brains, deep neural networks [7, 8] have advanced the state of the art in tasks such as machine translation [9, 10], speech recognition [11, 12], and image classification [13, 14]. Deep convolutional neural networks are a class of deep neural networks designed specifically to process images [8, 15].

Recent work has demonstrated that deep convolutional neural networks can achieve a high level of accuracy in extracting information from camera trap images—including species labels, count, and behavior—while being able to process hundreds of images in a matter of seconds [16, 17]. The wide availability of deep learning for fast, automatic, accurate, and inexpensive extraction of such information could save substantial amounts of time and money for conservation biologists.

The accuracy of deep neural networks depends on the abundance of their training data [8]; state-of-the-art networks typically require millions of labeled training images. This volume of labeled data is not typically available for camera trap projects; therefore, most projects cannot yet effectively harness deep learning. Even in cases where an extensive training set is available, training labels are almost always in the form of image-level or sequence-level species labels, i.e. they do not contain information about where animals occur within each image. This results in a strong dependency of deep networks on image backgrounds [18, 19], which limits the ability of deep learning models to produce accurate results even when applied to regions with species distributions that are similar to their training data, but with different backdrops due to different camera trap locations.

This paper aims to address these issues and to enable camera trap projects with few labeled images to take advantage of deep neural networks for fast, transferable, automatic information extraction. Using object detection models, transfer learning, and active learning, our results show that our suggested method can achieve the same level of accuracy as a recent study by Norouzzadeh et al. [16] that harnessed 3.2 million labeled training examples to produce 90.9% accuracy (using ResNet-50 architecture) at species classification, but with a 99.5% reduction in manually-annotated training data. We also expect our method to generalize better to new locations because we systematically filter out the background pixels.

2 Background and related work

2.1 Deep learning

The most common type of machine learning used for image classification is *supervised learning*, where input examples are provided along with corresponding output examples (for example, camera trap images with species labels), and algorithms are trained to translate inputs to the appropriate outputs [20].

Deep learning is a specific type of supervised learning, built around *artificial neural networks* [21, 8], a class of machine learning models inspired by the structure of biological nervous systems. Each artificial neuron in a network takes in several inputs, computes a weighted sum of those inputs, passes the result through a non-linearity (e.g. a sigmoid), and transmits the result along as input to other neurons. Neurons are usually arranged in several layers; neurons of each layer receive input from the previous layer, process them, and pass their output to the next layer. A *deep* neural network is a neural network with three or more layers [8]. Typically, the free parameters of the model that are trained are the *weights* (aka connections) between neurons, which determine the weight of each feature in the weighted sum.

In a *fully-connected layer*, each neuron receives input from all the neurons in the previous layer. On the other hand, in *convolutional layers*, each neuron is only connected to a small group of nearby neurons in the previous layer and the weights are trained to detect a useful pattern in that group of neurons [21, 8]. Additionally, convolutional neural networks inject the prior knowledge that translation invariance is helpful in computer vision (e.g. an eye in one location in an image remains an eye even if it appears somewhere else in the image). This is enforced by having a feature detector reused at many points throughout the image (known as *weight tying* or *weight sharing*. A neural network with one or more convolutional layers is called a *convolutional neural network*, or CNN. CNNs have shown excellent performance on image-related problems [7, 8].

The weights of a neural network (aka its parameters) determine how it translates its inputs into outputs; *training* a neural network means adjusting these parameters for every neuron so that the whole network produces the desired output for each input example. To tune these parameters, a measure of the discrepancy between the current output of the network and the desired output is computed; this measure of discrepancy is called the *loss function*. There are numerous loss functions used in the literature that are appropriate for different problem classes. After calculating the loss function, an algorithm called Stochastic Gradient Descent (SGD) [22, 23] (or modern enhancements of it [24, 25]) calculates the contribution of each parameter to the loss value, then adjusts the parameters so that the loss value is minimized. The backpropagation algorithm is an iterative algorithm, i.e. it is applied many times during

training, including multiple times for each image in the dataset. At every iteration of the backpropagation algorithm, the parameters take one step toward a minimum (i.e. the best solution in a local area of the search space of all possible weights: note that the term minima is used instead of maxima because we are minimizing the loss, or the error).

The accuracy of deep learning compared to other machine learning methods makes it applicable to a variety of complex problems. In this paper, we focus on enhancing deep neural networks to extract information from camera trap images more efficiently.

2.2 Image classification

In the computer vision literature, *image classification* refers to assigning images into several pre-determined classes. More specifically, image classification algorithms typically assign a probability that an image belongs to each class. For example, species identification in camera trap images is an image classification problem in which the input is the camera trap image and the output is the probability of the presence of each species in the image [16, 17]. Image classification models can be easily trained with image-level labels, but they suffer from several limitations:

1. Typically the most probable species is considered to be the label for the image; consequently, classification models cannot deal with images containing more than one species.
2. Applying them to non-classification problems like counting results in worse performance than classification [16].
3. What the image classification models see during training are the images and their associated labels; they have not been told what *parts* of the images they should focus on. Therefore, they not only learn about patterns representing animals, but will also learn some information about *backgrounds* [18]. This fact limits their transferability to new locations. Therefore, when applied to new datasets, accuracy is typically lower than what was achieved on the training data. For example, Tabak et al. [17] showed that their model trained on images from the United States was less accurate at identifying the same species in a Canadian dataset.

2.3 Object detection

Object detection algorithms attempt to not only classify images, but to locate instances of predefined object classes within images. Object detection models output coordinates of bounding boxes containing objects plus a probability that each box belongs to each class. Object detection models thus naturally handle images with objects from multiple classes. (Fig. 1). A hypothesis of this paper is that object detection models may also be less sensitive to image backgrounds (because the model is told explicitly which regions of each image to focus on), and may thus generalize more effectively to new locations.

The ability of object detection models to handle images with multiple classes makes them appealing for camera trap problems, where multiple species may occur in the same images. However, training object detection models requires bounding box and class labels for each animal in the training images. This information is rarely relevant for ecology, and obtaining bounding box labels is costly; consequently, few camera trap projects have such labels. This makes training object detection models impractical for many camera trap projects, although recent work has demonstrated the effectiveness of object detection when bounding box labels are available [26, 19].

2.4 Transfer learning

Despite not explicitly being trained to do so, deep neural networks trained on image datasets often exhibit an interesting phenomenon: early layers learn to detect simple patterns like edges [15]. Such patterns are not specific to a particular dataset or task, but they are general to different datasets and tasks. Subsequent layers detect more complex and more specific patterns to the dataset the network is trained on. Eventually, there is a transition from general features to dataset-specific features, and from simple to complex patterns within the layers of the network [27].

Transfer learning is the application of knowledge gained from learning a task to a similar, but different, task [27]. Transfer learning is highly beneficial when we have a limited number of labeled samples to learn a new task (for example, species classification in camera trap images when the new project has few labeled images), but we have a large amount of labeled data for learning a different, relevant task (for example, general-purpose image classification). In this case, a network can first be trained on the large dataset and then *fine-tuned* on the target dataset [27, 16]. Using transfer learning, the general features deep neural networks learn on a large dataset can be reused to learn a smaller dataset more efficiently.

Figure 1: Object detection models are capable of detecting multiple occurrences of several object classes.

2.5 Active learning

In contrast to the supervised learning scenario, in which we first collect a large amount of labeled examples and then train a machine learning model, in an *active learning scenario* we have a large pool of unlabeled data and an oracle (e.g. a human) that can label the samples upon request. Active learning iterates between training a machine learning model and asking the oracle for *some* labels, but it tries to minimize the number of such requests. The active learning algorithm must select the samples from the pool for the oracle to label so that the underlying machine learning model can quickly learn the requested task.

Active learning algorithms maintain an underlying machine learning model, such as a neural network, and try to improve that model by selecting training samples. Active learning algorithms typically start training the underlying model on a small, randomly-selected labeled set of data samples. After training the initial model, various criteria can be employed to select the most informative unlabeled samples to be passed to the oracle for labeling [28]. Among the most popular query selection strategies for active learning are model uncertainty [29], query-by-committee (QBC) [30], expected model change [31], expected error reduction [32], and density-based methods [31, 33]. For more information on the criteria we use in this paper, refer to the Supplementary Information (SI) sec. S2. After obtaining the new labels from the oracle and retraining the model, the same active learning procedure can be repeated until a pre-determined number of images have been labeled, or until an acceptable accuracy level is reached. Algorithm 1 summarizes an active learning workflow in pseudocode.

Algorithm 1 Active learning procedure

```
Start from a small, randomly-selected labeled subset of data
while Stopping criteria not met do
    Train the underlying model with the available labeled samples
    Compute a selection criterion for all the samples in the unlabeled pool
    Select n samples that maximize the criterion
    Pass the selected samples to the oracle for labeling
    Gather the labeled samples and add them to the labeled set
end while
```

2.6 Embedding learning

An *embedding function* maps data from a high-dimensional space to a lower-dimensional space, for example from the millions of pixel values in an image (high-dimensional) to a vector of dozens or hundreds of numeric values. Many

dimensionality reduction algorithms such as PCA [34] and LDA [34], or visualization algorithms like t-SNE [35], can be regarded as embedding functions.

Deep neural networks are frequently used for dimensionality reduction: the input to a deep network often has many values, but layers typically get smaller throughout the network, and the output of a layer can be viewed as a reduced representation of the network's input. In this paper, we use two common methods to train a deep neural network to produce useful embeddings:

1. We learn an embedding in the course of training another task (e.g., image classification). Here we follow common practice and train a deep neural network for classification with a *cross-entropy loss* and use the activations of the penultimate layer after training as the embedding. Cross-entropy is the most common loss function used for classification problems [8].
2. We learn an embedding that specifically maps samples from the same class to nearby regions in the learned embedding space [36, 37]. Triplet loss [37] is a popular loss function to accomplish this goal. For more details on triplet loss, refer to SI sec. S1.

We thus have two experimental treatments regarding embedding learning: one with a cross-entropy loss and another with a triplet loss.

2.7 Datasets

Three datasets will be used for training and evaluating models in our experiments: Snapshot Serengeti, eMammal Machine Learning, NACTI, and Caltech Camera Traps.

Snapshot Serengeti

The Snapshot Serengeti dataset contains 1.2 million multi-image sequences of camera trap images, totaling 3.2 million images. Sequence-level species, count, and other labels are provided for 48 animal categories by citizen scientists [6]. Approximately 75% of the images are labeled as empty. Wildebeest, zebra, and Thomson's gazelle are the most common species.

eMammal Machine Learning

eMammal is a data management platform for both researchers and citizen scientists working with camera trap images. We worked with a dataset provided by the eMammal team specifically to support machine learning research, containing over 450,000 images and over 270 species from a diverse set of locations across the world [38, 39].

NACTI

The North America Camera Trap Images (NACTI) dataset [40] contains 3.7 million camera trap images from five locations across the United States, with labels for 28 animal categories, primarily at the species level (for example, the most common labels are cattle, boar, and red deer). Approximately 12% of images are labeled as empty.

Caltech Camera Traps

The Caltech Camera Traps (CCT) dataset [41] contains 245 thousand images from 140 camera traps in the Southwestern United States. The dataset contains 22 animal categories. The most common species are opossum, raccoon, and coyote. Approximately 70% of the images are labeled as empty.

3 Methods

In this paper, we propose a pipeline to tackle several of the major roadblocks preventing the application of deep learning techniques to camera trap images. Our proposed pipeline takes advantage of transfer learning and active learning to concurrently help with the transferability issue, multi-species images, inaccurate counting, and limited-data problems. In this section, we explain the details of our procedure and the motivations for each step.

3.1 Proposed pipeline

Our pipeline begins with running a pre-trained object detection model, based on the Faster-RCNN object detection algorithm [42], over the images. The pre-trained model is available to download [43]. We utilized version 2 of the

model. This version of the model has only one class – *animal* – and was trained on several camera trap datasets that have bounding box annotations available. We threshold the predictions of the model at 90% confidence and do not consider any detection with less than 90% confidence. The pre-trained object detection model accomplishes three related tasks:

1. It can tell us if an image is empty or contains animals; any image with no detections above 90% confidence is marked as empty.
2. It can count how many animals are in an image; we count animals by summing the number of detections above 90% confidence.
3. By localizing the animals, it can be employed to crop the images to reduce the amount of background pixels; we crop detections above 90% confidence and use these cropped images to recognize species in the next steps of the pipeline.

After running the object detection model over a set of images, we have already marked empty images, counted animals in each image, and gathered the crops to be further processed. Image classification models require fixed-sized inputs; since crops are variable in size, we resize all the crops to 256×256 pixels regardless of their original aspect ratio using bilinear interpolation. This set of cropped, resized images – which now contain animals with very little background – is the data we process with active learning.

There are two major challenges for applying active deep learning on a large, high-dimensional dataset: (1) We expect to have relatively few labeled images for our target dataset, typically far too few to train a deep neural network from scratch. Consequently, when training a model for a new dataset, we would like to leverage knowledge derived from related datasets (i.e., other camera trap images); this is a form of *transfer learning* [27]. (2) Active learning usually requires cycling through the entire unlabeled dataset to find the next best sample(s) to ask an oracle to label. Processing millions of high-dimensional samples to select active learning queries is impractically slow. One could approximate the next best points by only searching a random subset of the data, but that comes at the cost of inefficiency in the use of the oracle's time (i.e., they will no longer be labeling the most informative images).

Our proposed method allows us to evaluate all data points in order to find the most significant examples to ask humans to label, while retaining speed. Before processing the crops from a target dataset, we learn an *embedding model* (a deep neural network) on a large dataset, and use this model to embed the crops from our target dataset into a 256-dimensional feature space. The embedding model turns each image into a 256-dimensional *feature vector*. Using this technique we can both take advantage of transfer learning and significantly speed up the active learning procedure. The speedup occurs because when cycling over all data points we already have a low-dimensional feature vector to process, instead of needing to process each high-dimensional input by running it through a neural network. As discussed in sec. 2.6, we experiment with two embeddings produced by the cross-entropy and triplet losses, respectively (discussed more in sec. 4.3.1).

After obtaining the features for each crop in the lower-dimensional space, we have all the necessary elements to start the *active learning loop* over our data. We employ a simple neural network with one hidden layer consisting of 100 neurons as our *classification model*. We start the active learning process by asking the oracle to label 1,000 randomly-selected images. We then train our classification model using these 1,000 labeled images. At each subsequent step, we select 100 unlabeled images that maximize our image selection criteria (we will discuss different image selection strategies in sec. 4.3.2), and ask the oracle to label those 100 images; the classifier model is re-trained after each step. Another important step in our active learning algorithm is fine-tuning the embedding model periodically, which we do every 20 steps, starting after 2,000 images have been labeled.

Our pipeline is presented in pseudocode form as Algorithm 2.

4 Experiments and results

As explained above, our suggested pipeline consists of three steps: (1) running a pre-trained detector model on images, (2) embedding the obtained crops into a lower-dimensional space, and (3) running an active learning procedure. In this section, we report the results of our pipeline and analyze the contribution of these steps to the overall results. For these results, the eMammal Machine Learning dataset is used to train the embedding model, and the target dataset is Snapshot Serengeti. We chose eMammal Machine Learning for training our embedding because it is the most diverse of the available datasets and thus likely provides the most general model for applying to new targets. We chose Snapshot Serengeti as our target dataset to facilitate comparisons with the results presented in [16].

Algorithm 2 Proposed pipeline

```
Run a pre-trained object detection model on the images
Run a pre-trained embedding model on the crops produced by the objection detection model
Select 1,000 random images and request labels from the human oracle
Run the embedding model on the labeled set to produce feature vectors
Train the classification model on the labeled feature vectors
while Termination condition not reached do
    Select 100 images using the active learning selection strategy, pass these to the human oracle for labeling
    Fine-tune the classification model on the entire labeled set of the target dataset
    if number of examples % 2,000 == 0 then
        Fine-tune the embedding model on the entire labeled set of the target dataset
    end if
end while
```

4.1 Empty vs. animal

We run a pre-trained object detection model on the target dataset, and we consider images containing any detections above 90% confidence to be an image containing an animal (i.e. non-empty). The remaining images (containing no detection with more than 90% confidence) are marked as empty images. As the results in Table 1 show, the detector model has 91.71% accuracy, 84.47% precision, and 84.23% recall. Compare these results with those of [16] which are 96.83% accuracy, 97.50% precision, and 96.18% recall. We stress that that this accuracy came"for free", without manually labeling any image for the target dataset, while Norouzzadeh et al. [16] used 1.6 million labeled images from the target dataset to obtain their results. The pre-trained model was trained on the few camera trap datasets for which bounding box information exists; we expect this accuracy to improve as the pre-trained object detection model gets trained on larger, more diverse datasets.

Table 1: The confusion matrix for the pre-trained object detection model applied to the Snapshot Serengeti dataset

		Model Predictions	
		Empty	**Animal**
Ground Truth Labels	**Empty**	2,219,404	131,288
	Animal	133,769	714,276

4.2 Counting

Using a pre-trained object detection model allows us to not only distinguish empty images from images containing animals, but also to count the number of animals in each image. This simply means counting the number of bounding boxes with more than 90% confidence for each image. This straightforward counting scheme can give us the exact number of animals for 72.4% of images, and the predicted count is either exact or within one bin for 86.8% of images (following [6] we bin counts into 1, 2, ..., 9, 10, 11-50, 51+). Comparing to counting accuracy in Norouzzadeh et al., both the top-1 accuracy and the percent within +/- 1 bin are slightly improved, and this improvement comes "for free" (i.e., without *any* labeled images from the target dataset).

4.3 Species identification

After eliminating empty images and counting the number of animals in each image, the next task is to identify the species in each image. As per above, for species identification, we first embed the cropped boxes into a lower-dimensional space, then we run an active learning algorithm to label the crops. In the next three subsections, we discuss the details of each step and compare several options for implementing them.

4.3.1 Embedding spaces

As described above, we experimented with both (1) using features of the last layer of an image classification network trained on a similar dataset using cross-entropy loss, and (2) using features obtained from training a deep neural network using triplet loss [37, 44] on a similar dataset. We used the ResNet-50 architecture [13] for both treatments; only the loss function differs between these methods. After extracting the features with both techniques, we run the same active learning strategy on both sets of features. For these experiments, we chose the active learning strategy

that worked the best in our experiments (Sec. 4.3.2), which is the "k-Center" method [33] (Sec. 4.3.2 provides a brief description of the method).

After only 25,000 labels (a low number by deep learning standards), we achieved 85.23% accuracy for the features extracted from the last layer of a classification model and 91.37% accuracy with the triplet loss features. Fig. 2 depicts the t-SNE visualization of the learned embedding space. These results indicate that using triplet loss to build the embedding space provides better accuracy than features derived from an image classification model. As mentioned above, fine-tuning the embedding model periodically by using the obtained labels has a significant positive effect on improving accuracy. The jumps in accuracy (Fig. 3, 4, and 5) at 2K, 4K, 6K, ..., 28K clearly depict the advantage of fine-tuning the embedding model periodically. The results suggest it is better to use triplet loss with limited data.

The performance benefits of triplet loss likely stem from additional constraints placed on the embedding. Cross-entropy loss uses each sample independently, but in triplet loss, we use combinations of labeled samples (i.e., triplets), and we may reuse each sample in many triplets. For example, consider having 1,000 labeled images (10 classes, 100 samples each). In the cross-entropy loss scenario, we have 1,000 constraints over the weights of the network we optimize. In the triplet loss scenario, we can make up to 1,000 (choice of the anchor sample) $\times 99$ (choice of the positive sample) $\times 900$ (choice of the negative sample) $= 89,100,000$ constraints over the parameters. Using triplets thus provides 8,910 times more constraints than cross-entropy loss. These additional constraints help find a more informed embedding, and that improvement is qualitatively evident in Fig. 2. Of course, not all the possible combinations for triplet loss are useful, because many of them are easily satisfied. That is why we mine for hard triplets during training (Sec. S1). As we fine-tune the embedding model with far more labeled images, we expect the gap between the performance of cross-entropy loss and triplet loss to get smaller, because eventually both methods have sufficient constraints to learn a good embedding.

4.3.2 Active learning strategies

Different strategies can be employed to select samples to be labeled by the oracle. The most naive strategy is selecting queries at random. Here we try five different query selection strategies and compare them against a control of selecting samples at random. In particular, we try model uncertainty criteria (confidence, margin, entropy) [29], information diversity [45], margin clustering [46], and k-Center [33]. For all of these experiments, we use triplet loss features. Considering the expensive computational time and cost of each experiment, we only ran each experiment once. All the active learning strategies show performance improvement over the random baseline (Fig. 4). The highest accuracy is achieved with the k-Center strategy, which reaches 92.2% accuracy with 25,000 labels. The k-Center method selects a subset of unlabeled samples such that the loss value of the selected subset is close to the "expected" loss value of the remaining data points [33]. At 14,000 labels, we match the accuracy of Norouzzadeh et al. for the same architecture; compared to the 3.2 million labeled images they trained with, our results represent over a 99.5% reduction in labeling effort to achieve the same results.

4.3.3 Crops vs. full-image classification

As per above, we identify species in images that have been cropped by the object detection model. To assess the contribution of this choice to our overall accuracy, we also tried to classify species using full images. Fig 5 shows that using crops produces significantly better results on our data than using full images. This is likely because cropped images eliminate background pixels, allowing the classification model to focus on animal patterns.

5 Further improvement

This paper demonstrates the potential to significantly reduce human annotation time for camera trap images via active learning. While we have explored some permutations of our active learning pipeline, we have not extensively explored the space of parameters and algorithmic design choices within this pipeline. We believe there are at least three mechanisms by which our results could be improved.

1. Every deep learning algorithm has numerous *hyperparameters*, options selected by the data scientist before machine learning begins. For this paper, we used well-known values of hyperparameters to train our models. Tuning hyperparameters is likely to improve results. In particular, we only used the ResNet-50 architecture for embedding and a simple two-layer architecture for classification. Further probing of the architecture space may improve results.
2. We use a pre-trained detector, and we do not modify this model in our experiments. However, if we also obtain bounding box information from the oracle during the labeling procedure, we can fine-tune the detector model in addition to the embedding and classification models.

3. After collecting enough labeled samples for a dataset, it is possible to combine the classification and detection stages into a single multi-class detector model. This may improve accuracy, but almost certainly will improve computational efficiency when applying the model to new datasets.

6 Conclusion

Our proposed pipeline may facilitate the deployment of large camera trap arrays by reducing the annotation bottleneck (in our case, by 99.5%), increasing the efficiency of projects in wildlife biology, zoology, ecology, and animal behavior that utilize camera traps to monitor and manage ecosystems.

This work suggests the following three conclusions:

1. Object detection models facilitate the handling of multiple species in images and can effectively eliminate background pixels from subsequent classification tasks. Thus, detectors can generalize better than the image classification models to other datasets.
2. The embeddings produced by a triplet loss outperform those from a cross-entropy loss, at least in case of having limited data.
3. *Active learning*–machine learning methods that leverage human expertise more efficiently by selecting example(s) for labeling–can dramatically reduce the human effort needed to extract information from camera trap datasets.

References

[1] Jane Elith, Michael Kearney, and Steven Phillips. The art of modelling range-shifting species. *Methods in ecology and evolution*, 1(4):330–342, 2010.

[2] Gleb Tikhonov, Nerea Abrego, David Dunson, and Otso Ovaskainen. Using joint species distribution models for evaluating how species-to-species associations depend on the environmental context. *Methods in Ecology and Evolution*, 8(4):443–452, 2017.

[3] Thomas Richard Edmund Southwood and Peter A Henderson. *Ecological methods*. John Wiley & Sons, 2009.

[4] Allan F O'Connell, James D Nichols, and K Ullas Karanth. *Camera traps in animal ecology: methods and analyses*. Springer Science & Business Media, 2010.

[5] A Cole Burton, Eric Neilson, Dario Moreira, Andrew Ladle, Robin Steenweg, Jason T Fisher, Erin Bayne, and Stan Boutin. Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes. *Journal of Applied Ecology*, 52(3):675–685, 2015.

[6] Alexandra Swanson, Margaret Kosmala, Chris Lintott, Robert Simpson, Arfon Smith, and Craig Packer. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific data*, 2:150026, 2015.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[9] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[10] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[11] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. End-to-end attention-based large vocabulary speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4945–4949. IEEE, 2016.

[12] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *2012 Advances in Neural Information Processing Systems (NIPS)*, 2012.

[16] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018.

[17] Michael A Tabak, Mohammad S Norouzzadeh, David W Wolfson, Steven J Sweeney, Kurt C VerCauteren, Nathan P Snow, Joseph M Halseth, Paul A Di Salvo, Jesse S Lewis, Michael D White, et al. Machine learning to classify animal species in camera trap images: applications in ecology. *Methods in Ecology and Evolution*, 2018.

[18] Zhongqi Miao, Kaitlyn M Gaynor, Jiayun Wang, Ziwei Liu, Oliver Muellerklein, Mohammad Sadegh Norouzzadeh, Alex McInturff, Rauri CK Bowie, Ran Nathan, X Yu Stella, et al. Insights and approaches using deep learning to classify wildlife. *Scientific reports*, 9(1):8137, 2019.

[19] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European Conference on Computer Vision*, Munich, Germany, 2018.

[20] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[21] Martin T Hagan, Howard B Demuth, Mark H Beale, and Orlando De Jesús. *Neural network design*, volume 20. Pws Pub. Boston, 1996.

[22] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[23] Robert Hecht-Nielsen. Theory of the backpropagation neural network. *1989 International Joint Conference on Neural Networks (IJCNN)*, 1989.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012.

[26] Stefan Schneider, Graham W Taylor, and Stefan Kremer. Deep learning object detection methods for ecological camera trap data. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 321–328. IEEE, 2018.

[27] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *2014 Advances in Neural Information Processing Systems (NIPS)*, 2014.

[28] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2009.

[29] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR94*, pages 3–12. Springer, 1994.

[30] H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 287–294. ACM, 1992.

[31] Burr Settles and Mark Craven. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1070–1079. Association for Computational Linguistics, 2008.

[32] Yuhong Guo and Russell Greiner. Optimistic active-learning using mutual information. In *IJCAI*, volume 7, pages 823–829, 2007.

[33] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

[34] Aleix M Martínez and Avinash C Kak. Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233, 2001.

[35] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[36] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.

[37] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[38] Tavis Forrester, William J McShea, RW Keys, Robert Costello, Megan Baker, and Arielle Parsons. emammal–citizen science camera trapping as a solution for broad-scale, long-term monitoring of wildlife populations. *Sustainable Pathways: Learning from the Past and Shaping the Future*, 2013.

[39] emammal project. `https://emammal.si.edu`. Accessed: 2019-07-10.

[40] North american camera trap images. `http://lila.science/datasets/nacti`. Accessed: 2019-07-10.

[41] Caltech camera traps. `http://lila.science/datasets/caltech-camera-traps`. Accessed: 2019-07-10.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[43] Microsoft AI for Earth. Detection Models. `https://github.com/Microsoft/CameraTraps`, 2018. [Online; accessed 19-April-2019].

[44] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[45] Sanjoy Dasgupta and Daniel Hsu. Hierarchical sampling for active learning. In *Proceedings of the 25th international conference on Machine learning*, pages 208–215. ACM, 2008.

[46] Zhao Xu, Kai Yu, Volker Tresp, Xiaowei Xu, and Jizhi Wang. Representative sampling for text classification using support vector machines. In *European Conference on Information Retrieval*, pages 393–407. Springer, 2003.

[47] Pankaj K Agarwal, Sariel Har-Peled, and Kasturi R Varadarajan. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

Supplementary Information

S1 Triplet loss

Triplet loss is originally designed for problems with a variable number of classes such as human face recognition [37]. Recent studies [44] showed the effectiveness of triplet loss in learning a useful encoding. Triplet loss tries to put samples with the same label nearby in the embedding space, while samples with different labels are mapped to distant points in the embedding space. To train a network using triplet loss, we arrange the labeled examples into triplets. Each triplet consists of a baseline sampled image (the anchor), another sampled image with the same class as the anchor (positive), and a sampled image belonging to a different class (negative). For a distance metric d and a triplet (A, P, N), triplet Loss is defined as:

$$L = max(d(A, P) - d(A, N) + margin, 0) \tag{1}$$

In Eq. 1, *margin* is a hyperparameter specifying the minimum acceptable difference between $d(A, P)$ and $d(A, N)$. According to the definition of triplet loss, we have three types of triplets: (1) *easy triplets* which already satisfy the condition of triplet loss (i.e., the negative sample is much further than the positive sample to the anchor) and thus have a loss of zero, (2) *semi-hard triplets* in which $d(A, N) > d(A, P)$ but $d(A, N) < d(A, P) + margin$, and (3) *hard triplets* in which the negative sample is closer to the anchor than the positive sample. Easy triplets have a loss of zero and thus have no effect on training the weights of the network. Therefore, we omit them when arranging the triplets. Various strategies could be utilized to form the triplets such as choosing the hardest negative (the negative sample with maximal loss) or randomly choosing a hard or semi-hard negative for each pair of anchor and positive. Just like the original triplet loss paper [37], we use the random semi-hard negative strategy in this paper.

S2 Active learning selection criteria

Many query selection criteria have been proposed in the literature; for our experiments, we employ two criteria based on model uncertainty (confidence-based and margin-based selection [31]) and three criteria based on identifying dense regions in the input space (informative diverse[45], margin cluster mean[46]), and k-Center[33]. In this section, we summarize each of these criteria. For more details on active learning query selection criteria, refer to [31].

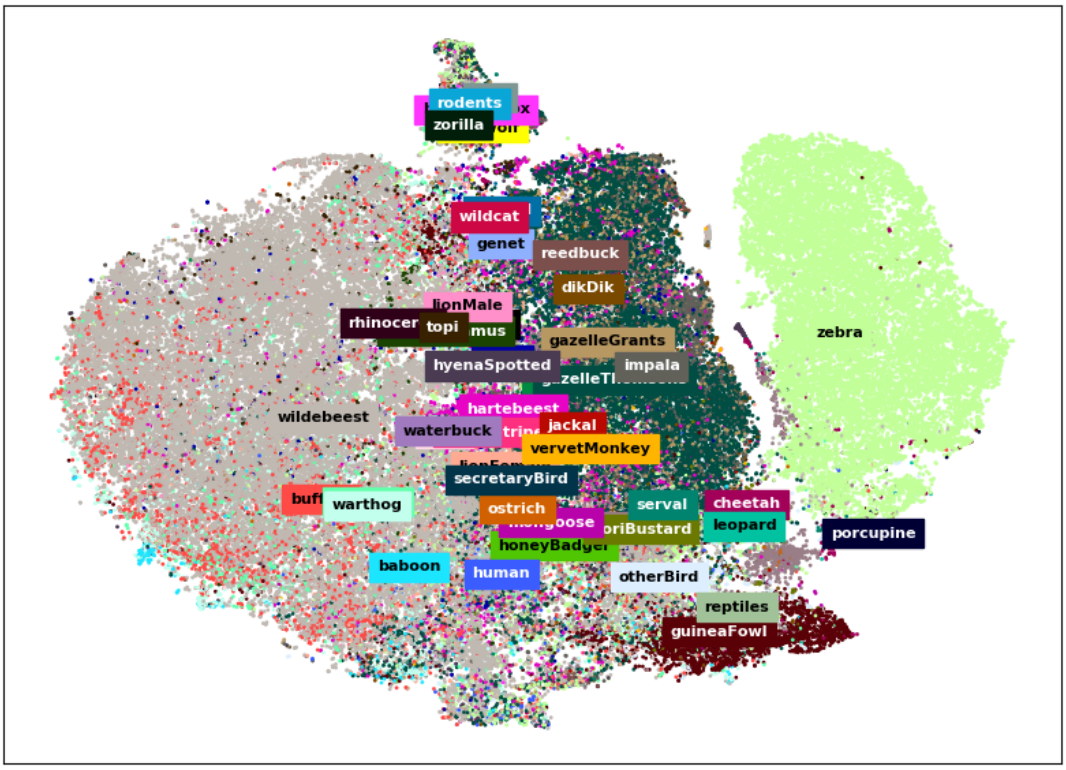
S2.1 Model uncertainty selection

Both the confidence-based and margin-based techniques belong to the model uncertainty selection category. The main assumption of these approaches is that when the underlying model is uncertain about predicting a sample, that sample could be more informative than the others. The uncertainty measure is interpreted from the model's output.

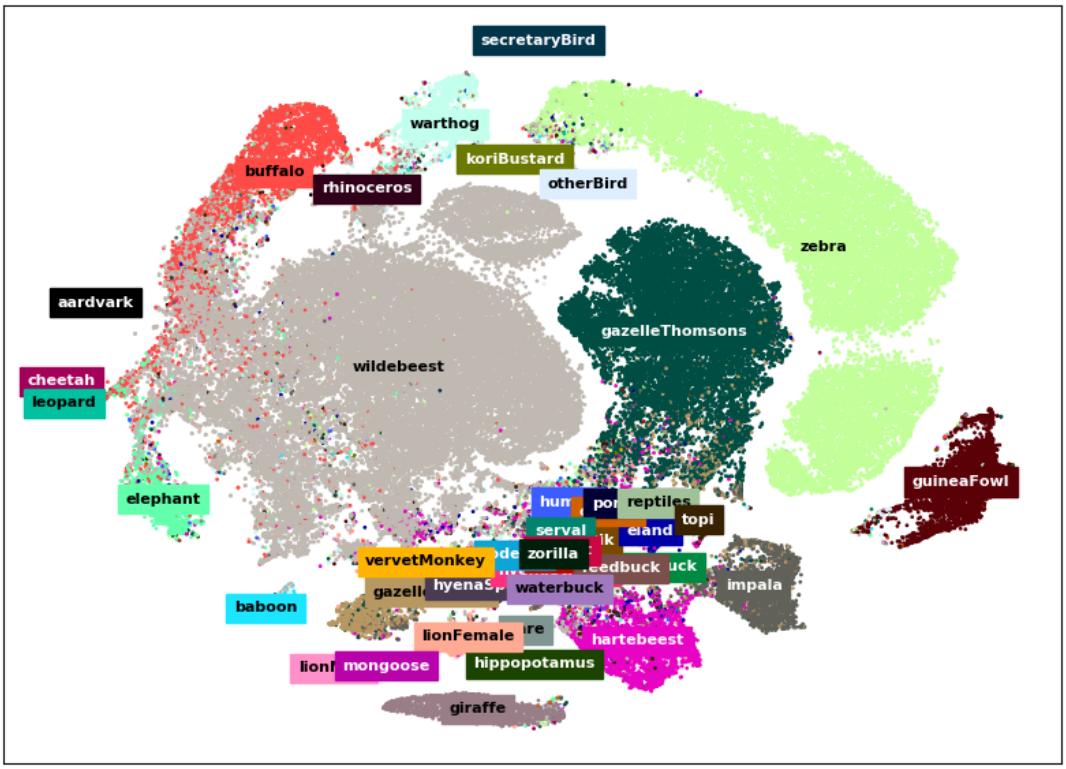
The confidence-based approach chooses the samples for which the model has the lowest confidence in the most probably class; the margin-based approach chooses the samples with the smallest gap between the model's most confident and second-most confident classes.

S2.2 Density-based selection

The primary assumption of these criteria is that for learning efficiently, we should not only query the labels of uncertain samples, but should also query those samples which are representative of many inputs, i.e. *dense* regions of the underlying input space. This assumption makes density-based methods more resilient to outliers. The informative diverse technique [45] first forms a hierarchical clustering of the unlabeled samples and then selects active learning queries so that the distribution of queries matches the distribution of entire data. The margin cluster mean criterion [46] clusters the samples lying within the margin of an SVM classifier trained on the labeled samples, and then selects the samples at cluster centers for human labeling. The k-center method [33], which has the best performance in our experiments, chooses a set of samples such that a model trained over the selected subset performs equally well on the remaining samples. The k-center method achieves this goal by defining the problem of active learning as a core-set selection problem [47] and then solving it.



(a) Softmax cross-entropy



(b) Triplet

Figure 2: t-SNE visualization of 100,000 randomly selected crops from the Snapshot Serengeti dataset with the embedding spaces produced by the (a) softmax cross-entropy loss and (b) triplet loss. The embedding based on triplet features shows a more intuitive, intelligent, separated distribution of species in the embedding space.

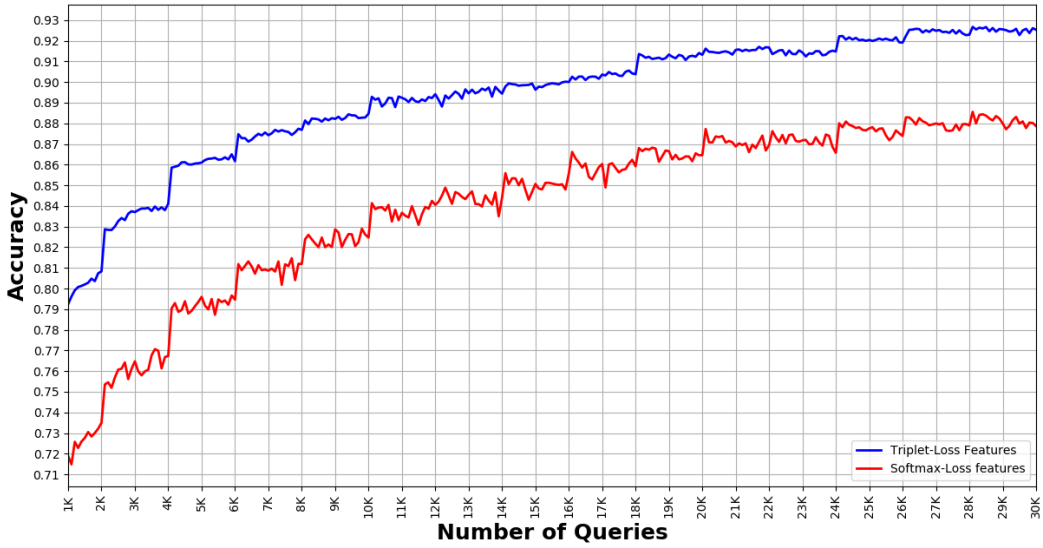


Figure 3: The accuracy of an active learning process using triplet loss features vs. using softmax cross-entropy loss features. Triplet loss features work better, but the gap closes as number of queries increases.

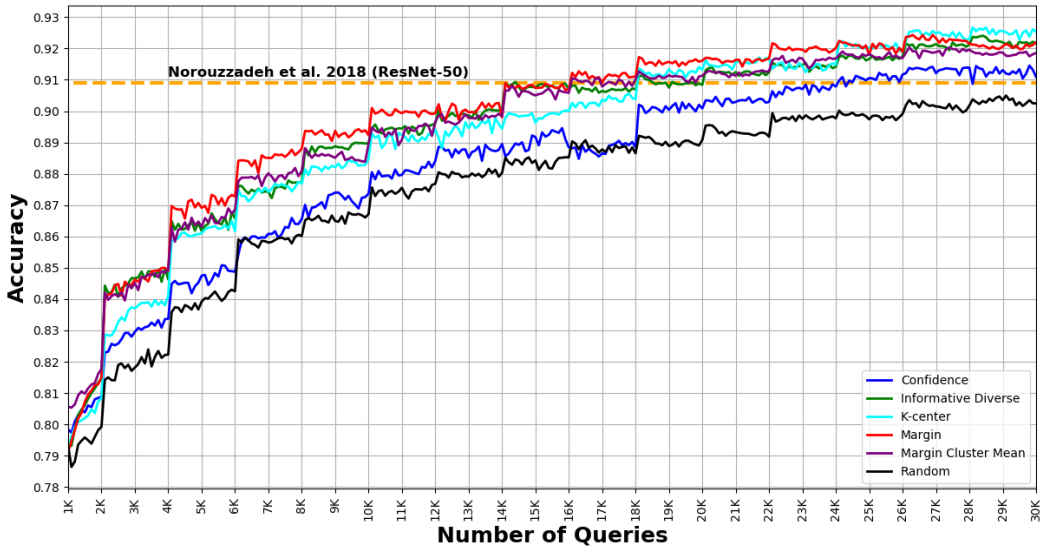


Figure 4: Performance of different active learning query strategies using triplet loss features over the Snapshot Serengeti dataset. k-Center achieves the best accuracy at 30,000 queries.

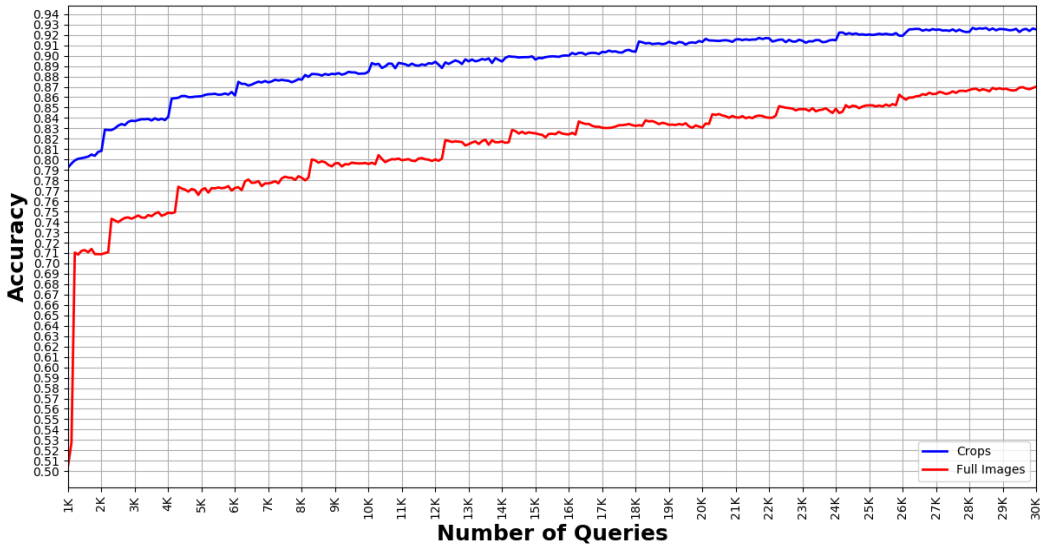


Figure 5: The accuracy of k-Center active learning using triplet loss features over crops vs. k-Center active learning using triplet loss features over full images on the Snapshot Serengeti dataset. Crops provide a substantial increase in accuracy.