Overview Paper

Deep Active Learning for Computer Vision: Past and Future

Rinyoichi Takezoe[1,2], Xu Liu[3], Shunan Mao[2], Marco Tianyu Chen[4], Zhanpeng Feng[1], Shiliang Zhang[2] and Xiaoyu Wang[1]*

[1]*Intellifusion Inc., China*
[2]*Peking University, China*
[3]*National University of Singapore, Singapore*
[4]*Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences (SIAT), China*

ABSTRACT

As an important data selection schema, active learning emerges as the essential component when iterating an Artificial Intelligence (AI) model. It becomes even more critical given the dominance of deep neural network based models, which are composed of a large number of parameters and data hungry, in application. Despite its indispensable role for developing AI models, research on active learning is not as intensive as other research directions. In this paper, we present a review of active learning through deep active learning approaches from the following perspectives: (1) technical advancements in active learning, (2) applications of active learning in computer vision, (3) industrial systems leveraging or with potential to leverage active learning for data iteration, (4) current limitations and future research directions. We expect this paper to clarify the significance of active learning in a modern AI model manufacturing process and to bring additional research attention to active learning. By addressing data automation challenges and coping with automated machine learning systems, active learning will facilitate democratization of AI technologies by boosting model production at scale.

Keywords: Computer vision, deep learning, active learning.

*Corresponding author: Xiaoyu Wang, fanghuaxue@gmail.com. The first four authors contributed equally to the paper.

1   Introduction

To democratize Artificial Intelligence (AI) for all industry verticals, researchers have been working on automating the machine learning design process commonly known as AutoML. Through techniques like neural network architecture search [69, 99], automated hyper-parameter optimization [3, 32], meta-learning [35, 36], etc., AutoML eases the requirements of domain knowledge and facilitates the building of high-quality machine learning models for developers from academia and industry.

Different from academia where the common practice is to use a fixed and pre-defined dataset for model iteration, the life cycle of developing an industrial model involves both model iteration and data iteration. Advances in deep learning have made it such that even standard deep models without careful design can perform decently. In contrast to the breadth of works advancing deep learning for developing sophisticated application-specific models, the investment in data iteration technologies is limited. However, robust data is pivotal to developing strong real-world models. The dataset largely determines the model's precision upper bound as well as its generalization capability when applied to novel scenes. In some scenarios, collecting more data is equally important, comparing with employing a more powerful machine learning algorithm. Therefore, the traditional paradigm of machine learning model development is arguably shifting from Model-centric to Data-centric approaches [5]. In our technology development process, a vast majority of efforts were spent on the curation of the right data, comparing to inventing better models when conducting AI model production. Therefore, automating or accelerating the data iteration process is the key for rapidly developing AI solutions.

In this paper, we focus on the process of data selection automation. Active learning (AL) is one of the core data selection automation technologies and has gained traction in recent decades. The goal of AL is to alleviate the expensive data labeling process and construct a resource-efficient yet robust training set by selecting the most valuable samples for human annotation. Specifically, AL operates in an iterative fashion. At each step, AL algorithms select and annotate informative samples from an unlabeled data pool, which are then used to train and update machine learning models. Although only a subset of the total data may be used, research suggests that this resource-efficient AL approach can still yield desired results. For example, models using AL identified data can use only 40% of the training data to achieve performance on par with that of using all data [18]. Furthermore, the improvement of AL may be even more significant in real-world industrial applications where data come from open-world scenarios and contain unavoidable noise, necessitating costly professional involvement.

Active learning methodologies are also evolving as deep learning is paving its way for all different kinds of applications. With millions or billions of parameters, training deep models requires much more data compared to that

of traditional machine learning algorithms, which makes efficient data selection technology increasingly critical. Deep Active Learning (DeepAL), an approach combining DL and AL to maximize task performance while minimizing the cost of data labeling, has attracted attention and shown great promise for various tasks such as image classification [115, 119], object detection [85, 88], text classification [4, 121], and sentiment analysis [11, 123].

There exist some efforts reviewing active learning related literature. [81] conducted a broad literature review for DeepAL in general AI applications and included relevant developments until 2020. In contrast, Wu *et al.* [109] provided a thorough survey concentrated on multi-label active learning for image classification tasks. More recently, Zhan *et al.* [118] delivered a comparative survey with an emphasis on the performance comparison of different DeepAL methods under varying tasks. While existing surveys either discuss active learning from the perspective of broad AI research including vision, nature language processing and machine learning, or dive deep into a specific vision task, there lacks a survey reviewing broader vision-based DeepAL methods. Moreover, active learning has the potential to significantly accelerate the data iteration or annotation in industrial systems. However, there is no discussion about how active learning is integrated into modern deep model manufacturing software and machine learning operations.

This paper provides a comprehensive survey about deep active learning in the context of computer vision from both academia and industry perspectives. Existing deep active learning research is far from mature and its application in practical systems is far from well explored. Through reviewing current technique developments, explaining and illustrating their applications in advanced artificial intelligence systems, and discussing unresolved problems, we hope this paper will serve as a succinct and relevant summary for the community.

The rest of the paper is organized as follows. First, we explain the basics of AL and present an overview of recent developments in DeepAL methods. Second, we discuss the usage of DeepAL algorithms for computer vision tasks, including image recognition, object detection, semantic segmentation, and video recognition. Third, we provide an introduction to the practical applications of DeepAL methods in industry software or systems. Lastly, to promote the prosperity of DeepAL, we discuss current limitations and potential directions in this field.

2 Overview of Active Learning

2.1 What is Active Learning?

Active learning is the task of choosing the most valuable data for a learning algorithm so that it can perform similarly or even better with the resulting less training data. More practically, an active learning system selects the most valuable

unlabeled samples by a query strategy, then sends them to a human for labeling or curation. The labeled data is then adopted for next stage model training.

There are arguably three commonly studied active learning scenarios in existing literature, including stream-based selective sampling, membership query synthesis, and pool-based sampling, respectively as shown in Figure 1. Stream-based selective sampling processes every unlabeled sample independently, so the resulting number of samples for labeling is not fixed. This sampling rule is suitable for sequentially provided online data, e.g. Robot-Assisted AL [98]. In this scenario, it is hard to calibrate the threshold to determine whether a sample should be selected or not. Membership query synthesis generates new data to be annotated. It is efficient in the interpretable feature space, but may suffer from low-quality generation because of the limited knowledge of data distribution in unseen domains. Pool-based sampling chooses the best query samples among the entire unlabeled set. It commonly computes a score function for each unlabeled sample and makes the selection according to the score, e.g., samples with the highest scores are sent for annotation. The vast majority of DeepAL methods in the literature use pool-based sampling and they try to select the most valuable data from the unlabeled pool. Some DeepAL methods adopt membership query synthesis with deep image generation methods.
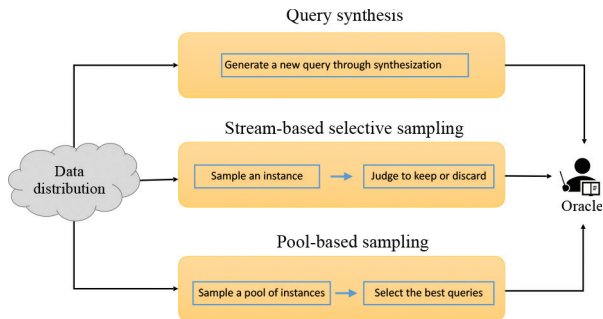

Figure 1: Three scenarios of active learning. 'Pool-based Sampling' is used in most DeepAL settings.

2.2   Deep Learning-based Methods

2.2.1   Generic Deep Active Learning Framework

The advent of deep learning has made DeepAL a popular topic. The primary characteristic of DeepAL is that it heavily relies on batch-based sample querying. Traditional per-sample query methods lead to frequent model retraining. Since each retraining of the model is computationally expensive, the one-by-one query method is inefficient for deep learning. Batch mode query strategies are

frequently employed in DeepAL to strike a balance between sampling (for annotation) cost and training cost. In a cycle of active learning, the first step is to get the optimal deep model using existing labeled data. Then, the most valuable batch from the unlabeled pool is selected using active learning based on features extracted by the resulting deep model. The performance of the deep model is expected to converge after several rounds of model training and active learning.

In each cycle, one can choose to retrain the deep model or just perform fine-tuning solely using newly collected data. While retraining a model could be very costly, fine-tuning a model could cause divergence or drifting from the original model. Except for a few works, e.g. CEAL [105], which fine-tune the model, most deep active learning methods retrain the deep model from scratch at each active learning cycle, to ensure the model performance is not compromised. With the fast growth of GPU computation power, the cost of training a deep model is also decreasing. The number of active cycles depends on the computation budget and time budget of a project. A project could last for weeks or years.

A generic deep active learning framework can be formulated as follows. Assuming the whole active training procedure consists of T cycles, where each cycle is further divided into two steps: the exploitation step and the exploration step. The exploitation step trains a deep model $f(\cdot; \theta^t)$ from the initial state with labeled samples $D_l = \{(x_i, y_i)\}_{i=1}^M$, where θ^t are the parameters of the deep model to be optimized in the t-th cycle. The exploration step selects a subset of unlabeled samples for labeling. First, each sample from the unlabeled pool $x \in D_u = \{(x_j)\}_{j=1}^N$ receives an informativeness score denoted as $\text{Score}(x, \theta)$. It is obtained based on extracted sample features or predictions by the deep model with the parameters θ. Then, a batch of samples with the highest scores are selected and sent to the labeling experts. Finally, the labeled batch joins the existing labeled set to formulate a new training set. Therefore, the t-th cycle in a typical active learning process can be formulated as

$$
\begin{aligned}
\theta^t &= \arg\min_{\theta} \sum_{\{x,y\} \in D_l^t} \mathcal{L}(f(x; \theta), y) \\
\mathcal{B}^t &= \arg\max_{\mathcal{B} \subset D_u^t} \sum_{x \in \mathcal{B}} \text{Score}(x, \theta^t) \\
D_l^{t+1} &= D_l^t \cup \mathcal{B}^t,
\end{aligned}
\tag{1}
$$

where D_l^t and D_u^t denote the labeled set and the unlabeled pool in the t-th cycle. $\mathcal{L}$ is the loss function employed to train the deep model $f(\cdot; \theta)$, and $\text{Score}(\cdot)$ is the function measuring the information obtained from each unlabeled sample.

Most existing DeepAL research focuses on solving the following two problems: (1) selecting the most valuable batch and (2) training a deep model with insufficient labeled data. Related methods will be discussed in the following two parts.
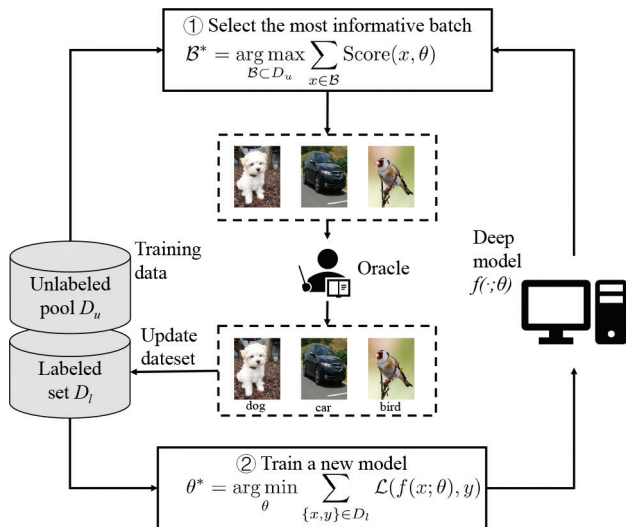
Figure 2: A typical deep active learning framework for vision tasks.

2.2.2 Selecting the Most Valuable Batch

DeepAL aims to achieve high performance with limited annotation budgets. To make full use of the annotation budget, it is critical to select a batch of informative samples to maximize performance gain. To achieve this goal, different methods have been proposed to measure the informativeness of each sample. These methods mainly identify informative samples by two criteria: samples that the model is uncertain about and samples that have a different feature distribution with respect to already labeled data. Some works also combine these two criteria to perform sample selection.

Model Uncertainty-based Approaches: The uncertain samples can be regarded as hard-to-learn samples for the current model. Using uncertain samples in the training loop can let the model focus on the task that it is not good at, instead of continuing to learn from simple samples. The uncertainty can be evaluated using the model's predictions. The following part reviews classic methods that are widely used in traditional AL.

Least Confidence (LeastConf) [65] selects data points whose predicted labels $\hat{y}$ have the lowest posterior probability, i.e.,

$$\text{Score}(x, \theta) = -p(\hat{y}|x, \theta), \tag{2}$$

where θ denotes the model parameters, $p(y|x, \theta)$ is the posterior probability that the sample x is predicted to class y by θ.

Margin [82] selects data points, where the difference between the highest predicted probability and the second-highest predicted probability is small. The corresponding score function can be denoted as,

$$\text{Score}(x, \theta) = -[p(\hat{y_1}|x, \theta) - p(\hat{y_2}|x, \theta)], \tag{3}$$

where $\hat{y_1}$ and $\hat{y_2}$ are the first and second predictions with the highest probability. A small difference indicates that the model is not able to distinguish one class from the other.

Entropy [56] selects data points that maximize the predictive entropy:

$$\begin{aligned}\text{Score}(x, \theta) &= \mathbb{H}[y|x, \theta] \\ &= -\sum_c p(y_c|x, \theta) \log p(y_c|x, \theta).\end{aligned} \tag{4}$$

Compared with LeastConf and Margin, which only consider predictions with the highest probability, Entropy utilizes the whole posterior probability distribution to measure the uncertainty. A larger entropy hence suggests a higher sample uncertainty.

Bayesian Active Learning by Disagreements (BALD) [52] aims to minimize the uncertainty of parameters. It can be approximated as selecting data points expected to maximize the conditional mutual information between model predictions and model parameters. The uncertainty score can be denoted as,

$$\begin{aligned}\text{Score}(x, \theta) &= \mathbb{I}[y, \theta|x, D_l] \\ &= \mathbb{H}[y|x, D_l] - \mathbb{E}_{p(\theta|D_l)}[\mathbb{H}[y|x, \theta]].\end{aligned} \tag{5}$$

Maximizing Equation (5) leads to seeking the data points where the model is most uncertain about the average predictions y of the probabilistic model. Meanwhile, predictions of these data points under a certain setting, e.g. data augmentation, are confident.

Mean Standard Deviation (Mean STD) [58] measures the model uncertainty by the mean standard deviation of the posterior probabilities over all classes:

$$\text{Score}(x, \theta) = \frac{1}{C} \sum_c \sqrt{Var_{q(\theta)}[p(y_c|x, \theta)]}. \tag{6}$$

Most of these score functions perform well in traditional active learning, but can not be extended to DeepAL. One of the reasons is that many score functions rely on model prediction, which easily become overconfident due to the characteristic of the softmax layer in a CNN model. In order to adapt the above methods to the DeepAL, Gal *et al.* [39] proposes Deep Bayesian Active Learning (DBAL) that combines Bayesian Convolutional Neural Networks (Bayesian CNNs) into the active learning framework. Bayesian CNNs

perform variational inference to approximate the true model posterior, which provides model uncertainty cues that can be used in the score functions. In practice, DBAL makes use of Monte-Carlo dropout (MC dropout) stochastic regularisation technique to perform approximate inference.

Similarly, another work [9] proposes Ensembles which uses an ensemble of CNNs to estimate prediction uncertainty. Specifically, Ensembles trains an ensemble of N classifiers with the same D_l and same network architecture, but with different random weight initializations. It uses the average softmax vectors of N classifiers as the output. The score functions can be applied to Ensembles as well. Ensembles generally perform better than DBAL due to the larger model capacity and the diversity of ensemble models.

In addition to estimating the uncertainty from outputs statistically, there also exist some methods that estimate the uncertainty from other perspectives. Ducoffe and Precioso [30] argue that commonly-used methods might be over-confident since a small modification on a sample can lead to an unexpected misclassification. They, therefore, propose Deep-Fool based Active learning (DFAL) to select data points near the decision boundary. DFAL approximates the distance between a sample and the decision boundary by the distance between this sample and its nearest adversarial sample. The selected and adversarial samples are added to the training set.

Another kind of method tries to generate samples with the highest uncertainty instead of selecting from available samples. They are more similar to active learning by query synthesis than standard pool-based active learning. Zhu and Bento [124] propose Generative Adversarial Active Learning (GAAL) which combines GAN and active learning. They generate query samples with the largest uncertainty to get good performance with the least number of labeled data. Tran *et al.* [101] introduce more data generation methods like Bayesian data augmentation to generate samples in the intersection of different classes.

According to the proposed influence function by Koh and Liang [63] and Wang *et al.* [107] have theoretically proven that selecting data points of higher gradient norm leads to a lower upper-bound of test loss, resulting in better test performance. However, the loss cannot be computed with a lack of labeled data. To address this issue, Ash *et al.* [6] proposes Batch Active learning by Diverse Gradient Embeddings (BADGE) that assigns pseudo labels based on model predictions, and Wang *et al.* [107] propose two schemes named expected-gradnorm and entropy-gradnorm, respectively. These schemes calculate expected loss and entropy as a proxy.

Since computing gradient norms needs to estimate the loss first, Yoo and Kweon [115] propose a task-agnostic method that directly predicts loss to measure the uncertainty. This method is called Learning Loss for Active Learning (LL4AL). LL4AL attaches a small parametric loss prediction module to the task model, and jointly trains both models. The loss prediction module is trained to predict the losses of the task model for unlabeled samples. Samples with high loss will be selected for annotation and training.

Feature Distribution-based Approaches: Another category of methods selects samples that are different from labeled ones in the feature space. Once a sample is labeled, it can be regarded as a representative of the feature distribution. This makes it no longer necessary to annotate similar samples. The key intuition is that if selected features are able to represent the true data distribution, the model trained with AL will perform similarly to the model trained with the full data.

A straightforward method is to utilize the distance or similarity between data points in the feature space and avoid selecting similar samples. Yin *et al.* [114] propose a similarity-based method consisting of two stages: exploitation and exploration. The exploitation stage selects samples with maximum uncertainty and minimum redundancy, where the redundancy is measured by the similarity within the selected set. The exploration stage selects samples farthest from the labeled set to find diverse samples. Bıyık *et al.* [14] employ k-Determinantal Point Processes (k-DPPs), a class of repulsive point processes that select a batch of samples with probability proportional, to the determinant of their similarity matrix.

Geifman and El-Yaniv [42] and Sener and Savarese [88] define the problem as core-set selection, where the goal is to find a subset such that a model learned on it can achieve comparable performance with learning on the entire dataset. Sener and Savarese [88] theoretically prove that the objective is equivalent to the k-center clustering problem. Since the problem is NP-Hard, Sener and Savarese [88] use the Farthest-First (FF) traversal to get a greedy approximation in practice. Given a set of unlabeled samples, FF traversal iteratively picks samples farthest from the labeled samples in the feature space generated by the representation layer of the neural network.

Adversarial learning has also been adopted in AL for querying samples. Specifically, adversarial learning can be conducted between a classifier and a sample selection model. Gissin and Shalev-Shwartz [43] propose Discriminative Active Learning (DAL), which poses active learning as a binary classification task between labeled and unlabeled classes. Choosing samples to label in this way makes the labeled and unlabeled data indistinguishable. Similarly, Sinha *et al.* [96] propose Variational Adversarial Active Learning (VAAL), which uses a β-variational autoencoder (β-VAE) to map labeled and unlabeled data into the same latent space and trains an adversarial network to distinguish the encoded features. The β-VAE and the adversarial network are learned together in an adversarial fashion. Shui *et al.* [93] further propose Wasserstein Adversarial Active Learning (WAAL) that adopts Wasserstein distance for distribution matching. They reveal that Wasserstein distance can better capture the diversity.

Hybrid Approaches: It is reasonable to combine the above two criteria for active learning, i.e., utilizing both model prediction and model features for sample selection. The model prediction is helpful for identifying difficult

data points but might ignore the correlation among them. The model feature makes it easier to identify representative data points by considering feature relationships but suffers from degraded effectiveness when the query batch is small. A variety of hybrid strategies have been proposed to leverage both complementary criteria, and have demonstrated better performance.

A simple hybrid approach is to divide the query procedure into two stages, and different types of samples are selected at each stage. BADGE [6] computes the gradient embedding for each unlabeled sample in the first stage, then uses the k-means++ seeding algorithm for sampling in the second stage. BADGE tends to select samples with both high diversity and magnitude. Zhdanov [122] first prefilter top βk ($\beta \geq 1$) informative samples based on Margin, then cluster them to k clusters with (weighted) k-means. Finally, k samples closest to the cluster center are selected. In contrast, Cluster-Margin [23] first performs Hierarchical Agglomerative Clustering (HAC) as a preprocessing step. For subsequent sampling iterations, Cluster-Margin retrieves clusters on which the model is least confident, and uses a round-robin scheme for sampling. Compared with other methods that need to run a diversification algorithm at each sampling iteration, Cluster-Margin enjoys better efficiency because HAC is only executed one time.

Another commonly used approach is to design a unified score function that considers both uncertainty and diversity. Yin *et al.* [114] propose a linear combination of two criteria as the score function for the exploitation step, i.e.,

$$\text{Score}(x, \theta) = \mathbb{H}[y|x, \theta] - \alpha \, \text{Sim}(x, S), \tag{7}$$

where $\text{Sim}(\cdot)$ computes the similarity between one sample and the selected set, and α denotes a parameter to balance uncertainty and diversity. During the acquisition process, the uncertain samples different from the selected set are chosen one by one.

Kirsch *et al.* [62] extend BALD [52] to BatchBALD to compute the mutual information between a batch of data points and model parameters, instead of independently computing for each single sample,

$$\begin{aligned}
\text{Score}(x_{1:b}, \theta) &= \mathbb{I}[y_{1:b}, \theta | x_{1:b}, D_l] \\
&= \mathbb{H}[y_{1:b}, |x_{1:b}, D_l] - \mathbb{E}_{p(\theta|D_l)}[\mathbb{H}[y_{1:b}, |x_{1:b}, \theta]],
\end{aligned} \tag{8}$$

where $x_{1:b}$ and $y_{1:b}$ denote b selected data points and their predictions. BALD may overestimate the joint mutual information due to double counting of overlaps between data points in the feature space. BatchBALD takes this issue into account and makes itself more likely to acquire a diverse cover.

Those methods need to compute the correlation between samples. This constraint suppresses the capability of computing representativeness scores individually for each sample. WAAL [93] tackles this issue by using a discriminator

to directly estimate the probability that a sample comes from the unlabeled pool.

Moreover, compared with explicitly considering the hybrid query strategy, some works aim to balance uncertainty and representativeness in an implicit way. Zhang *et al.* [119] propose a State Relabeling Adversarial Active Learning model (SRAAL) which is motivated by the fact that previous works such as VAAL [96] consider all the unlabeled samples of the same quality. They build the discriminator that maps data to a hard label, i.e., 0 and 1 for unlabeled and labeled data, respectively. To alleviate the issue of hard labels, an online uncertainty indicator in the discriminator is designed to relabel the state information of unlabeled data according to different importance. Task-Aware Variational Adversarial Active Learning (TA-VAAL) [60] is another extension based on VAAL [96], which combines with LL4AL [115]. By exploiting RankC-GAN [86], the goal of LL4AL is relaxed from predicting the accurate loss value to estimating ranking loss cues.

2.2.3   Training with Insufficient Data

Deep learning is data-hungry, and its high performance relies heavily on numerous training data. Standard active learning methods train deep models on the labeled sample set and ignore the unlabeled pool. Therefore, many semi-supervised methods have been proposed to alleviate the issue of insufficient training data. Semi-supervised learning and active learning are the two sides of the same coin in that, both pursue to achieve good performance with the fewest labeled data. The former focuses on the utilization of unlabeled data, while the latter focuses on selecting and annotating the most valuable data. The combination of semi-supervised learning and active learning has the potential to significantly increase the variety of training data. Semi-supervised learning methods in active learning can be achieved by three means: Pseudo-label, self-supervised learning and data generation.

Pseudo-label methods assign pseudo labels to samples in the unlabeled pool. Wang *et al.* [105] propose CEAL to assign pseudo labels according to model predictions. They progressively select two types of samples from the unlabeled set to fine-tune the deep model. Samples with high uncertainty are selected for labeling. Samples with high confidence are assigned pseudo labels without human cost. Siméoni *et al.* [95] propose a transductive way to generate pseudo labels. Following [54], the prediction of the unlabeled data is made according to the label propagation on their nearest neighbors. Model predictions of the unlabeled data are used to compute the certainty of pseudo labels. Gorriz *et al.* [45] propose a similar cost-efficient method to leverage unlabeled samples in active medical segmentation. They reduce the uncertainty map to a single uncertainty value for active sample selection and

pseudo label assignment. Samples with the lowest uncertainty are added to the training set along with their predicted masks.

Self-supervised methods utilize the unlabeled samples for training. Liu et al. [70] propose an unsupervised feature learning on all datasets to improve the training efficiency of Deep Belief Network (DBN). Gao et al. [40] propose a consistency-based unsupervised training to learn from unlabeled data by optimizing the feature distance between unlabeled samples and their augmentations. In addition, they also propose a consistency-based sample selection strategy to choose augmented samples with the largest inconsistency. Yu et al. [116] use a similar consistency-based training which considers both box regression and classification in detection tasks at the same time. Kim et al. [61] use Mixup [120] to generate augmented samples during training and propose an acquisition function which looks ahead at the effect of data augmentation. As data augmentation has exhibited great potential in semi-supervised training, future active learning works can integrate with more powerful data augmentation strategies to alleviate the expensive data annotation, and improve the performance of training with insufficient data.

3   Vision Tasks Empowered by DeepAL

Thanks to its capability of selecting the most valuable samples and saving annotation costs, active learning has been widely used in deep model training for various vision tasks. This section proceeds to introduce active learning methods applied in different vision tasks.

3.1   Image Classification and Recognition

Image classification and recognition is the cornerstone of many vision tasks. The majority of studies on active learning in the vision community are conducted on this task.

Traditional active learning methods for image classification can be summarized into three categories: uncertainty-based approaches, diversity-based approaches, and expected model change, respectively. In the category of uncertainty-based approaches, LeastConf, Margin and Entropy are simple and effective ways to query informative samples. Also, distances to the decision boundary of the classifier can be used to evaluate the uncertainty [66, 100]. Some other authors [76, 90] train multiple models to construct a committee and measure the uncertainty by disagreement between different members of the committee. In the diversity-based approaches, [77] cluster the data pool to select representative samples. [31, 47, 112] optimize the data selection in a discrete space. [13, 48, 74] consider the distance to surrounding data points and choose a subset to represent the global distribution. Expected model change

aims to find samples that can cause substantial changes to the current model. It can be estimated from different perspectives, such as gradient length [89], future errors [83] and output changes [38, 57], respectively.

Traditional methods introduced above cannot scale well to large-scale data and deep neural networks. Recent years have witnessed many active learning methods designed for deep learning. For example, [25, 68, 70] study the hyperspectral image classification for remote sensing. Liu *et al.* [70] train a DBN that includes two stages: unsupervised feature learning and supervised fine-tuning. The first stage can be used to estimate the representativeness, while the second stage is used to estimate the uncertainty. Medical image analysis requires medical experts for labeling, thus also faces the difficulty of obtaining well-annotated data. Many works attempt to introduce active learning to reduce the annotation cost [15, 28, 37, 87]. Stark *et al.* [97] solve the CAPTCHA recognition by performing active learning. Their method does not require oracles to annotate a large training set. Once the classifier solves a CAPTCHA correctly, the label is automatically obtained. Since using all correctly classified data for re-training would be inefficient, only the most uncertain samples are selected for training.

3.2 Object Detection

Object detection requires the model to locate and classify objects simultaneously, thus it is more complicated than image classification. The demand for instance-level annotation imposes a higher annotation cost than labeling image categories, making active learning for object detection valuable. In recent years, active learning for object detection has attracted more attention. According to their motivations, those works can be divided into two categories: proposing specific query strategies for object detection and reducing annotation costs.

3.2.1 Query Strategy

Query strategy for object detection denotes the way of selecting samples for annotating object bounding boxes. Most existing works follow the setting that queries the whole image and annotates all objects in those images. Although various query strategies have been proposed in image classification to find valuable samples, these methods do not perform well when directly applied to object detection. One reason is that those methods make image-level queries based on instance-level predictions, which leads to inconsistency.

First, we introduce traditional active learning methods for object detection. Yao *et al.* [113] use Hough forests as the detector, which detects objects using the generalized Hough transform and randomized decision trees. They estimate an optimal detection threshold based on the distribution of detection scores, which is hence used to model the annotation cost. Images with the highest annotation cost are selected because they are considered as most informative

samples. Bietti [12] use a linear SVM in a sliding window as the detector, and query points closest to the decision boundary. Roy *et al.* [84] model the object detection as a structured regression problem, and develop an active learning algorithm using the principled version space approach in the "difference of feature" space for structured prediction. The algorithm selects points for labeling such that the version space is maximally reduced.

With the development of CNN, deep active learning for object detection has been extensively studied. The uncertainty-based methods are widely used for querying samples. Feng *et al.* [33] use MC dropout and Deep Ensembles to obtain uncertainty estimations and select the most uncertain samples by various score functions, e.g., entropy, mutual information. Wang *et al.* [104] estimate the uncertainty by performing the cross image validation, i.e., pasting proposals from the unlabeled image into a certain annotated image, then evaluating the prediction consistency under different image contexts. Region proposals with low consistency will be asked for annotation. Roy *et al.* [85] propose a query by committee paradigm based on the SSD detector. The disagreement for a particular candidate bounding box among different convolution layers is used for selecting samples. The intuition is that a detection network is likely to have similar predictions in a local region. Similarly, Aghdam *et al.* [2] compute pixel-level uncertainty scores by the divergence among predictions at different levels of decoder for each pixel and their neighborhoods. Furthermore, the pixel-level scores are aggregated into an image-level score, which is used to select informative samples. Brust *et al.* [16] evaluate a set of aggregation functions, including calculating the sum, the average and the maximum over all detection results. Yuan *et al.* [117] argue that many methods ignore the interference from a large number of noisy instances in object detection, which leads to the inconsistency between instance-level and image-level uncertainty. Therefore, they introduce a multiple instance learning module to re-weight instance uncertainty. This method is expected to highlight the informative instances and depress noisy ones.

The above-mentioned methods mainly focus on the classification part in object detection for selecting samples. Some other works also consider the bounding box regression for selecting informative samples. Choi *et al.* [22] learn a Gaussian Mixture Model (GMM) for the network outputs including both localization and classification. Consequently, both aleatoric and epistemic uncertainties are computed by parameters of the GMM, which include mean, variance and mixture weights. Kao *et al.* [59] introduce two metrics of localization uncertainty: Localization Tightness with the Classification Information (LT/C) and Localization Stability with the Classification Information (LS+C). LT/C is based on the overlap ratio between the region proposal and the final prediction. LS+C is based on the variation of predicted object locations when input images are corrupted by noises. Yu *et al.* [116] apply data augmentations to unlabeled images, and measure the consistency between predictions of the

original and augmented images. In contrast to LS+C, which considers box regression and classification independently, [116] unify them with a single metric.

The diversity-based method has also been explored in object detection. Since the large receptive fields in deep CNNs may cause confusion among spatially neighboring classes, Agarwal *et al.* [1] propose contextual diversity to measure the confusion associated with spatially co-occurring classes. Such a measure helps to select samples with diverse spatial contexts. Moreover, [110] presents a hybrid method that considers both uncertainty and diversity. The uncertainty is estimated by basic detection entropy and the instance diversity can be divided into intra-image diversity and inter-image diversity. For the intra-image diversity, entropy-based Non-Maximum Suppression is performed to remove redundant instances within images. For the inter-image diversity, the class-specific prototypes of each image can help to improve the intra-class diversity, and the adaptive budget size for each class can enhance the inter-class diversity.

3.2.2   Annotation Reduction

In addition to the above works studying query strategies, there are other works aiming to further reduce the annotation cost. Desai *et al.* [27] propose an adaptive supervision framework that combines active learning with weakly supervised learning. All selected samples are first annotated with weak labels, i.e., center locations of objects. Those weak labels are then used to generate pseudo labels. Accurate bounding box annotations are required only for objects where the model can not produce confident predictions. The cost of different annotation types is measured by median annotation times in statistics. Pardo *et al.* [80] also consider a hybrid training set consisting of both weak and strong labels. Given a fixed budget, appropriate annotation action is chosen for each image based on certain measurements such as the uncertainty score. Then a teacher-student model is trained to make use of both types of annotations. Since the annotation cost of each image depends on the number of objects present in the image, Desai and Balasubramanian [26] propose a fine-grained sampling strategy to selectively pick the most informative subset of bounding boxes rather than whole images.

3.3   Image Segmentation

Image segmentation is also a fundamental task in computer vision with wide applications like scene understanding, medical image analysis, robotic perception, and autonomous driving. Image segmentation classifies each pixel in an image into a specific class. Annotating ground-truth for segmentation is hence more expensive than labeling image semantics and categories. Therefore, it is appealing to use active learning to relieve the annotation burden.

Existing active learning methods for image segmentation can be divided into image-based methods and region-based methods according to the granularity of selected data for annotation. Image-based approaches consider each image as a sample while region-based approaches divide unlabeled images into non-overlapping regions and consider each region as a sample.

3.3.1   Image-based Active Segmentation

Image-based approaches for segmentation share the same intuition as standard pool-based batch-mode active learning. Suggestive Annotation [111] is a milestone work in deep active segmentation. It presents a new deep active learning framework by combining FCNs and active learning. Samples are first sorted by uncertainty and then the Core-set [88] is applied to select the batch for annotation. Instead of using another image descriptor network, it computes the overall uncertainty of each training sample with the mean of uncertainty of all pixels and calculates feature descriptors for the core-set by average pooling the output of the last convolution layer. Ozdemir *et al.* [79] propose a Borda-count based sample querying strategy, which considers both uncertainty and representativeness. Uncertainty is measured by the variance of predictions in their model. Representativeness is measured by "Content Distance", which is computed on the whole outputs of the last convolution layer before spatial pooling. Wang *et al.* [108] propose a Nodule R-CNN model that attains state of-the-art pulmonary nodule segmentation performance. In addition, they train the model with a weakly-supervised method to leverage both labeled and unlabeled samples.

VAAL [96] proposes another framework for active image segmentation. It learns a VAE together with an adversarial network trained to discriminate unlabeled and labeled data. The proposed method implicitly learns the uncertainty for samples from the unlabeled pool. SRAAL [119] designs an online uncertainty indicator, which endues unlabeled samples with different importance. Unlabeled samples with high confidence are treated as simple samples, and will not be selected for annotation. TA-VAAL [60] proposes to incorporate LL4AL and RankCGAN into VAAL by relaxing loss prediction with a ranker for ranking loss information. Agarwal *et al.* [1] introduce a novel information-theoretic distance measurement named Contextual Diversity (CD) to capture the diversity in spatial and semantic contexts of various object categories.

3.3.2   Region-based Active Segmentation

Different from image-based approaches, many active segmentation methods belong to region-based approaches. Region-based active learning is efficient since only a small number of regions is required to be labeled for each time.

In addition, since the Region of Interest (ROI) and confusing areas may only occupy a small percentage of the whole image, segmenting the whole image is not necessary. Those properties make region-based approaches perform better than image-based approaches under the same annotation budge.

One kind of region-based active segmentation methods divides the image into mesh. Mackowiak *et al.* [75] propose CEREALS, which only needs human to label a few automatically selected blocks. Casanova *et al.* [19] propose a novel modification to the deep Q-network to select regions based on predictions and uncertainties.

Another kind of region-based active segmentation method utilizes superpixels to simplify the annotation. Superpixels are non-overlapping local regions generated by grouping similar pixels together. Object boundaries can be preserved among superpixels and pixels within each superpixel commonly share the same class. In this way, annotators only need to classify the selected superpixels instead of annotating each pixel on the image. Among those methods, [94] select samples according to viewpoint entropy and superpixel uncertainty. Golestaneh and Kitani [44] enforce pixel-wise self-consistency computed on outputs of segmentation network between each image and its transformation. Cai *et al.* [17] propose a class-balanced score function to further boost the performance of the superpixel-based approach by favoring the selection of informative samples from under-represented object categories. Another kind of works [92] design a labeling system which assists workers to label certain superpixels on the selected images.

3.4 Video Recognition

Compared with image recognition, video recognition needs to consider both spatial cues and temporal correlations. Active learning for video recognition also shows great potential to save the labeling cost.

Vondrick and Ramanan [102] propose an AL framework for object motion track prediction. The proposed framework selects frames with the largest difference in the predicted track of an object, if the frame is labeled. A large difference is generally caused by distractions like occlusions or cluttered backgrounds. This approach automatically labels tracks of stationary objects and leaves visually ambiguous tracks for manual annotation. Hossain *et al.* [51] study human activity recognition and propose a deep and active learning enabled model (DeActive) that adopts a simple k-means clustering AL approach. DeActive clusters features and selects the most informative samples according to a density-weighted heuristic. Samples found within the most dense regions are generally more representative.

Active learning has also been employed by person re-identification task, too. Person re-id matches query person images or tracklets across videos recorded by different cameras. An increasing demand for the accurate identification

of persons from varying camera angles and times has attracted attention to person re-id. One fundamental challenge faced by person re-id is the difficulty of data annotation. Human annotators need to identify the same person across myriad non-overlapping cameras. This procedure is also easily affected by factors like varied camera viewpoints, lighting change, and similar-appearance imposters, making traditional supervised learning hard to generalize to real applications. Many works have thus proposed AL frameworks for person re-id. Those methods have achieved comparable performance to supervised learning while maintaining a substantially lower annotation cost.

Early AL approaches for person re-id maintains a gallery of labeled images for each person, and matches unlabeled images to those galleries. Wang *et al.* [103] follow this paradigm and propose a Highly Efficient Regression (HER) model for scalable person re-id. HER$^+$, an incremental version of the batch-based HER, is also introduced. A trichotomous joint exploration-exploitation active sampling criteria is applied. Three components of this criteria are (1) appearance diversity exploration that values unlabeled samples farthest away from labeled data, (2) matching uncertainty exploitation that values samples that do not closely resemble any galleries, and (3) ranking uncertainty exploitation that values samples resembling multiple galleries. These criteria balances uncertainty and diversity to select a representative subset of data.

Above methods for person re-id require an annotated gallery for each person, hence are still costly to some extent. Some other active learning methods do not require a labeled dataset for initialization, thus is gaining more popularity. Liu *et al.* [72] propose an active learning framework with pairwise constraint (EALPC). EALPC enforces similar samples closer to each other, thereby facilitates the selection of a more representative set. Liu *et al.* [71] further introduce another active learning method with pairwise diversity maximization (EAL-PDM). Unlike EALPC, EAL-PDM selects the most diverse and informative pairs instead of instances for annotation. Pairwise uncertainty estimation and pairwise diversity maximization are used during the selection procedure.

Wang *et al.* [106] also select pairs of data instead of instances to build AL algorithms. Specifically, they chose tracklets for annotation, which are successive bounding boxes of a tracked person. Their AL algorithm selects the most confident tracklets as true positives. This criterion is supported by their experimental results that, true positive matches are significantly more informative than other relations. Once tracklet pairs are chosen, an oracle is involved to confirm their relation and merge corresponding tracklet pairs. In addition, a view-aware approach is adopted to quickly identify true positives and filter false positives to enlighten the workload.

Liu *et al.* [73] leverage Reinforcement Learning (RL) in AL framework and introduce a Deep Reinforcement Active Learning (DRAL) method. In DRAL, the agent for AL sampling and the CNN for re-id are optimized independently. The agent has an AL policy that is updated iteratively based on hard triplet

loss computed from binary pairwise relation produced by the oracle. As the AL policy is refined, the agent is able to select more informative samples that are then annotated and used to incrementally update the CNN.

In addition to the aforementioned approaches, a clustering approach like that used in [51] is also applied for person re-id. Gao *et al.* [41] use a clustering approach to generate a base cluster to provide pseudo labels. Annotations for inter-centroid relations are then used to split and merge clusters to refine the clustering accuracy. The refined cluster then guides the refinement of the re-id CNN. Similarly, Jin *et al.* [55] propose Support Pair Active Learning (SPAL) that also starts with an unsupervised clustering. In SPAL, a dual certainty selection strategy searches for pairs with the highest likelihood of being a false negative or a false positive. After annotation, inter-support pair relations are used to derive Must-Link and Cannot-Link sets that propagate new labels to the rest of the cluster, as well as to update the re-id CNN.

Related works that we have discussed are summarized and categorized in Table 1.

4   Industrial Applications

This section commences by highlighting the importance of DeepAL in real applications. It then discusses about incorporating DeepAL into the industrial-level model production process, namely deep fusion of DeepAL with GUI-based model production software.

4.1   Importance of DeepAL in Industry

For a typical AI model manufacturing process in industry, the initial steps are to assign annotators to label data and construct a training set. An intuitive way is to label all available samples from the unlabeled data pool. This is not sustainable and scalable for industry, since it is prohibitive to label millions of samples.

A more appealing alternative is to form an iterative loop where each step randomly labels a subset of samples to train the model. In practical vision applications, most of the collected datasets exhibit a long-tailed distribution. In other words, many head categories occupy the majority of samples, while lots of tail categories have a small number of samples. The consequence of random sample selection is that most of the selected samples are from several dominant categories and samples in tail categories are scarcely selected and annotated. This largely degrades the performance of the trained model.

DeepAL has the potential to alleviate the long-tailed distribution issue in the raw dataset by selecting the most informative samples, hence significantly reducing labeling costs. Equipped with DeepAL, both the quality of the

Table 1: Taxology of the described DeepAL methods.

	Uncertainty	Representative	Semi supervised	Classification	Detection	Segmentation	Video
[39]	✓			✓	✓	✓	
[9]	✓			✓	✓	✓	
[88]		✓		✓	✓	✓	
[115]	✓			✓	✓	✓	
[96]		✓	✓	✓	✓	✓	
[6]	✓	✓		✓			
[62]	✓	✓		✓			
[105]	✓		✓	✓			
[95]	✓		✓	✓			
[70]	✓		✓	✓			
[40]	✓		✓	✓			
[61]	✓		✓	✓			
[59]	✓				✓		
[85]	✓				✓		
[27]	✓				✓		
[16]	✓				✓		
[1]		✓			✓		
[117]	✓		✓		✓		
[22]	✓				✓		
[116]	✓	✓	✓		✓		
[110]	✓	✓			✓		
[45]	✓		✓			✓	
[111]	✓	✓				✓	
[79]	✓	✓				✓	
[108]	✓	✓	✓			✓	
[1]	✓	✓				✓	
[75]	✓		✓			✓	
[19]	✓		✓			✓	
[94]	✓		✓			✓	
[44]	✓		✓			✓	
[17]	✓	✓	✓			✓	
[92]	✓	✓	✓			✓	
[72]		✓					✓
[51]	✓	✓					✓
[73]	✓						✓
[71]	✓	✓					✓
[41]	✓	✓	✓				✓
[55]	✓		✓				✓

training set and model performance can be enhanced by the iterative annotation and training loop. Moreover, a closed loop of the AutoML systems can be established by integrating DeepAL and incremental model enhancement algorithms to greatly expand the model production efficiency. One possible solution to easily use DeepAL in the industry is to cope with DeepAL with GUI-based model production software.

4.2 Coping DeepAL with GUI-based Model Production Software

This part starts by discussing the necessity of building GUI-based model production software. GUI-based AI model production software mostly supports the full cycle of the AI model development, including data preparation, model design, training, and deployment. As active learning is closely related to data preparation, data annotation, and model training, it can be easily incorporated into those systems to decrease the cost of data annotation and accelerate the model development. DeepAL thus has shown great potential to be applied in GUI-based production software.

When developing AI-based industrial applications, the process of interaction between customers and AI companies is usually tedious and time-consuming. For example, some customers may not have clear goals for the AI-empowered products they want to develop, and thus the algorithm experts need to interact repetitively with users to bring things to light. In addition, customers are often hesitant to invest heavily in a product until they see how effective it is. This could make for a slow and low-quality development process.

To tackle the challenge and put the initiative in the hands of users, GUI-based model production software has emerged to help public sector and enterprise customers build AI-empowered solutions for their data science problems [8, 24]. Typically, the software supports the full life cycle of a model development process. The implementation details of the software components, such as data preprocessing and model training, are usually hidden. Users can implement their ideas with just clicks of a mouse. The appearance of such easy-to-use software fundamentally advances the democratization of AI technologies and realizes the transformation of model production from fragmentation to scale.

There are many well-known software developed by tech giants or AI startups. We divide these software into two categories, depending on whether they integrate DeepAL component into them. Next, we first introduce some representative non-DeepAL-based model production software, such as SageMaker Studio[1], ModelArts[2], and EasyDL[3], to help readers understand the specific functionalities supported by the modern software. Then we elaborate YMIR[4],

---

[1]https://aws.amazon.com/sagemaker/data-scientist
[2]https://www.huaweicloud.com/intl/en-us/product/modelarts.html
[3]https://ai.baidu.com/easydl/
[4]http://www.viesc.com, https://github.com/IndustryEssentials/ymir

a DeepAL-based software that can greatly facilitate the production of models at scale.

4.2.1 SageMaker Studio

SageMaker Studio, developed by Amazon, provides a single, web-based visual interface for data science team to perform all machine learning steps. As shown in Figure 3, it consists of four main components. During data preparation, users are allowed to prepare and label both structured and unstructured data at large scale. Afterwards, SageMaker not only provides all the tools to build machine learning models, but also relieves users of the need to manage infrastructure, allowing the models to be trained and tuned at scale. After users obtain satisfactory models, SageMaker makes it easy to deploy models and monitors the prediction results.
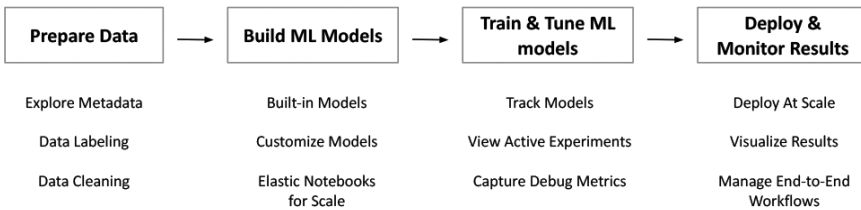
Prepare Data	Build ML Models	Train & Tune ML models	Deploy & Monitor Results
Explore Metadata	Built-in Models	Track Models	Deploy At Scale
Data Labeling	Customize Models	View Active Experiments	Visualize Results
Data Cleaning	Elastic Notebooks for Scale	Capture Debug Metrics	Manage End-to-End Workflows

Figure 3: The main components of Amazon SageMaker Studio.

4.2.2 ModelArts

ModelArts, developed by Huawei, is a one-stop GUI-based AI model production software that enables developers and data scientists of any skill level to quickly build, train, and deploy models anywhere, from the cloud to the edge. In addition to the basic components in machine learning life cycle, ModelArts offers unique features such as automatic labeling, neural architecture search, built-in large pretrained models, large-scale distributed training, and support for diverse training paradigms such as deep learning, reinforcement learning, and federated learning.

4.2.3 EasyDL

EasyDL is an AI development software developed by Baidu. It highlights two unique properties. First, the design of EasyDL is simple and extremely easy to understand, which is suitable for enterprise users who have zero algorithm foundation in AI. Second, EasyDL integrates Baidu Wenxin super large-scale pretraining model and self-developed AutoDL technology, so it can train a high-precision model based on a small amount of data.

4.2.4 YMIR

Although the above software are powerful, they currently do not support DeepAL and users have to randomly select data for annotation, resulting in poor performance in actual use. Fortunately, Huang *et al.* [53] introduces an open source GUI-based and DeepAL-based model production software You Mine In Recursion (YMIR) to enable the swift development of computer vision applications. The following part briefly reviews the pipeline of YMIR, and highlight its differences with previous model production software. More detailed descriptions to YMIR can be found in [53].

YMIR is an implementation of data-centric AI, which puts more emphasis on the quality of data procurement/mining instead of model optimization. It is the first AI algorithm development tool that integrates active learning methods with data version control in the process of developing AI algorithm. In addition, YMIR is completely code-free which lowers the technical requirements of its users to develop algorithms for industry application. In essence, YMIR is designed with the idea of easy development of AI algorithms for everybody while conforming to the industry standard and scale.

YMIR prioritizes the development of datasets in the product and system design. Therefore, the main components of YMIR are aligned with the typical steps in DeepAL, i.e., YMIR iteratively builds datasets and models through DeepAL mining, data labeling, and model training modules. It also borrows ideas from code version control to better track and manage data and models at different stages, and utilizes concepts such as projects to enable fast parallel iteration of multiple task-specific datasets. The included components are implemented through GUI interaction with explicit process guidance. Given a typical industrial example in the development of AI models, YMIR starts with few samples for the initialization of training data set, normally 100 samples for training, the validation will setup at the beginning of DeepAL circle and remain unchanged. The model start with a low accuracy (about 50% of map50 in the validation set) YMIR gradually boost the model performance during the process of DeepAL, it starts with 100 samples for the first iteration and gather 500 samples during each iteration through DeepAL selection until the model performance match the requirement of the developer. In real industrial applications developers suffer from gathering enough training samples to start the AL model training due to the expensive labeling cost. YMIR introduces a new AI model production pipeline for user to start the model development with few samples and gain valuable data during each DeepAL iteration.

The advent of YMIR changes the responsibilities of annotators and algorithm engineers in the model production process, as shown in Figure 4. Traditionally, for a typical model production requirement in the industry, the algorithm engineer first hands over the obtained raw data to the annotators. Because the magnitude of data is usually in the thousands or tens of thousands,
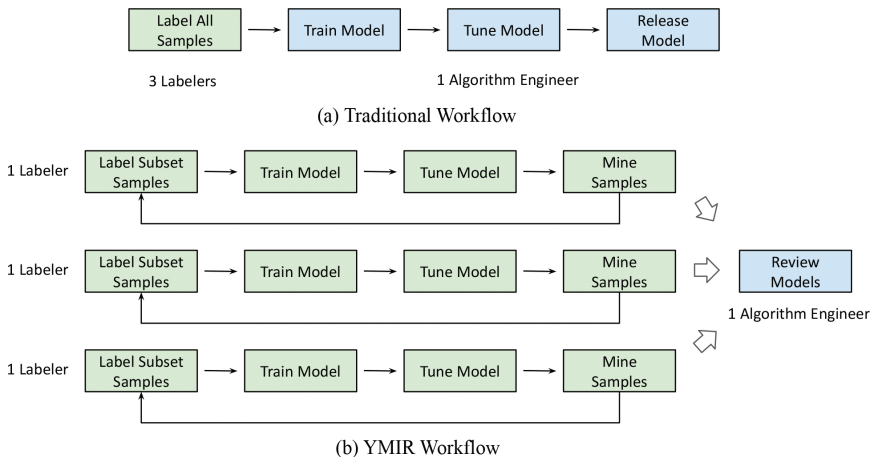
Figure 4: Comparisons between traditional model production workflow and YMIR model production workflow. The green and blue colors stand for the responsibilities for the annotator and the algorithm engineer, respectively. Here we assume there are three labelers and one algorithm engineer. In (a), labelers are assigned to annotate all samples in one task, and the algorithm engineer is responsible for the following steps. While in (b), each labeler is able to produce a model independently. They even can work on different tasks. The algorithm engineer only needs to review the produced models in this workflow.

it takes the annotators lots of time to label all the data. After the annotation is completed, the algorithm engineer is responsible for training, tuning, and releasing the model. In contrast, by using YMIR, the annotators are able to independently produce models thanks to the GUI interface. More importantly, the integration of DeepAL in YMIR forms a cycle that includes data labeling, model training, model tuning, and DeepAL-based sample mining. In such a workflow, the algorithm engineer only needs to review the produced models and can concentrate on designing more advanced model architectures. The benefits of using DeepAL are twofold, namely a remarkable reduction in labeling efforts and a significant increase in model performance with similar manpower.

5   Discussion of Future Directions

The effectiveness of DeepAL in computer vision has been demonstrated through extensive efforts. However, there are still many open problems to be coped with. We present six potential future directions in this section, the first three of which are more related to the training strategy of DeepAL and the latter three are more relevant to DeepAL algorithms.

5.1 Universal Benchmark for DeepAL

After reviewing the literature, we find that the reported performance of the random sampling baseline and some DeepAL methods differ significantly across works. For example, experimental evaluations in [116] show that the method proposed in [59] is inferior to the random selection baseline, which contradicts with results presented in [59].

Moreover, another issue is that different studies apply different settings when conducting experiments, making their performance not directly comparable. For instance, [115] and [59] select a different number of samples in the initial training sets. [117] and [116] use different stop criteria, resulting in a different number of training samples in total. It is obvious that training with more data helps the final performance, which makes the improvements reported in the papers indefinite.

The above problems point to an urgent need to develop a fair performance assessment platform for determining a standard accuracy for the random sampling baseline and evaluating various DeepAL approaches. Fortunately, some researchers have noticed this issue and developed a benchmark named ALBench [34]. In ALBench, different DeepAL algorithms share the same training and testing settings, effectively ensuring equitable comparisons. We believe ALBench will greatly facilitate the research in the community. While effective, ALBench is now only for object detection tasks. A further study may conduct to incorporate more computer vision tasks and the latest DeepAL algorithms.

5.2 Integrating SSL with DeepAL

Self-supervised learning (SSL) becomes very popular in these two years. It obtains supervisory signals from the tremendous unlabeled data itself and aims to learn generalizable and robust representations. The mainstream SSL approaches fall into the categories of contrastive-based and generative-based methods. Some well-known contrastive learning methods include SimCLR [21], MoCo [50], and BYOL [46]. More recently, generative-based methods, e.g., MAE [49], BEiT [7], and MST [67], have become the most successful self-supervised methods in the vision community and they have surpassed the promising performance achieved by contrastive learning methods.

A potential future direction is to combine SSL with DeepAL in order to further reduce the efforts of human labeling. The reason is that after pretraining by SSL methods, the model itself can already generalize well on downstream tasks. At this time, selecting few samples by DeepAL may be enough for enabling the model to achieve competitive performance. However, Chan *et al.* [20] have conducted extensive experiments on integrating SSL with DeepAL and found that the enhancement from DeepAL is marginal. They

further performed ablation study to explore the reason. The results showed that data augmentation could be the source of this phenomena, as augmentation itself has strong capability in improving label efficiency. Bengar *et al.* [10] also attempt to investigate whether SSL and DeepAL can be combined in a complementary way and they find that DeepAL can be beneficial for SSL only when the budget size is high. Therefore, the benefits of integrating SSL with DeepAL remain an open problem that deserves further exploration.

5.3   Incremental Training in DeepAL

Generally, existing methods require training a new model from scratch at each iteration of the DeepAL process. This training paradigm partially offsets the advantages of DeepAL, since retraining large-scale deep learning models is computationally expensive and time-consuming.

In order to tackle this issue and speed up model training, intuitively, we can combine the newly annotated samples with the initial training set and then fine-tune the model from the previous round [91]. However, some researchers have pointed out that such a method introduces bias and degrades the model performance [78]. Specifically, the model tends to fit the new data and overwrites the learned knowledge. ActiveLink [78] delivers an unbiased incremental training approach to alleviate this problem and thus offers a balance between the new and old annotation samples. Since this method is targeting for the link prediction in graphs, future research can be conducted to adapt ActiveLink or develop novel methods for vision tasks.

5.4   Task Independent DeepAL

Most DeepAL methods only target one or part of the subtasks in the computer vision domain. This trend is becoming more pronounced as researchers expect better performance on a certain subtask. For example in the object detection task, MI-AOD [117] argue that there is a gap between instance-level uncertainty and image-level uncertainty due to noisy instances in the background. They propose to apply a multiple instance learning module to force these two levels of uncertainty to be consistent.

The disadvantage of this fragmented situation is that the DeepAL algorithm is overly focused on specific subtasks and lacks generalization ability. Then a natural question is whether a unified DeepAL algorithm framework can be used to solve all mainstream computer vision tasks. This expectation of a unified algorithm is consistent with the current pursuit of unifying computer vision and natural language processing by using the same model architecture.

5.5 DeepAL with Real-world Data

Existing DeepAL methods are generally tested on crafted benchmark datasets, e.g., MS COCO and Pascal VOC in object detection, which are clean and balanced. However, they may fall short in dealing with complicated data distribution in real-world applications, e.g., out-of-distribution (OOD) samples and imbalanced data distribution. In industry, for example, we can get a lot of data from surveillance cameras. By looking closely at the data, we see that pedestrians, motor vehicles, and non-motor vehicles are present most of the time, while other objects to be detected are rarely present. This results in the collected datasets often exhibiting long-tailed distributions. Therefore, it is crucial to make DeepAL methods robust to these factors.

There has been some pioneering work to concentrate on this issue. For example, Contrastive Coding Active Learning (CCAL) [29] point out that existing DeepAL methods often assume that labeled and unlabeled data come from the same class distribution. This assumption, however, does not hold when the unlabeled datasets contain many OOD samples. To address this, CCAL utilizes contrastive learning to extract both semantic and distinctive features and combines them in the query strategy to choose the unlabeled samples with matched categories. Another notable effort is Submodular Information Measures Based Active Learning (SIMILAR) [64], which works to develop a unified DeepAL framework that takes care of many realistic scenarios such as OOD samples, rare classes, redundancy, and imbalanced datasets. Research in this direction is still at an early stage, and we hope that more research will focus on this direction in the future.

5.6 Hierarchical Annotation for DeepAL

Annotating the selected samples is an important step in the DeepAL process. Since DeepAL typically involves only a small fraction of data, the quality of the annotations determines the upper bound on the performance the models can achieve.

In some applications that require domain knowledge to label data, such as medical image analysis, the cost of hiring domain experts can be high. One possible direction to save cost is that, in addition to just selecting samples, DeepAL approaches also provide recommendations of the expertise required by samples, thus building a hierarchical annotation system.

For example, some brain tumor magnetic resonance imaging (MRI) labels can be difficult for junior doctors to judge, but easier for doctors with decades of experience. Moreover, the cost of hiring doctors of different specialties varies widely. If the DeepAL methods can provide the required expertise of the selected samples, we can save costs by only providing difficult samples to experienced doctors and easy samples to junior doctors.

5.7   *Industrial Integration of DeepAL*

Development Operations (DevOps) is a popular technique in the field of software engineering, whose purpose is to enable better communication and collaboration between developers and operations staff, and to deliver applications and services at high velocity and reliability through automated processes. With the advent of the AI era, the concept of MLOps has inevitably emerged. Similar to DevOps, MLOps unifies machine learning systems development and deployment, so as to standardize the model production process and enable continuous delivery of high-performance production models.

Many MLOps systems have been proposed for public use. For example, Amazon SageMaker MLOps[5] helps machine learning engineers easily train, test, troubleshoot, deploy, and govern machine learning models at scale to increase productivity while maintaining model performance in production. Current systems have not integrated with DeepAL components. Since data-centric AI has attracted increasing attention and DeepAL is a key technology for enabling data iteration, integrating DeepAL into the MLOps workflow is an important direction for the future industrial application.

6   Conclusion

In this paper, we present a comprehensive review of DeepAL applied in computer vision. We first give a brief overview of AL and systematically summarize the recent developments in DeepAL. Then we describe the applications of DeepAL in common computer vision tasks, including image classification and recognition, object detection, image segmentation and video recognition, and show how DeepAL is adopted to industrial-level software and systems to accelerate model production. Lastly, we discuss the limitations of DeepAL and propose interesting directions and challenges for the future development of DeepAL.

References

[1]   S. Agarwal, H. Arora, S. Anand, and C. Arora, "Contextual Diversity for Active Learning," in *European Conference on Computer Vision*, Springer, 2020, 137–53.

[2]   H. H. Aghdam, A. Gonzalez-Garcia, J. van de Weijer, and A. M. López, "Active Learning for Deep Detection Neural Networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 3672–80.

---

[5]https://aws.amazon.com/cn/sagemaker/mlops/

[3]   T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-Generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, 2623–31.

[4]   B. An, W. Wu, and H. Han, "Deep Active Learning for Text Classification," in *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, 2018, 1–6.

[5]   N. Andrew, A. Lora, C. Cody, D. Greg, R. Vijay, V. Joaquin, W. Carole-Jean, and Z. Sharon, in *Advances in Neural Information Processing Systems Data-centric AI Workshop*, 2021, https://datacentricai.org/neurips21/.

[6]   J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds," in *arXiv preprint arXiv:1906.03671*, 2019.

[7]   H. Bao, L. Dong, and F. Wei, "Beit: Bert Pre-Training of Image Transformers," in *arXiv preprint arXiv:2106.08254*, 2021.

[8]   D. Baylor, E. Breck, H.-T. Cheng, N. Fiedel, C. Y. Foo, Z. Haque, S. Haykal, M. Ispir, V. Jain, L. Koc, *et al.*, "Tfx: A Tensorflow-based Production-Scale Machine Learning Platform," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, 1387–95.

[9]   W. H. Beluch, T. Genewein, A. Nürnberger, and J. M. Köhler, "The Power of Ensembles for Active Learning in Image Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, 9368–77.

[10] J. Z. Bengar, J. van de Weijer, B. Twardowski, and B. Raducanu, "Reducing Label Effort: Self-Supervised Meets Active Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 1631–9.

[11] S. D. Bhattacharjee, A. Talukder, and B. V. Balantrapu, "Active Learning Based News Veracity Detection with Feature Weighting and Deepshallow Fusion," in *2017 IEEE International Conference on Big Data*, IEEE, 2017, 556–65.

[12] A. Bietti, "Active Learning for Object Detection on Satellite Images," in *Technical report, Technical report, Caltech*, Citeseer, 2012.

[13] M. Bilgic and L. Getoor, "Link-Based Active Learning," in *Advances in Neural Information Processing Systems Workshop on Analyzing Networks and Learning with Graphs*, Vol. 4, 2009.

[14] E. Bıyık, K. Wang, N. Anari, and D. Sadigh, "Batch Active Learning Using Determinantal Point Processes," *arXiv preprint arXiv:1906.07975*, 2019.

[15] E. Bochinski, G. Bacha, V. Eiselein, T. J. Walles, J. C. Nejstgaard, and T. Sikora, "Deep Active Learning for in situ Plankton Classification," in *International Conference on Pattern Recognition*, Springer, 2018, 5–15.

[16] C.-A. Brust, C. Käding, and J. Denzler, "Active Learning for Deep Object Detection," in *International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019, 181–90.

[17] L. Cai, X. Xu, J. H. Liew, and C. S. Foo, "Revisiting Superpixels for Active Learning in Semantic Segmentation with Realistic Annotation Costs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 10988–97.

[18] R. Caramalau, B. Bhattarai, and T.-K. Kim, "Sequential Graph Convolutional Network for Active Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 9583–92.

[19] A. Casanova, P. O. Pinheiro, N. Rostamzadeh, and C. J. Pal, "Reinforced Active Learning for Image Segmentation," in *International Conference on Learning Representations*, 2020.

[20] Y.-C. Chan, M. Li, and S. Oymak, "On the Marginal Benefit of Active Learning: Does Self-supervision Eat its Cake?" In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2021, 3455–9.

[21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," in *International Conference on Machine Learning*, PMLR, 2020, 1597–607.

[22] J. Choi, I. Elezi, H.-J. Lee, C. Farabet, and J. M. Alvarez, "Active Learning for Deep Object Detection via Probabilistic Modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 10264–73.

[23] G. Citovsky, G. DeSalvo, C. Gentile, L. Karydas, A. Rajagopalan, A. Rostamizadeh, and S. Kumar, "Batch Active Learning at Scale," *Advances in Neural Information Processing Systems*, 34, 2021, 11933–44.

[24] P. Das, N. Ivkin, T. Bansal, L. Rouesnel, P. Gautier, Z. Karnin, L. Dirac, L. Ramakrishnan, A. Perunicic, I. Shcherbatyi, *et al.*, "Amazon SageMaker Autopilot: a White Box AutoML Solution at Scale," in *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, 2020, 1–7.

[25] C. Deng, Y. Xue, X. Liu, C. Li, and D. Tao, "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), 2018, 1741–54.

[26]   S. V. Desai and V. N. Balasubramanian, "Towards Fine-Grained Sampling for Active Learning in Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, 924–5.

[27]   S. V. Desai, A. L. Chandra, W. Guo, S. Ninomiya, and V. N. Balasubramanian, "An Adaptive Supervision Framework for Active Learning in Object Detection," in *British Machine Vision Conference*, 2019.

[28]   B. Du, Q. Qi, H. Zheng, Y. Huang, and X. Ding, "Breast Cancer Histopathological Image Classification via Deep Active Learning and Confidence Boosting," in *International Conference on Artificial Neural Networks*, Springer, 2018, 109–16.

[29]   P. Du, S. Zhao, H. Chen, S. Chai, H. Chen, and C. Li, "Contrastive Coding for Active Learning Under Class Distribution Mismatch," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, 8927–36.

[30]   M. Ducoffe and F. Precioso, "Adversarial Active Learning for Deep Networks: a Margin Based Approach," *arXiv preprint arXiv:1802.09841*, 2018.

[31]   E. Elhamifar, G. Sapiro, A. Yang, and S. S. Sasrty, "A Convex Optimization Framework for Active Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2013, 209–16.

[32]   S. Falkner, A. Klein, and F. Hutter, "BOHB: Robust and Efficient Hyperparameter Optimization at Scale," in *International Conference on Machine Learning*, PMLR, 2018, 1437–46.

[33]   D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, "Deep Active Learning for Efficient Training of a Lidar 3d Object Detector," in *2019 IEEE Intelligent Vehicles Symposium*, IEEE, 2019, 667–74.

[34]   Z. Feng, S. Zhang, R. Takezoe, W. Hu, M. Chandraker, L.-J. Li, V. K. Narayanan, and X. Wang, "ALBench: A Framework for Evaluating Active Learning in Object Detection."

[35]   C. Finn, P. Abbeel, and S. Levine, "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks," in *International Conference on Machine Learning*, PMLR, 2017, 1126–35.

[36]   C. Finn, A. Rajeswaran, S. Kakade, and S. Levine, "Online Meta-Learning," in *International Conference on Machine Learning*, PMLR, 2019, 1920–30.

[37]   J. Folmsbee, X. Liu, M. Brandwein-Weber, and S. Doyle, "Active Deep Learning: Improved Training Efficiency of Convolutional Neural Networks for Tissue Classification in Oral Cavity Cancer," in *International Symposium on Biomedical Imaging*, IEEE, 2018, 770–3.

[38]   A. Freytag, E. Rodner, and J. Denzler, "Selecting Influential Examples: Active Learning with Expected Model Output Changes," in *European Conference on Computer Vision*, Springer, 2014, 562–77.

[39] Y. Gal, R. Islam, and Z. Ghahramani, "Deep Bayesian Active Learning with Image Data," in *International Conference on Machine Learning*, PMLR, 2017, 1183–92.

[40] M. Gao, Z. Zhang, G. Yu, S. Ö. Arık, L. S. Davis, and T. Pfister, "Consistency-Based Semi-Supervised Active Learning: Towards Minimizing Labeling Cost," in *European Conference on Computer Vision*, Springer, 2020, 510–26.

[41] W. Gao and M. Li, "Unsupervised Clustering Active Learning for Person Re-identification," *arXiv preprint arXiv:2112.13308*, 2021.

[42] Y. Geifman and R. El-Yaniv, "Deep Active Learning Over the Long Tail," *arXiv preprint arXiv:1711.00941*, 2017.

[43] D. Gissin and S. Shalev-Shwartz, "Discriminative Active Learning," *arXiv preprint arXiv:1907.06347*, 2019.

[44] S. A. Golestaneh and K. Kitani, "Importance of Self-Consistency in Active Learning for Semantic Segmentation," in *British Machine Vision Conference*, 2020.

[45] M. Gorriz, A. Carlier, E. Faure, and X. Giro-i-Nieto, "Cost-Effective Active Learning for Melanoma Segmentation," *arXiv preprint arXiv:1711.09168*, 2017.

[46] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, *et al.*, "Bootstrap your Own Latent-a New Approach to Self-Supervised Learning," *Advances in Neural Information Processing Systems*, 33, 2020, 21271–84.

[47] Y. Guo, "Active Instance Sampling via Matrix Partition," in *Advances in Neural Information Processing Systems*, Vol. 23, 2010.

[48] M. Hasan and A. K. Roy-Chowdhury, "Context Aware Active Learning of Activity Recognition Models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2015, 4543–51.

[49] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked Autoencoders are Scalable Vision Learners," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 16000–9.

[50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 9729–38.

[51] H. S. Hossain, M. A. Al Haiz Khan, and N. Roy, "DeActive: Scaling Activity Recognition with Active Deep Learning," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2), 2018, 1–23.

[52] N. Houlsby, F. Huszár, Z. Ghahramani, and M. Lengyel, "Bayesian Active Learning for Classification and Preference Learning," *arXiv preprint arXiv:1112.5745*, 2011.

[53] P. X. Huang, W. Hu, W. Brendel, M. Chandraker, L.-J. Li, and X. Wang, "YMIR: A Rapid Data-Centric Development Platform for Vision Applications," *arXiv preprint arXiv:2111.10046*, 2021.

[54] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Label Propagation for Deep Semi-Supervised Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 5070–9.

[55] D. Jin and M. Li, "Towards Fewer Labels: Support Pair Active Learning for Person Re-identification," *arXiv preprint arXiv:2204.10008*, 2022.

[56] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-Class Active Learning for Image Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2009, 2372–9.

[57] C. Käding, E. Rodner, A. Freytag, and J. Denzler, "Active and Continuous Exploration with Deep Neural Networks and Expected Model Output Changes," *arXiv preprint arXiv:1612.06129*, 2016.

[58] M. Kampffmeyer, A.-B. Salberg, and R. Jenssen, "Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2016, 1–9.

[59] C.-C. Kao, T.-Y. Lee, P. Sen, and M.-Y. Liu, "Localization-Aware Active Learning for Object Detection," in *Asian Conference on Computer Vision*, Springer, 2018, 506–22.

[60] K. Kim, D. Park, K. I. Kim, and S. Y. Chun, "Task-Aware Variational Adversarial Active Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 8166–75.

[61] Y.-Y. Kim, K. Song, J. Jang, and I.-C. Moon, "LADA: Look-Ahead Data Acquisition via Augmentation for Deep Active Learning," *Advances in Neural Information Processing Systems*, 34, 2021, 22919–30.

[62] A. Kirsch, J. Van Amersfoort, and Y. Gal, "Batchbald: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning," in *Advances in Neural Information Processing Systems*, Vol. 32, 2019.

[63] P. W. Koh and P. Liang, "Understanding Black-Box Predictions via Influence Functions," in *International Conference on Machine Learning*, PMLR, 2017, 1885–94.

[64] S. Kothawade, N. Beck, K. Killamsetty, and R. Iyer, "Similar: Submodular Information Measures Based Active Learning in Realistic Scenarios," in *Advances in Neural Information Processing Systems*, Vol. 34, 2021, 18685–97.

[65] D. D. Lewis and J. Catlett, "Heterogeneous Uncertainty Sampling for Supervised Learning," in *Machine Learning Proceedings 1994*, Elsevier, 1994, 148–56.

[66] X. Li and Y. Guo, "Multi-level Adaptive Active Learning for Scene Classification," in *European Conference on Computer Vision*, Springer, 2014, 234–49.

[67] Z. Li, Z. Chen, F. Yang, W. Li, Y. Zhu, C. Zhao, R. Deng, L. Wu, R. Zhao, M. Tang, *et al.*, "Mst: Masked Self-supervised Transformer for Visual Representation," *Advances in Neural Information Processing Systems*, 34, 2021, 13165–76.

[68] J. Lin, L. Zhao, S. Li, R. Ward, and Z. J. Wang, "Active-Learning-Incorporated Deep Transfer Learning for Hyperspectral Image Classification," 11(11), 2018, 4048–62.

[69] C. Liu, B. Zoph, M. Neumann, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy, "Progressive Neural Architecture Search," in *European Conference on Computer Vision*, 2018, 19–34.

[70] P. Liu, H. Zhang, and K. B. Eom, "Active Deep Learning for Classification of Hyperspectral Images," *IEEE Booktitle of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(2), 2016, 712–24.

[71] W. Liu, X. Chang, L. Chen, D. Phung, X. Zhang, Y. Yang, and A. G. Hauptmann, "Pair-Based Uncertainty and Diversity Promoting Early Active Learning for Person Re-identification," *ACM Transactions on Intelligent Systems and Technology*, 11(2), 2020, 1–15.

[72] W. Liu, X. Chang, L. Chen, and Y. Yang, "Early Active Learning with Pairwise Constraint for Person Re-identification," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2017, 103–18.

[73] Z. Liu, J. Wang, S. Gong, H. Lu, and D. Tao, "Deep Reinforcement Active Learning for Human-in-the-loop Person Re-identification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 6122–31.

[74] O. Mac Aodha, N. D. Campbell, J. Kautz, and G. J. Brostow, "Hierarchical Subquery Evaluation for Active Learning on a Graph," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2014, 564–71.

[75] R. Mackowiak, P. Lenz, O. Ghori, F. Diego, O. Lange, and C. Rother, "CEREALS-Cost-Effective REgion-based Active Learning for Semantic Segmentation," in *British Machine Vision Conference*, 2018.

[76] A. K. McCallumzy and K. Nigamy, "Employing EM and pool-based Active Learning for Text Classification," in *International Conference on Machine Learning*, Citeseer, 1998, 359–67.

[77] H. T. Nguyen and A. Smeulders, "Active Learning Using Pre-clustering," in *International Conference on Machine Learning*, 2004, 79.

[78] N. Ostapuk, J. Yang, and P. Cudré-Mauroux, "Activelink: Deep Active Learning for Link Prediction in Knowledge Graphs," in *The World Wide Web Conference*, 2019, 1398–408.

[79] F. Ozdemir, Z. Peng, C. Tanner, P. Fuernstahl, and O. Goksel, "Active Learning for Segmentation by Optimizing Content Information for Maximal Entropy," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Springer, 2018, 183–91.

[80] A. Pardo, M. Xu, A. Thabet, P. Arbeláez, and B. Ghanem, "BAOD: Budget-aware Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 1247–56.

[81] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of Deep Active Learning," *ACM Computing Surveys*, 54(9), 2021, 1–40.

[82] D. Roth and K. Small, "Margin-based Active Learning for Structured Output Spaces," in *European Conference on Machine Learning*, Springer, 2006, 413–24.

[83] N. Roy and A. McCallum, "Toward Optimal Active Learning Through Monte Carlo Estimation of Error Reduction," in *International Conference on Machine Learning, Williamstown*, Vol. 2, 2001, 441–8.

[84] S. Roy, V. P. Namboodiri, and A. Biswas, "Active Learning with Version Spaces for Object Detection," *arXiv preprint arXiv:1611.07285*, 2016.

[85] S. Roy, A. Unmesh, and V. P. Namboodiri, "Deep Active Learning for Object Detection.," in *British Machine Vision Conference*, 2018, 91.

[86] Y. Saquil, K. I. Kim, and P. Hall, "Ranking Cgans: Subjective Control Over Semantic Image Attributes," *arXiv preprint arXiv:1804.04082*, 2018.

[87] G. Sayantan, P. Kien, and K. Kadambari, "Classification of ECG Beats Using Deep Belief Network and Active Learning," *Medical and Biological Engineering and Computing*, 56(10), 2018, 1887–98.

[88] O. Sener and S. Savarese, "Active Learning for Convolutional Neural Networks: A Core-Set Approach," in *International Conference on Learning Representations*, 2018.

[89] B. Settles, M. Craven, and S. Ray, "Multiple-instance Active Learning," in *Advances in Neural Information Processing Systems*, Vol. 20, 2007.

[90] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by Committee," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, 287–94.

[91] Y. Shen, H. Yun, Z. C. Lipton, Y. Kronrod, and A. Anandkumar, "Deep Active Learning for Named Entity Recognition," in *arXiv preprint arXiv:1707.05928*, 2017.

[92]   G. Shin, W. Xie, and S. Albanie, "All You Need are a Few Pixels: Semantic Segmentation with PixelPick," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, October 2021, 1687–97.

[93]   C. Shui, F. Zhou, C. Gagné, and B. Wang, "Deep Active Learning: Unified and Principled Method for Query and Training," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, 1308–18.

[94]   Y. Siddiqui, J. Valentin, and M. Nießner, "Viewal: Active Learning with Viewpoint Entropy for Semantic Segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 9433–43.

[95]   O. Siméoni, M. Budnik, Y. Avrithis, and G. Gravier, "Rethinking Deep Active Learning: Using Unlabeled Data at Model Training," in *International Conference on Pattern Recognition*, IEEE, 2021, 1220–7.

[96]   S. Sinha, S. Ebrahimi, and T. Darrell, "Variational Adversarial Active Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, 5972–81.

[97]   F. Stark, C. Hazırbas, R. Triebel, and D. Cremers, "Captcha Recognition with Active Deep Learning," in *Workshop new challenges in neural computation*, Vol. 2015, Citeseer, 2015, 94.

[98]   K. Suzuki, T. Sunagawa, T. Sasaki, and T. Katoh, "Annotation Cost Reduction of Stream-based Active Learning by Automated Weak Labeling Using a Robot Arm," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2021, 9000–7.

[99]   M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware Neural Architecture Search for Mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 2820–8.

[100] S. Tong and D. Koller, "Support Vector Machine Active Learning with Applications to Text Classification," *Booktitle of Machine Learning Research*, 2(Nov), 2001, 45–66.

[101] T. Tran, T.-T. Do, I. Reid, and G. Carneiro, "Bayesian Generative Active Deep Learning," in *International Conference on Machine Learning*, PMLR, 2019, 6295–304.

[102] C. Vondrick and D. Ramanan, "Video Annotation and Tracking with Active Learning," in *Advances in Neural Information Processing Systems*, Vol. 24, 2011.

[103] H. Wang, S. Gong, and T. Xiang, "Highly Efficient Regression for Scalable Person Re-identification," in *arXiv preprint arXiv:1612.01341*, 2016.

[104] K. Wang, X. Yan, D. Zhang, L. Zhang, and L. Lin, "Towards Human-Machine Cooperation: Self-supervised Sample Mining for Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, 1605–13.

[105] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective Active Learning for Deep Image Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, 27(12), 2016, 2591–600.

[106] M. Wang, B. Lai, Z. Jin, X. Gong, J. Huang, and X. Hua, "Deep Active Learning for Video-based Person Re-identification," *arXiv preprint arXiv:1812.05785*, 2018.

[107] T. Wang, X. Li, P. Yang, G. Hu, X. Zeng, S. Huang, C.-Z. Xu, and M. Xu, "Boosting Active Learning via Improving Test Performance," *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8), 2022, 8566–74.

[108] W. Wang, Y. Lu, B. Wu, T. Chen, D. Z. Chen, and J. Wu, "Deep Active Self-paced Learning for Accurate Pulmonary Nodule Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2018, 723–31.

[109] J. Wu, V. S. Sheng, J. Zhang, H. Li, T. Dadakova, C. L. Swisher, Z. Cui, and P. Zhao, "Multi-label Active Learning Algorithms for Image Classification: Overview and Future Promise," in *ACM Computing Surveys*, Vol. 53, No. 2, ACM New York, NY, USA, 2020, 1–35.

[110] J. Wu, J. Chen, and D. Huang, "Entropy-based Active Learning for Object Detection with Progressive Diversity Constraint," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 9397–406.

[111] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2017, 399–407.

[112] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class Active Learning by Uncertainty Sampling with Diversity Maximization," *International Booktitle of Computer Vision*, 113(2), 2015, 113–27.

[113] A. Yao, J. Gall, C. Leistner, and L. Van Gool, "Interactive Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, 2012, 3242–9.

[114] C. Yin, B. Qian, S. Cao, X. Li, J. Wei, Q. Zheng, and I. Davidson, "Deep Similarity-based Batch Mode Active Learning with Exploration-Exploitation," in *2017 IEEE International Conference on Data Mining*, IEEE, 2017, 575–84.

[115] D. Yoo and I. S. Kweon, "Learning Loss for Active Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, 93–102.

[116] W. Yu, S. Zhu, T. Yang, and C. Chen, "Consistency-based Active Learn-ing for Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, 3951–60.

[117] T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji, and Q. Ye, "Multiple Instance Active Learning for Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, 5330–9.

[118] X. Zhan, Q. Wang, K.-h. Huang, H. Xiong, D. Dou, and A. B. Chan, "A Comparative Survey of Deep Active Learning," *arXiv preprint arXiv:2203.13450*, 2022.

[119] B. Zhang, L. Li, S. Yang, S. Wang, Z.-J. Zha, and Q. Huang, "State-Relabeling Adversarial Active Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, 8756–65.

[120] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond Empirical Risk Minimization," in *ICLR*, 2018.

[121] Y. Zhang, M. Lease, and B. Wallace, "Active Discriminative Text Representation Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31, No. 1, 2017.

[122] F. Zhdanov, "Diverse Mini-batch Active Learning," in *arXiv preprint arXiv:1901.05954*, 2019.

[123] S. Zhou, Q. Chen, and X. Wang, "Active Deep Networks for Semi-Supervised Sentiment Classification," in *Coling 2010: Posters*, 2010, 1515–23.

[124] J.-J. Zhu and J. Bento, "Generative Adversarial Active Learning," in *arXiv preprint arXiv:1702.07956*, 2017.