Applied Intelligence (2021) 51:5953–5964
https://doi.org/10.1007/s10489-020-02121-4

Dual generative adversarial active learning

Jifeng Guo[1] · Zhiqi Pang[1] · Miaoyuan Bai[1] · Peijiao Xie[1] · Yu Chen[1]

Accepted: 5 December 2020 / Published online: 20 January 2021

Abstract

The purpose of active learning is to significantly reduce the cost of annotation while ensuring the good performance of the model. In this paper, we propose a novel active learning method based on the combination of pool and synthesis named dual generative adversarial active learning (DGAAL), which includes the functions of image generation and representation learning. This method includes two groups of generative adversarial network composed of a generator and two discriminators. One group is used for representation learning, and then this paper performs sampling based on the predicted value of the discriminator. The other group is used for image generation. The purpose is to generate samples which are similar to those obtained from sampling, so that samples with rich information can be fully utilized. In the sampling process, the two groups of network cooperate with each other to enable the generated samples to participate in sampling process, and to enable the discriminator for sampling to co-evolve. Thus, in the later stage of sampling, the problem of insufficient information for selecting samples based on the pool method is alleviated. In this paper, DGAAL is evaluated extensively on three data sets, and the results show that DGAAL not only has certain advantages over the existing methods in terms of model performance but can also further reduces the annotation cost.

Keywords Deep learning · Generative adversarial networks · Image generation · Active learning

1 Introduction

Classification tasks based on deep learning [1, 2] often require large-scale annotation samples [3] for training. However, in practice, the annotation cost of samples may be prohibitive, or they may even be impossible to obtain on a large scale. To remedy this shortcoming, researchers have proposed active learning [4, 5]. The purpose of active learning is to select or generate samples that are most useful for model training from unlabelled data sets and then send the selected samples to an oracle and add them to the training set to reduce the cost of annotation while ensuring the good performance of the task model. Practical application shows that in image classification tasks [6, 7], active learning can effectively reduce the annotation cost of samples while ensuring the model performance.

The variational adversarial active learning (VAAL) [8] is a query-acquiring (pool-based) active learning algorithm with superior performance in image classification and image segmentation. However, active learning methods based on query acquisition have a common problem: the samples taken from the unlabelled pool are sent to the labelled pool after the oracle and do not participate in the subsequent sampling process. Since the number of samples in the unlabelled pool is limited and the algorithm takes samples based on the amount of information, this process will inevitably lead to a decrease in the amount of information contained in the unit sample in the unlabelled pool with an increase in the number of samples, thus reducing the performance improvement rate of the task model.

The generative adversarial networks (GANs) [9], with its strong image processing capability, is outstanding in the fields of image generation [10, 11], style transfer [12, 13] and image recognition [14, 15]. GAN's performance in image generation exceeded that of variational auto-encoder (VAE) [16].

In this paper, an image generation module based on GAN is designed in DGAAL to endow the ability of this model

This work was supported by the Basic Scientific Research Projects of Central Universities 2572018BH07 and the Natural Science Foundation of Heilongjiang Province under Grant LH2019C003.

✉ Yu Chen
1370890813@qq.com

[1] College of information and computer engineering, Northeast Forestry University, Harbin, 150040, China

to generate image, and two groups of adversarial network are designed. The two groups of adversarial network are used to sample and to generate images respectively so that the samples after reconstruction can continue to participate in the sampling process; that is, the method of this paper maintains the number of samples in the candidate pool by adding generated samples to the candidate pool. At the same time, the samples are fully utilized, thereby addressing the shortcomings of the query-acquiring active learning method. In addition, the function of co-evolution of discriminator for sampling is endowed in this paper, which enables the sampling module to update synchronously with the sampling process, so that samples with the most rich information in current stage can be selected for each sampling.

The original contributions of this paper are as follows: 1) This paper proposes an active learning method (DGAAL) that combines query acquisition and query synthesis, which can continuously provide task models with samples that have a rich amount of information to address the shortcomings of pool-based active learning methods. 2) DGAAL proposed in this paper includes two groups of adversarial networks which can not only make full use of samples with rich information, but also make the sampling model evolve with the change of the sample pool. In addition, DGAAL further reduces the cost of labeling. 3) This paper conducts multiple experiments on three public data sets to verify the effectiveness of DGAAL.

2 Related work

2.1 Active learning

The purpose of active learning is to significantly reduce the annotation cost of data sets while ensuring model performance. Currently, mainstream active learning algorithms can be divided into two categories: query-acquiring methods and query-synthesizing methods.

The idea of a query-acquiring method is to use a set sampling strategy to select the samples with the most information from the sample pool. Pool-based methods have been theoretically indicated to be effective and achieve better performance than taking a random sampling of points [17–19]. According to the sampling strategy, pool-based methods can be subdivided into uncertainty-based methods [20–22], representation-based methods [6, 23], and a combination of the two [24, 25].

There are many methods based on uncertainty; for example, uncertainty can be estimated through a probability model in a Bayesian framework, such as Gaussian processes [26, 27] and Bayesian neural networks [28]. At the same time, uncertainty heuristics in non-Bayesian classical active learning methods have been widely studied [29–32], such as the distance to the decision boundary [29] and conditional entropy [30]. Kuo et al. [33] also proposed using an ensemble of models to represent uncertainty, but Melville et al. [34] showed that the use of an ensemble does not always produce high diversity in predictions, which leads to a sampling of redundant instances. Lv et al. [35] proposed a method for uncertain sampling based on the features of the model output. In addition, they also designed an average margin method to control the sampling rate of each category. Learning loss for active learning (LLAL) [36] designed a loss prediction module (LPM) that uses the output of multiple hidden layers of the task model as the input of LPM to predict the loss of the sample. Similarly, Zhao et al. [37] used output of both the hidden layer and the last layer in the segmentation network as the input of the active learning model to evaluate the uncertainty of the sample.

Representation-based methods make sample selections by increasing the diversity in a given batch. For example, the core-set technique [6] has been proven to be very effective in classification methods with a small number of classes, but the performance of this method declines with an increase in the number of classes. VAAL overcomes this limitation by using VAEs, which have been shown to be effective in unsupervised and semisupervised representation learning of high-dimensional data [38]. Similarly, DGAAL solves this problem with coding-decoding-based generators.

Other scholars have proposed an active learning method that combines the above two strategies. For example, Yang et al. [25] used uncertainty and representativeness to select a set of areas for annotation. Task-aware variational adversarial active learning (TA-VAAL) [39] integrates LPM and RankCGAN [40] into VAAL, taking into account the distribution and uncertainty of data. Similarly, state-relabeling adversarial active learning (SRAAL) [41] also considers the distribution and uncertainty of the data. In addition, SRAAL re-marked the status of the samples.

The query-synthesizing method [42–45] is to promote the training of the model by synthesizing samples with abundant information. Among such methods, generative adversarial active learning (GAAL) [44] is of groundbreaking significance. Unlike the query-acquiring approach, GAAL aims to produce new samples useful to models rather than to select the most informative samples from the pool. Ideally, the samples generated by GAAL contain more information than all existing samples. However, GAAL's acquisition function must be easy to calculate and optimize, so the method has certain limitations [46] in application. In subsequent research, Tran et al. proposed Bayesian generative active deep learning (AL w. VAEACGAN) [45], which first selects the samples with the maximum information in the sample pool according to the "information content", and then generates new samples based on the obtained samples.

Compared with the existing query-synthesizing method, this method has advantages in training efficiency and classification performance. Adversarial representation active learning (ARAL) [47] not only uses labeled and unlabeled samples, but also uses generated samples to train the entire model, which further improves the learning ability of the model.

2.2 Generative adversarial networks

GANs [9] are one of the most promising methods in the field of unsupervised learning, and they include a generator and a discriminator. In the field of image generation, the purpose of the generator is to map random noise Z to the image and make the generated image as close to the real image as possible. The purpose of the discriminator is to distinguish the generated image from the real image. The generator and the discriminator are trained alternately until reaching a Nash equilibrium [48]. The objective function of the GAN can be expressed as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1-D(G(z)))], \tag{1}$$

where $p_{data}(x)$ and $p_z(z)$ are the probability distributions of the real data and random noise, respectively; E is the mathematical expectation.

Since the original GAN could not control the type of image generated, Mirza et al. proposed conditional generative adversarial networks (CGANs) [49]. A CGAN introduces condition variables in the generator and discriminator to guide the generation process of the generator. The objective function of a CGAN can be described as:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data}(x)}[\log D(x \mid y)] + E_{z\sim p_z(z)}[\log(1-D(G(z \mid y)))], \tag{2}$$

where $p_{data}(x)$, $p_z(z)$ and E are the same as in formula (1); y is a condition variable in any form.

In subsequent studies, some researchers found that since the original objective function of the GAN was to optimize based on the Jenson-Shannon (JS) divergence [50] between the generated data and the real data, it could easily give rise to the vanishing gradient problem [51] in the initial phase of optimization. Therefore, Arjovsky et al. proposed the Wasserstein GAN (WGAN) [10]. The WGAN's core idea is to introduce the Wasserstein distance into the original objective function. The objective of the WGAN can be expressed as:

$$\min_{G}\max_{f,||f||_L\le 1} E_{x\sim p(x)}[f(x)] - E_{z\sim q(z)}[f(G(z))], \tag{3}$$

where $f(x)$ is a discriminator function subject to Lipschitz constraints [52].

3 Dual generative adversarial active learning

The implementation process of DGAAL proposed in this paper includes model training, sampling and image generation. This section first introduces the model training phase of DGAAL, gives the overall network framework, and then introduces the specific details of the sampling and image generation phases.

3.1 Model training

The DGAAL proposed in this paper includes three subnetworks—the image generator G and discriminators D_1 and D_2—and G, D_1 and D_2 constitute two groups of adversarial networks (G and D_1 undergo adversarial training, as do G and D_2), which are used for representation learning [53, 54] and image generation respectively. The overall structure of this phase is shown in Fig. 1.

G is based on U-net [55], which is composed of an encoder, decoder and skip structure. In addition, residual blocks [56] are added after the encoder to improve the generation capacity of G. D_1 is a multilayer perceptron (MLP), and D_2 is an image classifier. In this paper, the structure of G is divided into G_1 and G_2, where G_1 includes an encoder and G_2 includes residual blocks and a decoder.

3.1.1 Representation learning

In the representation learning phase, G_1 aims to map x_L and x_U to the same feature space [57, 58], extract the feature matrix of the image, and then input the extracted feature matrix to D_1 in an attempt to make D_1 predict that all feature matrices come from the labelled pool. The goal of D_1 is to determine whether the input feature matrix is from x_L and to output the probability that the feature matrix is from x_L. The objective function of this phase is:

$$\begin{aligned}&\min_{G_1}\max_{D_1} V(D_1,G_1)\\ &= E_{x_L\sim p_{\text{data}}(x_L)}\left[\log D_1\left(G_1\left(x_L\right)\right)\right]\\ &\quad + E_{x_U\sim p_{\text{data}}(x_U)}\left[\log\left(1-D_1\left(G_1\left(x_U\right)\right)\right)\right],\end{aligned} \tag{4}$$

where x_L and x_U represent labelled images and unlabelled images, respectively. This phase updates only the G_1 and D_1 parameters. The purpose of representation learning is to enable D_1 to select the most informative samples.

3.1.2 Generation training

In the phase of generation training, G aims to generate images that are close to the real images in an attempt to make D_2 predict that all input images are real. The goal of D_2 is to distinguish between real images and generated images.

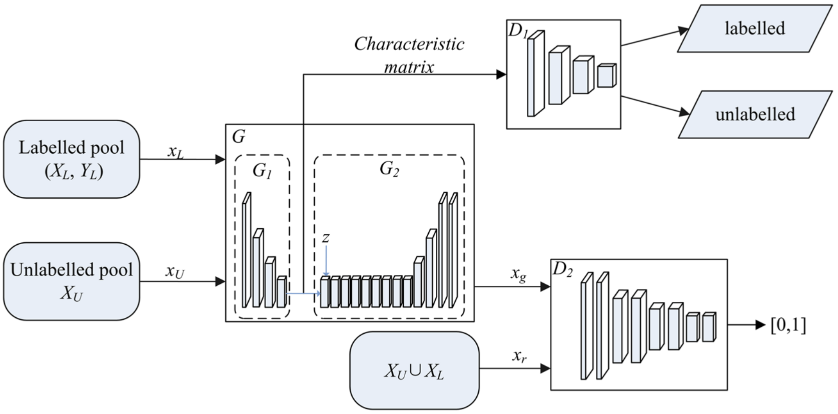


Fig. 1 Overall structure of the training phase. x_L represents an image in the labelled pool (X_L, Y_L), x_U represents an image in the unlabelled pool X_U, x_g represents the generated image, and x_r represents the real image. There are two sets of adversarial networks in the figure, of which G_1 and D_1 form a set of adversarial networks for representation learning. G_1 extracts the feature matrix of images, and D_1 judges whether the feature matrix comes from labeled samples. D_2 and G including G_1 and G_2 form a set of adversarial networks for generating images, G is used to generate images, and D_2 is used to distinguish the authenticity of images

To ensure an appropriate distance between the generated image and the original image, random noise z and pixel-wise mean squared error (pMSE) [59] are introduced into G. The purpose of introducing z is to make the generated image different from the original image. The specific implementation is as follows: convolution kernels with a size of 1×1 are introduced into the head of G_2, as shown in Fig. 1, where the weights of the convolution kernels are random values in [0.95,1.05], and these weights do not participate in the parameter update process to ensure that the features of the generated image and the original image are not exactly the same. The purpose of pMSE is to ensure that the difference between the generated image and the original image is not too large in order to retain enough information and the class label of the original image. The pMSE can be defined as:

$$L_{\text{pMSE}} = \frac{1}{WH}\sum_{x=1}^{W}\sum_{y=1}^{H}\left(I_{x,y} - G({I'}_{x,y})\right)^2, \tag{5}$$

where $I_{x,y}$ and $I'_{x,y}$ are the pixel values at $(x,\ y)$ in the generated image and the real image, respectively; W and H are the height and width of the image, respectively.

The purpose of designing D_2 in this paper is to guide G to generate images that are close to real; that is, D_2 takes real images or generated images as input and then outputs the probability of the images being real to guide G's training process. This paper introduces the Wasserstein distance into the original objective function, and the overall objective function is:

$$\begin{aligned} &\min_{G}\ \max_{f,\|f\|_L\le 1}\ E_{x_r\sim p_{data}(x_r)}\left[f\left(x_r\right)\right] \\ &-E_{x_r\sim p_{data}(x_r)}\left[f\left(G\left(x_r\right)\right)\right] \end{aligned} \tag{6}$$

where x_r represents a real image taken from all the sample pools, $G\left(x_r\right)$ is x_g, and $f(x)$ is the discriminator function, which needs to satisfy the Lipschitz constraints. In this paper, the spectral norm [60] of the matrix is used to make D_2 satisfy the Lipschitz constraints in the global scope. The physical meaning of the spectral norm of the matrix is defined as:

$$\frac{\|f(x+\delta)-f(x)\|_2}{\|\delta\|_2} = \frac{\|W\delta\|_2}{\|\delta\|_2} \le \sigma(W) \tag{7}$$

where $\sigma(W)$ is the spectral norm of the weight matrix, x is the input vector of the current layer and δ is the variation in x. The formula means that the length of any vector after the matrix transform is less than or equal to the product between the vector and the matrix spectral norm.

The overall objective function can be expressed as:

$$L_{\text{total}} = L_{adv} + \lambda L_{\text{pMSE}} \tag{8}$$

where L_{adv} is the adversarial loss shown in (6), L_{pMSE} is the pixel-wise mean squared error shown in (5) and λ is a hyper-parameter for proportional control ($\lambda = 0.1$). In the generation training phase, the parameters of G_1, G_2 and D_2 are updated.

3.1.3 Algorithm process

The training phase algorithm of DGAAL is shown as Alg. 1, where θ_{G_1}, θ_G, θ_{D_1} and θ_{D_2} are the parameters of G_1, G, D_1 and D_2, respectively.

Algorithm 1 Training strategy.

```
Input: Labeled pool $(X_L, Y_L)$, Unlabeled pool $X_U$, Initialized models for $\theta_{G_1}, \theta_G, \theta_{D_1}$ and $\theta_{D_2}$, Hyperparameters $epochs_1, \alpha_1, \alpha_2, \alpha_3, \alpha_4$.
for $e = 1$ to $epochs_1$ do
    sample $(x_L, y_L)$ from $(X_L, Y_L)$
    sample $x_U$ from $X_U$
    Compute $L_{G_1}$ by min$\{Eq.4\}$
    Update $\theta_{G_1}$ by descending stochastic gradients:
    $\theta_{G_1}{}' \leftarrow \theta_{G_1} - \alpha_1 \nabla L_{G_1}$
    Compute $L_{D_1}$ by max$\{Eq.4\}$
    Update $\theta_{D_1}$ by descending stochastic gradients:
    $\theta_{D_1}{}' \leftarrow \theta_{D_1} - \alpha_2 \nabla L_{D_1}$
    Compute $L_G$ by min$\{Eq.8\}$
    Update $\theta_G$ by descending stochastic gradients:
    $\theta_G{}' \leftarrow \theta_G - \alpha_3 \nabla L_G$
    Compute $L_{D_2}$ by max$\{Eq.8\}$
    Update $\theta_{D_2}$ by descending stochastic gradients:
    $\theta_{D_2}{}' \leftarrow \theta_{D_2} - \alpha_4 \nabla L_{D_2}$
end for
return Trained $\theta_{G_1}, \theta_G, \theta_{D_1}$ and $\theta_{D_2}$.
```

3.2 Sampling and image generation

In the sampling phase, DGAAL maintains a candidate pool X_C (X_C is initialized with X_U) and uses a combination of G_1 and D_1 to sample the most informative images x_s in X_C. DGAAL differs from other query-acquiring active learning methods in that DGAAL's sampling model D_1 is updated synchronously with the change of the sample pools. The detailed process is as follows: DGAAL first selects the N samples with the lowest probability of D_1 output from X_C, determines x_s from them, and sends x_s to the oracle; then, it transfers (x_s, y_s) from X_C to (X_L, Y_L). Finally, the concept of co-evolution is introduced; that is, the parameters of D_1 are updated with the updated X_C and (X_L, Y_L). The objective function during the update process is:

$$\begin{aligned} &\min_{G_1} \max_{D_1} V(D_1, G_1) \\ &= E_{x_L \sim p_{\text{data}}(x_L)} \left[\log D_1 \left(G_1 \left(x_L\right)\right)\right] \\ &\quad + E_{x_C \sim p_{\text{data}}(x_C)} \left[\log \left(1 - D_1 \left(G_1 \left(x_C\right)\right)\right)\right], \end{aligned} \tag{9}$$

where x_L and x_C represent images in (x_L, y_L) and X_C, respectively. Co-evolution enables D_1 to monitor the real-time changes in (X_L, Y_L) and X_C so that for each sample of D_1, the most informative sample can be selected at the current stage.

The purpose of image generation is to make use of G to generate new images based on x_s so that x_s can be fully utilized. In the generation phase, this paper first inputs x_s into G, uses G to generate a new image x_g similar to the original image, then sets the label of x_g to be the same as the label of the original image, and finally adds x_g to X_C. In the next round of sampling, x_g still has the opportunity to be sampled, thus indirectly making full use of x_s. The structure of sampling and image generation is shown in Fig. 2.

The sampling and generating phase algorithm of DGAAL is shown as Algorithm. 2, where θ_T is the parameter of T.

Algorithm 2 Sampling and generating strategies.

```
Input: Labeled pool $(X_L, Y_L)$, Unlabeled pool $X_U$, Initialized models for $\theta_T$, Hyperparameters $epochs_2, \alpha_2, \alpha_5, N$.
$X_C \leftarrow X_U$
for $e = 1$ to $epochs_2$ do
    Select samples $(x_s)$ by $D_1$ from $X_C$
    for $i = 1$ to $N$ do
        if $y_s{}^i = null$ then
            $y_s{}^i \leftarrow ORACLE\left(x_s{}^i\right)$
        end if
    end for
    $X_C \leftarrow X_C - x_s$
    $(X_L, Y_L) \leftarrow (X_L, Y_L) \cup (x_s, y_s)$
    Compute $L_{D_1}$ by max$\{Eq.9\}$
    Update $D_1$ by descending stochastic gradients:
    $\theta_{D_1}{}' \leftarrow \theta_{D_1} - \alpha_2 \nabla L_{D_1}$
    Generate samples $(x_g)$ by G$(x_s)$
    $y_g \leftarrow y_s$
    $X_C \leftarrow X_C \cup x_g$
    Train and update $T$:
    $\theta_T{}' \leftarrow \theta_T - \alpha_5 \nabla L_T$
end for
return trained $\theta_T$.
```

4 Experiments and discussion

This section first introduces the experimental setup of this paper, then carries out a simplified model test on DGAAL, and finally makes a comparative analysis between DGAAL and the current baseline methods.

4.1 Experimental setup

Data set: An image classification task was used to evaluate the method of this paper. The data sets used include CIFAR10 [61], CIFAR100 [61] and a self-selected small ImageNet [62] (self-ImageNet).

CIFAR10 consists of 32*32 colour images, for which the training set contains 50,000 samples and the test set contains 10,000 samples, with a total of 10 classes. CIFAR00 is similar to CIFAR10, with 100 classes, each with 500 training images and 100 test images. Self-ImageNet is composed of 200 classes selected from the ImageNet data

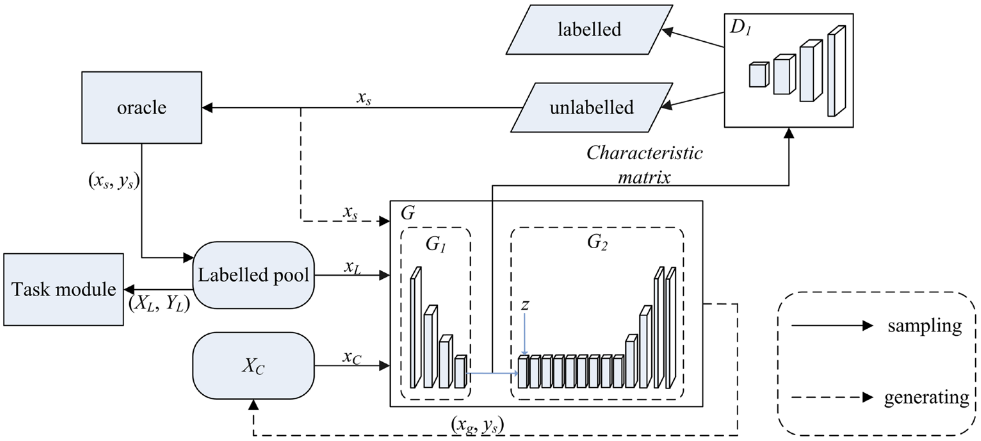


Fig. 2 Flow chart of the sampling and image generation phases. After (x_s, y_s) is moved from X_C to (X_L, Y_L), the task model (T) is trained. This paper first samples based on the predicted value of D_1, and then uses oracle to label the sampled x_s to obtain (x_s, y_s); then this paper transfers (x_s, y_s) from X_C to (X_L, Y_L) and adds $\left(x_g, y_s\right)$ to X_C; finally, this paper updates D_1 with updated X_C and (X_L, Y_L), and uses updated (X_L, Y_L) to train and update T

set, including 500 images for each class in the training set and 100 images for each class in the test set.

Since the samples in the unlabelled pool are derived from a public data set (with labels), this paper does not use the original labels of the samples when initializing the unlabelled pool. When the selected samples need to be sent to the oracle, this paper directly takes the original labels as the annotations of the oracle and then sends the annotated samples to the labelled pool.

Task model: This paper uses the mean accuracy of task model (T) to evaluate the performance of DGAAL in image classification tasks. The architecture used in the task model is VGG16 [63] with the Xavier initialization [64].

Experimental parameters and environment: The two groups of generative adversarial networks and task models in this paper use the Adam optimizer [65], where the first and second momentum terms β_1 and β_2 were set to 0.9 and 0.99, respectively. α_1 and α_2 are set to 0.001, and α_3 , α_4 and α_5 are set to 0.003. Considering the difference in image size and amount of information in different data sets, this paper tested the impact of latent space dimension on performance on each data set, and eventually set the latent space dimensions on CIFAR10, CIFAR100, and self-ImageNet to 32, 32, and 64, respectively. Similarly, we compared the influences of different batch sizes and finally set the batch size to 64.

4.2 Model simplification test

This section verifies the effects of co-evolution and image generation by testing DGAAL and its simplified method. In this paper, 10% of the samples were randomly selected from the original training set as the initial labelled pool, and the remaining 90% of the samples were used as the initial unlabelled pool. During the experiment, 5% of the total number of training sets were added to the labelled pool each time (until the number of samples in the labelled pool reached 40% of the total number of training sets) in order to train T (three random samples were taken under the same conditions to obtain three different initial pools, the experiments were carried out, and finally, the average of the three results was taken as the experimental results for comparative analysis).

The models for comparison include VAAL, VAAL+"co-evolution" and DGAAL. Figure 3 shows the experimental results on different data sets.

It can be concluded from Fig. 3 that when the data ratios are lower than 20%, the effects of the three models are roughly the same; when the data ratios reach 25%, co-evolution starts to work, and the effect of VAAL+"co-evolution" is slightly better than VAAL and similar to DGAAL; when the data ratios reach 40%, the performance of VAAL+ "co-evolution" drops, and it is roughly the same as VAAL. When the data ratios are higher than 30%, DGAAL is significantly better than the other two methods. It can be seen that co-evolution has a greater contribution to the performance improvement when the data ratios are 25%∼35%, and image generation has a greater contribution to the performance improvement when the data ratios are greater than 30%. The combination can make the performance of DGAAL better than the original model. This paper visualizes some of the sampled images when the data ratio is 30% in the third experiment, and the results are shown in Fig. 4.

It can be concluded from Fig. 4 that in the three data sets, the proportion of generated images in the sampled images exceeds 10%, and the generated images tend to be "difficult samples" (more difficult to distinguish than real images), thus verifying the effectiveness of image generation. In addition, image generation can not only improve the utilization of information in x_s, but because x_g has the same label as x_s, it can also reuse the label of x_s to reduce the cost of manual annotation.

Figure 5 shows the manual annotation cost of DGAAL and the other two methods as a function of the number of samples in the labelled pool. Taking the CIFAR10 data set

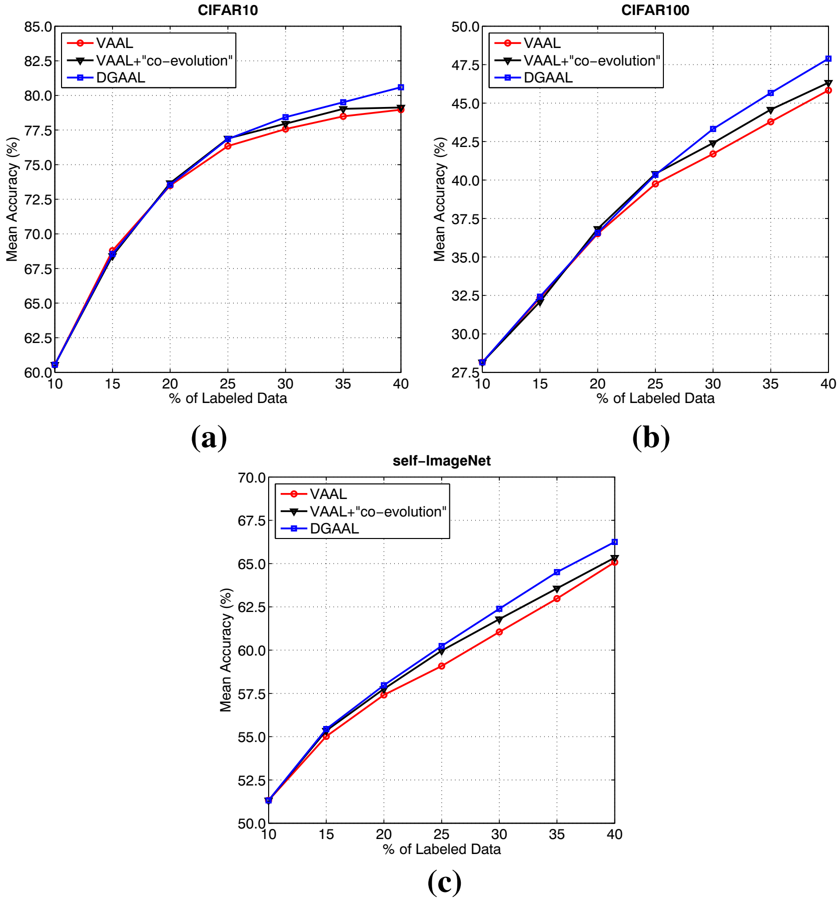


Fig. 3 By using CIFAR10, CIFAR100 and self-ImageNet, we compared the performance of DGAAL, VAAL and VAAL+“co-evolution” on classification tasks

as an example, when the number of labelled pool samples reaches 12,500 (25% of the training set), the number of labelled samples that need to be added is 7500 (15% of the training set), the number of manual annotations required by other methods is 7500 (15% of the training set), and the number of manual annotations required by DGAAL is 7285 (14.57% of the training set), which is approximately 97.1% of other methods; when the number of samples in the labelled pool reaches 15,000 (30% of the training set), the number of manual annotations required by DGAAL is 9561 (19.12% of the training set), which is approximately 95.6% of other methods; when the number of samples in the labelled pool reaches 40%, the number of manual annotations required by DGAAL is 13,429 (26.86% of the training set), and the proportion of manual annotations in the number of samples added to the labelled pool further decreases, to only 89.5% of other methods. DGAAL is also effective on the CIFAR100 and self-ImageNet data sets; that is, when the number of samples in the labelled pool reaches 40%, the number of manual annotations required by DGAAL is 90.3% and 92.5% of other methods, respectively. It can be concluded that DGAAL can further reduce the manual annotation cost of the active learning method through image generation, and it has strong robustness.

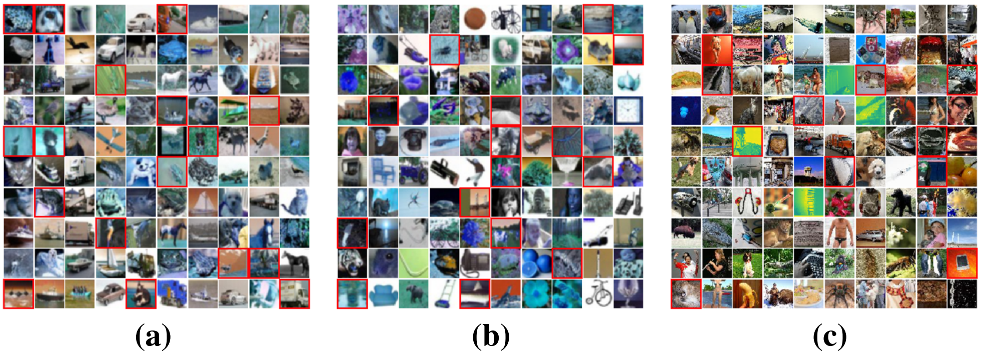
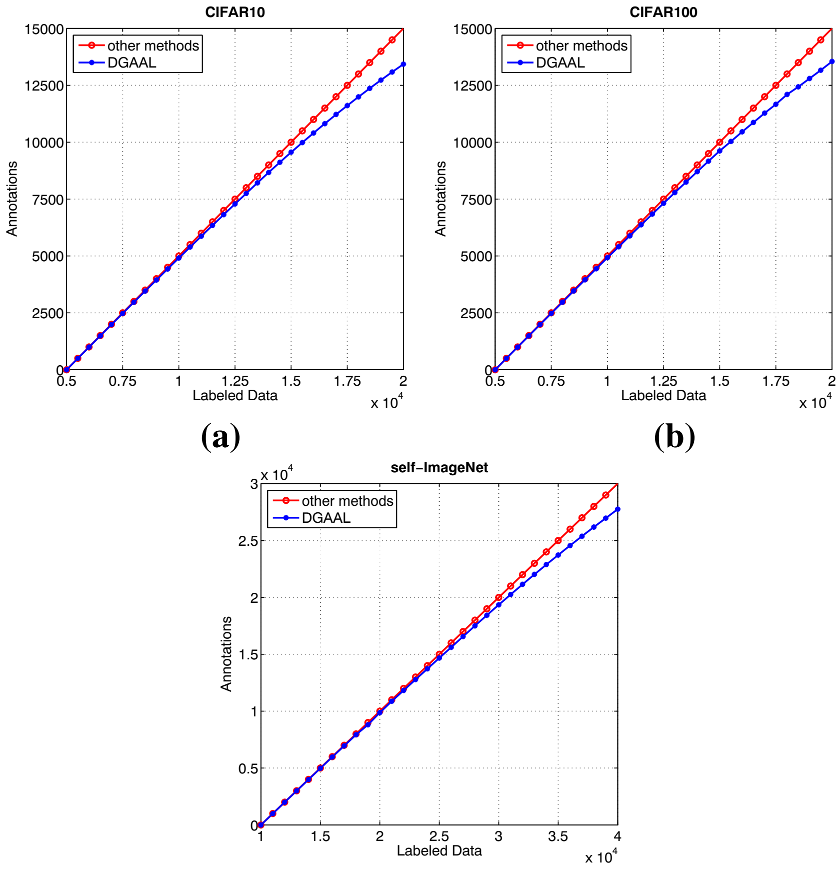


Fig. 4 DGAAL sampling results on CIFAR10, CIFAR100 and self-ImageNet. This paper randomly selects 100 images for visualisation from the sampling results on each data set, in which the image marked with a red frame is the generated image

Fig. 5 Comparison of the annotation costs of different models on CIFAR10, CIFAR100 and self-ImageNet. In this paper, annotation costs are recorded for every 1% increase in the data ratios of the labeled pool



(a) **(b)**

(c)

4.3 Comparative experiment

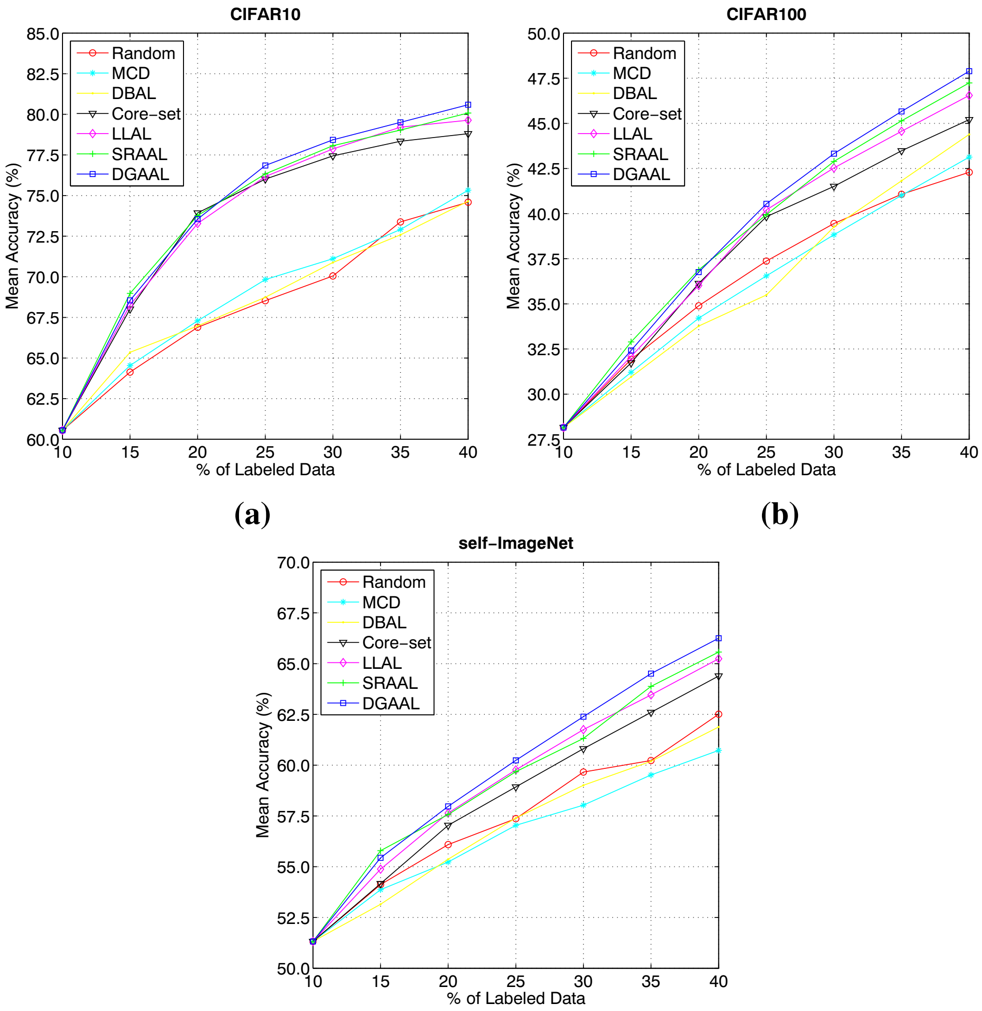
4.3.1 Compared with query-acquiring methods

For the active learning method of query acquisition, the baselines selected in this paper include random sampling (samples are uniformly sampled at random from the unlabelled pool), core-set [6], Monte Carlo dropout (MCD) [66], LLAL [36], SRAAL [41] and deep Bayesian active learning (DBAL) [7].

Figure 6 shows the performance of DGAAL and baselines on the three data sets. On CIFAR10, the mean accuracy using 100% of data is 89.61% while our method achieves 80.59% by only using 40% of it. When data ratio is less than or equal to 20%, DGAAL, LLAL and core-set are in a competitive relationship, they are slightly lower than SRAAL and better than the other three methods. As the sampling continues, the number of samples in candidate pool of other methods continues to decrease. Because DGAAL uses generated samples to supplement the candidate pool, the number of samples in the candidate pool of DGAAL can remain unchanged. Compared with other methods, DGAAL is more likely to obtain samples with rich information. When the data ratio is greater than or equal to 25%, DGAAL is always better than other methods. When the data ratio reaches 25%, 30% and 35% respectively, the mean accuracy of DGAAL is 0.52%, 0.36% and 0.48% higher than that of the optimal method in other methods. When the data ratios reach 40%, DGAAL is 0.51% higher than the mean accuracy of SRAAL which is the most competitive baseline.

Compared with CIFAR10, CIFAR100 has more categories and fewer in-category samples, so CIFAR100 is more challenging than CIFAR10. On CIFAR100, DGAAL achieves mean accuracy of 47.89% by using 40% of the data whereas using the entire dataset yields accuracy of 62.56%. In the early stage of training, due to the small number of labeled samples in each category, and LLAL and core-set only rely on labeled samples for training, their performance is limited. In contrast, DGAAL and SRAAL use labeled samples and unlabeled samples for training, so they have better robustness. When the data ratio is less than or equal to 20%, DGAAL and SRAAL are in a competitive relationship and are superior to the other five methods. When the data ratio is greater than or equal to 25%, DGAAL is always superior to other methods. When the data ratio reaches 25%, 30% and 35% respectively, the mean accuracy of DGAAL is 0.33%, 0.43% and 0.52% higher than that of the

Fig. 6 By using CIFAR10, CIFAR100 and self-ImageNet, we compared the performance on classification tasks of DGAAL, core-set, MCD, DBAL, LLAL, SRAAL and random sampling



(a)

(b)

(c)

optimal method in other methods. When the data ratios reach 40%, DGAAL is 0.65% higher than the mean accuracy of SRAAL which is the most competitive baseline. It can be found that on CIFAR100, with the increase of data ratio, the performance gap between DGAAL and other models gradually increases, which verifies the stability and superiority of DGAAL.

The number of categories contained in self-ImageNet has further increased, making it more challenging than the previous two data sets. DGAAL performed equally well on this data set. Only when the data ratio is 15%, SRAAL is slightly higher than DGAAL, and when the data ratio is greater than or equal to 20%, DGAAL has the best performance among the seven methods. When the data ratio reaches 20%, 25%, 30% and 35% respectively, the mean accuracy of DGAAL is 0.33%, 0.47%, 0.64% and 0.64% higher than the suboptimal method. When the data ratios reach 40%, DGAAL is 0.68% higher than the mean accuracy of SRAAL. In other words, same as CIFAR100, in self-ImageNet, with the increase of data ratio, the performance gap between DGAAL and other models gradually increases, which further verifies the stability of DGAAL.

Based on the above analysis, when the data ratio is less than or equal to 20%, the performance of DGAAL is similar to that of the existing state-of-the-art methods. When the data ratio is greater than or equal to 25%, the performance of DGAAL is better than that of the existing state-of-the-art methods. In addition, the performance of DGAAL is more stable than other methods on two relatively complex data sets CIFAR100 and self-ImageNet. The above are all analyzed under the same data ratios. As shown in Fig. 5, under the same data ratios, the annotation cost of DGAAL is less than other methods, and the cost difference will increase with the increase of the data ratio, so the comprehensive performance of DGAAL is better than baselines.

4.3.2 Compared with the query-synthesizing method

The baseline model selected in this section is AL w. VAEACGAN [45]and ARAL [47]. To ensure the fairness of the experiment, the samples taken by DGAAL are expanded and added to the labelled pool to train T. The sample expansion ratio is real samples : generated samples=1:1; that is, the number of labelled data values in Fig. 7 is twice the number of real samples.

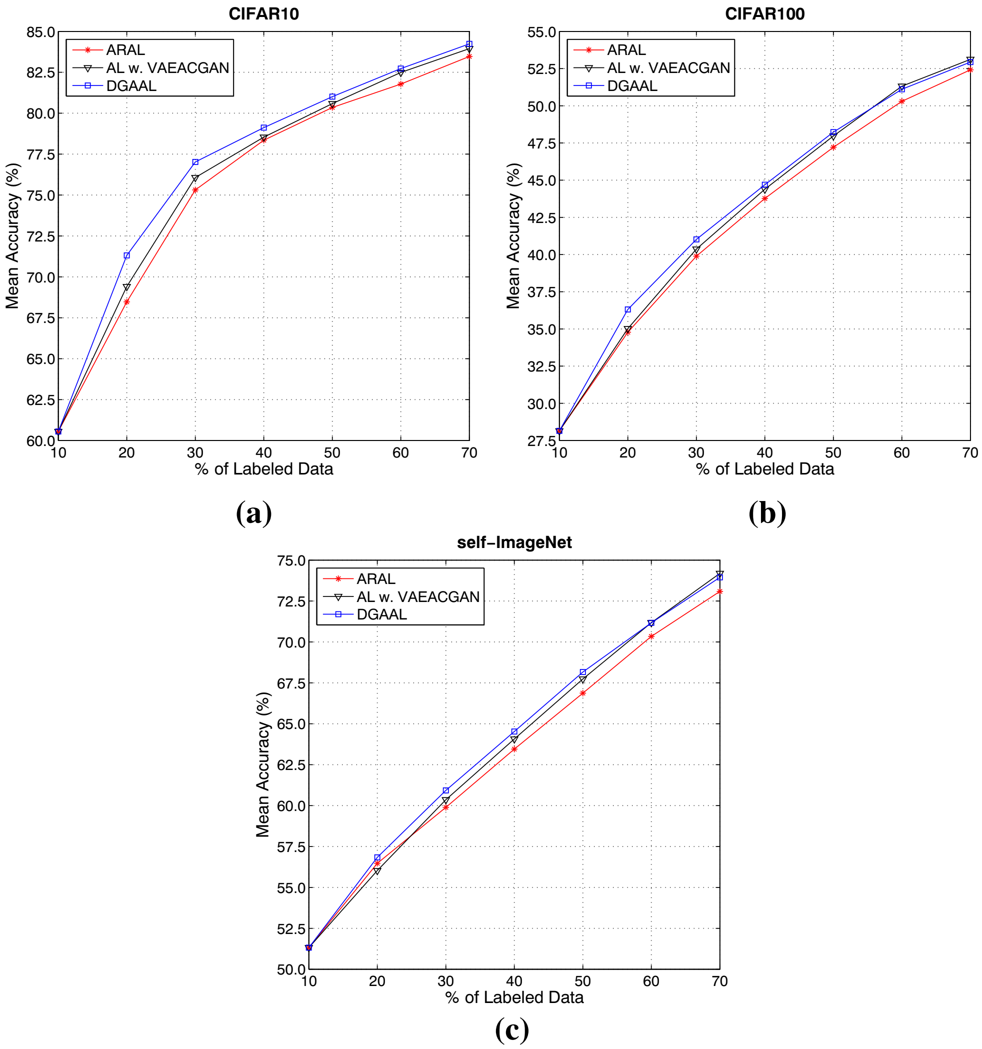
As shown in Fig. 7, on CIFAR10, the overall performance of DGAAL is superior than AL w. VAEACGAN and ARAL, and on CIFAR100 and self-ImageNet, DGAAL is in a competitive relationship with AL w. VAEACGAN and is better than ARAL.

The overall analysis of the performance on the three data sets found that DGAAL is superior than the other two methods when the data ratios are between 20% and 40%. This is because compared with the other two methods, the discriminator D_2 used for image generation in DGAAL uses all images for training and directly guides the generation process of G, which makes the images generated by DGAAL more realistic. In contrast, discriminators of the other two methods participate in the training process of adversarial generation together with the classifier. Due to the low performance of the classifier at the initial stage of training, it is difficult to provide accurate guidance for generator, so the image quality generated by the other two methods is low at the initial stage of training. The image quality directly affects the performance of task model.

Therefore, DGAAL is more suitable for scenarios with a small amount of annotation than the other two methods, that is, DGAAL is closer to active learning main idea.

5 Conclusion

This paper proposes a novel active learning method (DGAAL) and introduces the concepts of image generation



(a) (b)

(c)

Fig. 7 By using CIFAR10, CIFAR100 and self-ImageNet, we compared the performance of DGAAL, ARAL and AL w. VAEACGAN on classification tasks

and co-evolution into the method, which enables DGAAL to continuously provide task models with samples that have a rich amount of information in order to address the shortcomings of query-acquiring active learning methods. In this paper, random noise z is introduced into the generator in the form of convolution to ensure an appropriate distance between the generated image and the original image, so as to increase the diversity of images. In addition, pMSE is also used in this paper to make the generated image maintain the abundant information and the class label of the original image. This paper describes the experimental performance of DGAAL on three data sets (CIFAR10, CIFAR100 and self-Imagenet). The results show that DGAAL can not only further improve the performance of the task model compared to the other methods but can also reduce the annotation cost.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

References

1. Yuan C, Wu Y, Qin X, Qiao S, Pan Y, Huang P, Liu D, Han N (2019) An effective image classification method for shallow densely connected convolution networks through squeezing and splitting techniques. Appl Intell 49(10):3570–3586
2. Lin E, Chen Q, Qi X (2020) Deep reinforcement learning for imbalanced classification. Appl Intell, 1–15
3. Sun C, Shrivastava A, Singh S, Gupta A (2017) Revisiting unreasonable effectiveness of data in deep learning era. In: Proceedings of the IEEE international conference on computer vision, pp 843–852
4. Liu P, Zhang H, Eom KB (2016) Active deep learning for classification of hyperspectral images. IEEE J Sel Top Appl Earth Obs Remote Sens 10(2):712–724
5. Shao W, Sun L, Zhang D (2018) Deep active learning for nucleus classification in pathology images. In: 2018 IEEE 15th international symposium on biomedical imaging, pp 199–202
6. Sener O, Savarese S (2017) Active learning for convolutional neural networks: A core-set approach. arXiv:1708.00489
7. Gal Y, Islam R, Ghahramani Z (2017) Deep bayesian active learning with image data arXiv:1703.02910
8. Sinha S, Ebrahimi S, Darrell T (2019) Variational adversarial active learning. In: Proceedings of the IEEE international conference on computer vision, pp 5972–5981
9. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S et al (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
10. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein gan. arXiv:1701.07875
11. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of wasserstein gans. In: Advances in neural information processing systems, pp 5767–5777
12. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232
13. Choi Y, Choi M, Kim M, Ha JW, Kim S, Choo J (2018) Stargan: unified generative adversarial networks for multi-domain image-to-image translation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8789–8797
14. Tang XL, Du YM, Liu YW, Li JW, Ma YW (2018) Image recognition with conditional deep convolutional generative adversarial networks. J Autom Autom 44(05):855–864
15. Deng J, Cheng S, Xue N, Zhou Y, Zafeiriou S (2018) Uv-gan: adversarial facial uv map completion for pose-invariant face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7093–7102
16. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv:1312.6114
17. Freund Y, Seung HS, Shamir E, Tishby N (1997) Selective sampling using the query by committee algorithm. Mach Learn 28(2-3):133–168
18. Gilad-Bachrach R, Navot A, Tishby N (2006) Query by committee made real. In: Advances in neural information processing systems, pp 443–450
19. Dasgupta S, Hsu D (2008) Hierarchical sampling for active learning. In: Proceedings of the 25th international conference on Machine learning, pp 208–215
20. Beluch WH, Genewein T, Nürnberger A, Köhler JM (2018) The power of ensembles for active learning in image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9368–9377
21. Gorriz M, Carlier A, Faure E, Giro-i-Nieto X (2017) Cost-effective active learning for melanoma segmentation. arXiv:1711.09168
22. Wang K, Zhang D, Li Y, Zhang R, Lin L (2016) Cost-effective active learning for deep image classification. IEEE Trans Circ Syst Video Technol 27(12):2591–2600
23. Dutt Jain S, Grauman K (2016) Active image segmentation propagation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2864–2873
24. Nguyen HT, Smeulders A (2004) Active learning using pre-clustering. In: Proceedings of the twenty-first international conference on machine learning, p 79
25. Yang L, Zhang Y, Chen J, Zhang S, Chen DZ (2017) Suggestive annotation: A deep active learning framework for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, New York, pp 399–407
26. Kapoor A, Grauman K, Urtasun R, Darrell T (2007) Active learning with gaussian processes for object categorization. In: 2007 IEEE 11th international conference on computer vision, pp 1–8
27. Roy N, McCallum A (2001) Toward optimal active learning through monte carlo estimation of error reduction. ICML, 441–448
28. Ebrahimi S, Elhoseiny M, Darrell T, Rohrbach M (2019) Uncertainty-guided continual learning with bayesian neural networks. arXiv:1906.02425
29. Tong S, Koller D (2001) Support vector machine active learning with applications to text classification. J Mach Learn Res 2(Nov):45–66
30. Li X, Guo Y (2013) Adaptive active learning for image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 859–866
31. Brinker K (2003) Incorporating diversity in active learning with support vector machines. In: Proceedings of the 20th international conference on machine learning, pp 59–66
32. Wang Z, Ye J (2015) Querying discriminative and representative samples for batch mode active learning. ACM Trans Knowl Disc Data 9(3):1–23

33. Kuo W, Häne C, Yuh E, Mukherjee P, Malik J (2018) Cost-sensitive active learning for intracranial hemorrhage detection. In: International conference on medical image computing and computer-assisted intervention. Springer, New York, pp 715–723
34. Melville P, Mooney RJ (2004) Diverse ensembles for active learning. In: Proceedings of the twenty-first international conference on Machine learning, p 74
35. Lv X, Duan F, Jiang JJ, Fu X, Gan L (2020) Deep active learning for surface defect detection. Sensors 20(6):1650
36. Yoo D, Kweon IS (2019) Learning loss for active learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 93–102
37. Zhao Z, Yang X, Veeravalli B, Zeng Z (2020) Deeply supervised active learning for finger bones segmentation. arXiv:2005.03225
38. Sohn K, Lee H, Yan X (2015) Learning structured output representation using deep conditional generative models. In: Advances in neural information processing systems, pp 3483–3491
39. Kim K, Park D, Kim KI, Chun SY (2020) Task-aware variational adversarial active learning. arXiv:2002.04709
40. Saquil Y, Kim KI, Hall P (2018) Ranking cgans: Subjective control over semantic image attributes. arXiv:1804.04082
41. Zhang B, Li L, Yang S, Wang S, Zha ZJ, Huang Q (2020) State-relabeling adversarial active learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8756–8765
42. Mahapatra D, Bozorgtabar B, Thiran JP, Reyes M (2018) Efficient active learning for image classification and segmentation using a sample selection and conditional generative adversarial network. In: International conference on medical image computing and computer-assisted intervention. Springer, New York, pp 580–588
43. Mayer C, Timofte R (2020) Adversarial sampling for active learning. In: The IEEE winter conference on applications of computer vision, pp 3071–3079
44. Zhu JJ, Bento J (2017) Generative adversarial active learning. arXiv:1702.07956
45. Tran T, Do TT, Reid I, Carneiro G (2019) Bayesian generative active deep learning. arXiv:1904.11643
46. Gal Y, Islam R, Ghahramani Z (2017) Deep bayesian active learning with image data. arXiv:1703.02910
47. Mottaghi A, Yeung S (2019) Adversarial representation active learning. arXiv:1912.09720
48. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Advances in neural information processing systems, pp 6626–6637
49. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv:1411.1784
50. Grosse I, Bernaola-Galván P, Carpena P, Román-Roldán R, Oliver J, Stanley HE (2002) Analysis of symbolic sequences using the jensen-shannon divergence. Phys Rev E 65(4):041905
51. Jin Q, Luo X, Shi Y, Kita K (2019) Image generation method based on improved condition GAN. In: 2019 6th international conference on systems and informatics, pp 1290–1294
52. Cui S, Jiang Y (2017) Effective lipschitz constraint enforcement for wasserstein GAN training. In: 2017 2nd IEEE international conference on computational intelligence and applications, pp 74–78
53. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
54. Donahue J, Simonyan K (2019) Large scale adversarial representation learning. In: Advances in neural information processing systems, pp 10542–10552
55. Ma K, Shu Z, Bai X, Wang J, Samaras D (2018) Docunet: Document image unwarping via a stacked u-net. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4709
56. Targ S, Almeida D, Lyman K (2016) Resnet in resnet: Generalizing residual architectures. arXiv:1603.08029
57. DeVries T, Taylor GW (2017) Dataset augmentation in feature space. arXiv:1702.05538
58. Yu B, Zhu DH (2009) Combining neural networks and semantic feature space for email classification. Knowl-Based Syst 22(5):376–381
59. Xue W, Mou X, Zhang L, Feng X (2013) Perceptual fidelity aware mean squared error
60. Miyato T, Kataoka T, Koyama M, Yoshida Y (2018) Spectral normalization for generative adversarial networks. arXiv:1802.05957
61. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images
62. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, pp 248–255
63. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556
64. Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics, pp 249–256
65. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv:1412.6980
66. Gal Y, Ghahramani Z (2016) Dropout as a bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning, pp 1050–1059