Sun and Jia
Journal of Engineering and Applied Science (2024) 71:97
https://doi.org/10.1186/s44147-024-00429-3

Journal of Engineering and Applied Science

RESEARCH Open Access

SKZC: self-distillation and k-nearest neighbor-based zero-shot classification

Muyang Sun[1*†] and Haitao Jia[2†]

†MuyangSun and Haitao Jia contributed equally to this study.

*Correspondence: cmoswolf@163.com

[1] University of Electronic Science and Technology of China, Yangyze Delta Region Institute (Huzhou), 819 Xisai Mountain Road, Building B1, 7th Floor, Huzhou 313000, Zhejiang, China
[2] University of Electronic Science and Technology of China, School of Resources and Environment, No. 2006, Xiyuan Avenue, Chengdu 611731, Sichuan, China

Abstract

Zero-shot learning represents a formidable paradigm in machine learning, wherein the crux lies in distilling and generalizing knowledge from observed classes to novel ones. The objective is to identify unfamiliar objects that were not included in the model's training, leveraging learned patterns and knowledge from previously encountered categories. As a crucial subtask of open-world object detection, zero-shot classification can also provide insights and solutions for this field. Despite its potential, current zero-shot classification models often suffer from a performance gap due to limited transfer ability and discriminative capability of learned representations. In pursuit of advancing the subpar state of zero-shot object classification, this paper introduces a novel model for image classification which can be applied to object detection, namely, self-distillation and k-nearest neighbor-based zero-shot classification method. First, we employ a diffusion detector to identify potential objects in images. Then, self-distillation and distance-based classifiers are used for distinguishing unseen objects from seen classes. The k-nearest neighbor-based cluster heads are designed to cluster the unseen objects. Extensive experiments and visualizations were conducted on publicly available datasets on the efficacy of the proposed approach. Precisely, our model demonstrates performance improvement of over 20% compared to contrastive clustering. Moreover, it achieves a precision of 0.910 and a recall of 0.842 on CIFAR-10 datasets, a precision of 0.737, and a recall of 0.688 on CIFAR-100 datasets for the macro average. Compared to a more recent model (SGFR), our model realized improvements of 10.9%, 13.3%, and 7.8% in Sacc, Uacc, and H metrics, respectively. This study aims to introduce fresh ideas into the domain of zero-shot image classification, and it can be applied to open-world object detection tasks. Our code is available at https://www.github.com/CmosWolf1/Code_implementation_for_paper_SKZC.

Keywords: Image classification, Zero-shot, Self-distillation, k-NN, Cluster

Introduction

As a crucial task in computer vision, image classification [1] tasks involve assigning predefined labels or categories to input data based on their characteristic or features. It is also an important subtask within the field of object detection. There is no doubt that enhancements in the performance of classification models can also lead to improvements in the classification abilities of performance of object detection models. Tasks of classification depend on the availability of a large volume of tagged data [2]. Due to

advances in deep learning techniques [3–5], most image classification methods used in the domain of computer vision are supervised learning methods, depending on large extensive volumes of tagged data for training. However, existing datasets are unable to encompass all possible classes, and human society's evolution continually gives rise to fresh classifications [6]. It leads these supervised classification methods to perform unsatisfying when some categories have scarce or even no tagged data [7].

Zero-shot classification also seen as zero-shot learning (ZSL) [8, 9] or zero-shot recognition is suggested to address the problem of lacking data enabling the recognition of objects belonging to unseen categories. It is a sub-field of machine learning that aims to classify objects or instances into unseen classes during training by leveraging the knowledge transfer from related classes for which labeled data is available.

Traditional zero-shot classification can be divided into three main approaches. The first approach utilizes pre-trained word embedding vectors to represent and understand the relationship among different categories. For instance, DeViSE [10] utilizes a pre-trained convolutional neural network (CNN) to project image features and word embedding of labels into a shared space. ConSE [11], on the other hand, merges the k highest-probability image embeddings. The second approach directly incorporates the relationships between classes using either a graph convolutional network (GCN) or a predefined class hierarchy like WordNet [3]. GCNZ [12] and DGPZ [13] employ GCNs to propagate knowledge between seen and unseen classes, while incorporating CNN and word embedding. An alternative method, HZSL [14], projects both image and text embedding into a hyperbolic space that organizes child and parent classes within the hierarchical structure of WordNet [3]. Lastly, some approaches, such as [15–17], depend on human-tagged attributes to model class semantics. These methods consider attribute annotations as informative cues for understanding the characteristics and distinguishing features of various classes. Different from CNN-based methods, vision transformers (ViT) [18] have surfaced as a substitute for convolutional neural networks in the field of visual recognition [18–20]. The emergence of self-distillation [21] has provided new solutions for zero-shot. Self-knowledge distillation [21] seeks to educate a student model by emulating the learning patterns of an already-trained teacher model, which is a pre-trained ViT model. Many zero-shot learning methods, such as [22, 23], utilize self-distillation models to acquire features for unseen categories.

However, these prior approaches suffer from several limitations. First, their focus lies primarily on improving the correspondence between image features extracted from pre-trained CNNs and pre-trained word embedding models like Glove [24]. Moreover, they employ predefined class hierarchies, such as WordNet [3], which confines category modeling to a tree structure, thereby failing to capture the complex inter-class relationships observed in real-world objects. Moreover, relying solely on class hierarchies restricts the classification scope to only those categories included in the hierarchy. Lastly, attribute-based methods lack the ability to generalize to categories lacking seen attributes, thereby limiting their applicability.

Based on the aforementioned observation, we introduce a novel self-distillation and k-nearest neighbor-based model for zero-shot classification problems namely, self-distillation and k-nearest neighbor-based zero-shot classification. When unseen categories are underrepresented or completely absent in datasets, and lack clear semantic

relationships with other seen classes, conventional zero-shot image classification algorithms often struggle to achieve satisfactory classification performance. In contrast, our model effectively addresses this issue. Firstly, we use a diffusion detector [25] to detect potential objects in the image. Secondly, we design a self-distillation and distance-based classifier (SDDC) to classify seen and unseen objects. Lastly, we propose a k-nearest neighbor-based cluster head (KCH) to cluster those different kinds of unseen objects. As shown in Fig. 1, the clustering process is performed using KCH on several unseen objects in a given embedding space. Extensive experiments have demonstrated the efficacy of our model. We conducted tests on four datasets: CIFAR-100, CIFAR-10, ImageNet-10, and STL-10 [26–28]. In cluster performance, we achieved varying degrees of improvement compared to the contrastive clustering [29] method. Moreover, we achieve a precision of 0.910 and a recall of 0.842 on CIFAR-10 datasets, and a precision of 0.737 and a recall of 0.688 on CIFAR-100 datasets for the macro-average. Compared to a more recent model (SGFR), our model realized improvements of 10.9%, 13.3%, and 7.8% in Sacc, Uacc, and H metrics, respectively.

Our main contributions are as follows:

(1) For the first time, we have applied diffusion model to the detection of seen and unseen objects. This implies that the methods in our model can be applied not only to classification tasks but also provide solutions and insights for detection tasks, particularly open-world object detection [6, 30] (OWOD) tasks.
(2) We propose self-distillation and distance-based classifier (SDDC) and the k-nearest neighbor-based cluster head (KCH) to classify seen and unseen objects.
(3) Our model is capable of lifelong learning, meaning it can without the need for human intervention once it is initialized.

Related work

Generative-based ZSL methods

In the domain of zero-shot image classification, leveraging generative adversarial networks (GANs) that are capable of synthesizing highly authentic imagery has emerged as a novel and promising approach [31, 32]. These advanced GAN variants enable the generation of visual feature representations for unseen categories by utilizing the known

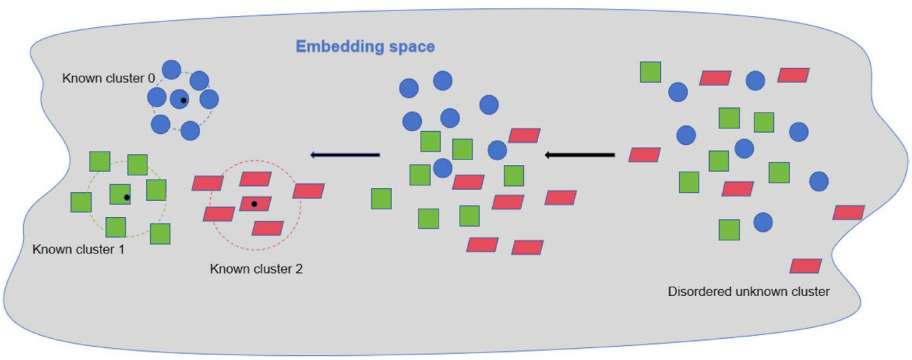


Fig. 1 Clustering process of unseen classes using the KCH

visual data from seen classes coupled with semantic attributes of the target unseen classes. Xian et al. [33] devised an enhanced model incorporating Wasserstein GAN (WGAN) [34], integrating the WGAN's loss function with a classification loss to not only ensure the discriminative nature of the synthetically produced features but also to bolster the stability of the training regimen. Subsequently, numerous researchers have refined the WGAN framework, aiming to address challenges associated with generated samples' quality, diversity, and semantic relevance [30, 35, 36]. Vyas et al. [37] introduced the leveraging of the semantic relationship GAN (LsrGAN), which utilizes a semantic-regularized loss component to facilitate knowledge transfer between classes. To counteract issues related to training instability, certain studies have adopted variational auto-encoder (VAE) known for their robust training characteristics in zero-shot learning tasks [38–40]. Other research efforts have focused on developing a joint embedding space through VAE for multi-modal data integration [41, 42], effectively narrowing the divide between the visual and semantic spectra.

Embedding-based ZSL methods

Embedding-based approaches are designed to create a shared embedding space for images and their corresponding semantic attributes. These approaches can be categorized into three distinct types. The first category concentrates on mastering a conversion from the visual space to semantic space [43–45] which encounters issues such as projection domain shift and the hubness phenomenon. To mitigate these issues, the second type of approach inverts this direction by mapping the semantic information onto the visual domain [46, 47]. The third category aims to reconcile the disparities between visual and semantic domains by jointly mapping both visual and semantic features into an intermediary shared space [48, 49]. This common space is calibrated using bi-directionally aligned knowledge from both visual and semantic representations, addressing the limitations associated with direct mappings and transfer of model parameters. Despite these improvements, embedding-based techniques continue to grapple with challenges such as semantic information loss and a deficiency in representing unseen class features, leading to a prediction bias towards classes that have been observed during training [50].

Methods

Problem definition

Let's assume that the set of categories to which all objects in an open-world belong comprises the set $S^t = \{1,2,3,...,C\} \subset N^+$, where N^+ denotes the set of positive integers, C is the number of all the classes in the open world. Seen and unseen categories can be respectively defined as K^t and U^t. Let's define embedding vector sets set as F^t.

It is evident that K^t, $U^t \subseteq S^t$, and both K^t and U^t are empty at the onset of the task. Moreover, the seen and unseen objects come from the detector. Then, those seen and unseen objects will be added into set K^t or set U^t according to the result of a classifier. Subsequently, we need to cluster these unseen categories. It is worth noting that vector clusters in the embedding set will continue to increase as the task progresses. Therefore, due to the limitations of computational power and cost, we need to put these unseen categories into several embedding sets before clustering (further particulars will

be elaborated in the subsequent subsections). These embedding sets are combined to form set F^t.

Overall architecture

Figure 2 presents the comprehensive structure of our proposed model for zero-shot image classification. We have incorporated a detector into our model for classification tasks and continuously update it to enhance its performance in real-world classification tasks. Additionally, cropping the images detected by the detector allows our model to iterate by itself at a fast pace.

Firstly, we use diffusion model detector [25] as the based detector. Then, we crop the image detected by the detector according to the box predictor. These cropped images are sent into the self-distillation and distance-based classifier (SDDC) to differentiate between categories that have been previously encountered and those that have not. After that, unseen categories will be sent into a k-nearest neighbor-based clustering head (KCH) for clustering. Seen classes will be added to the existing seen cluster. Lastly, we update the boxes predictor module so that the detector can recognize the newly added classes. Additionally, we will integrate the already clustered unseen clusters into the embedding vector set to accomplish the transformation from unseen classes to seen classes. As time progresses, the number of seen clusters will increase, allowing the model to recognize an ever-growing of classes.

Self-distillation and distance-based classifier

Due to the limited capability of backbone network models such as ResNet [51] and Swin-Base [52] in effectively extracting foreground features from images, we employ a self-distillation learning model to extract foreground features.

The architecture of the self-distillation learning model is shown in Fig. 3. This model is demonstrated using a single pair of views (x_1,x_2) for simplicity and clarity. It applies two distinct random transformations to an input image and provides them as inputs to both the student and teacher networks. Although these networks have identical

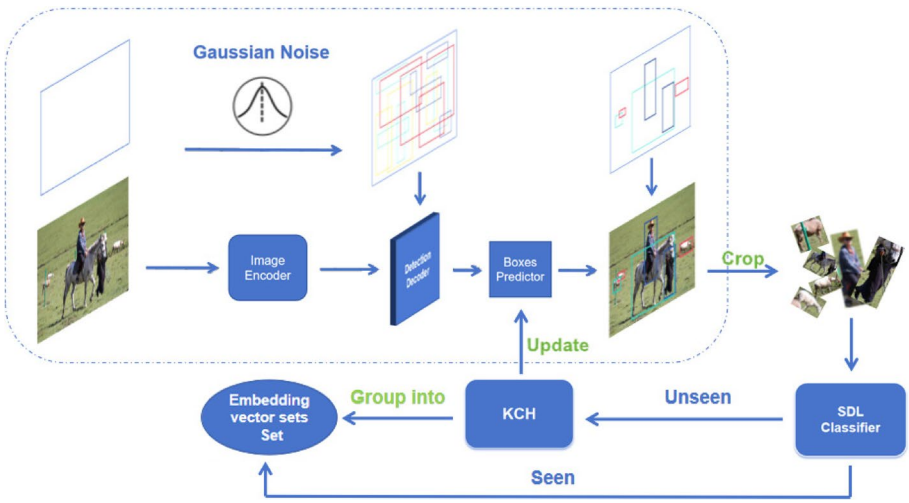


Fig. 2 The comprehensive structure of our model

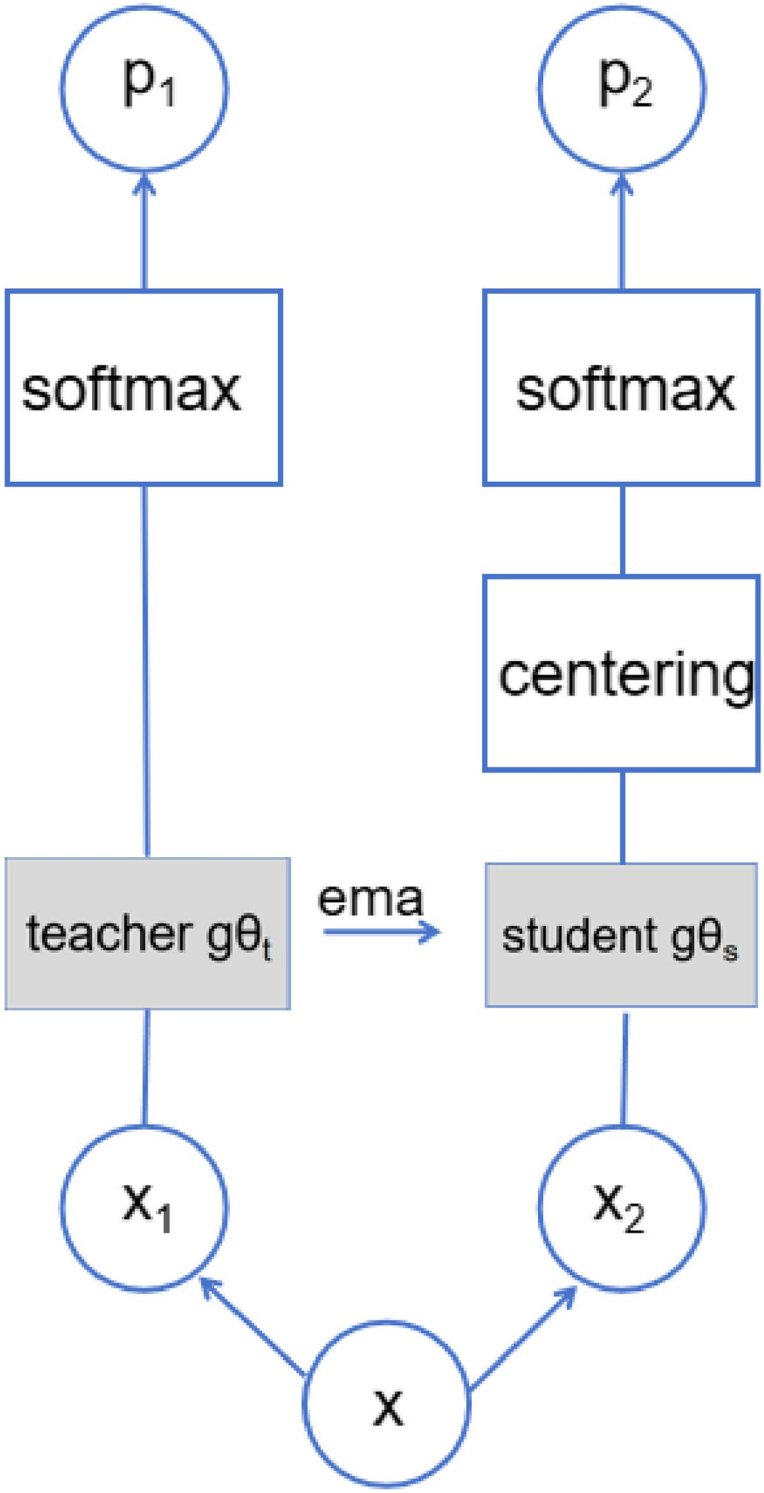


Fig. 3 Architecture of self-distillation learning model

structures, their parameters are different. The teacher network generates K-dimensional feature vectors that are normalized using a temperature softmax function. These feature vectors are then compared using a cross-entropy loss to measure their similarity

[53]. The teacher network's output is normalized by calculating the mean over the batch. The student network [53] $g_{\theta s}$ is a neural network model that learns to perform a task by trying to mimic or replicate the behavior of the teacher network [53] $g_{\theta t}$. During the training phase, the student network is updated using standard backpropagation techniques, where gradients are calculated based on the difference between the student's predictions and the teacher's outputs. The goal is for the student network to learn representations that are good enough to match those produced by the teacher. For an input image x, the student and teacher network each produce a set of probabilities across M categories, indicated as P_s for the student and P_t for the teacher. Their probabilities $P_s(x)$ are the result of applying a softmax function to normalize the outputs from the network $g_{\theta s}(x)$. More precisely:

$$P_s(x)^{(i)} = \frac{exp(\mathrm{g}_{\theta s}(x)^{(i)}/\tau_s)}{\sum_{m=1}^{M} exp(\mathrm{g}_{\theta s}(x)^{(m)}/\tau_s)}, \tag{1}$$

with $\tau_s > 0$, a temperature parameter is utilized to regulate the sharpness of the output distribution, with a corresponding expression governing P_t when modulated by the temperature τ_t.

In our classifier, we use the student network to extract feature vectors of objects. The student network has been trained on the ImageNet-200 datasets [54]. We calculate the Euclidean distance d_E between these feature vectors f_{1n} and the center vector of each cluster f_{2n} within every embedding vector sets as follows:

$$d_E = \sqrt{\sum_{n=1}^{N} (f_{1n} - f_{2n})^2}, \tag{2}$$

where $f_{1n} = (f_{11}, f_{12}, f_{13}, ..., f_{1N})$ and $f_{2n} = (f_{21}, f_{21}, f_{22}, ..., f_{2N})$, $\mathrm{N} \subset \mathrm{N}^+$ are both N-dimensional feature vectors. These cluster radii R_i in an embedding vector set E are formulated as follows:

$$R_i = \alpha * \max_{j=1,2,3...\mathrm{S_i}} \{\frac{\sum_{k=1}^{S_i} V_{ik}}{S_i} - V_{ij}\}, \tag{3}$$

where S is the vectors' number of a seen cluster, V_{ij} is a feature vector in a seen cluster, and α is a parameter which determines the size of a cluster's radius. Regarding parameter α, we will delve into the specifics in Section "Patch size".

After that, for an input feature vector, we compute its distance d_E with every cluster centroid in each embedding vector sets set. Then, we use whether d_E is less than the cluster radius R_i as a criterion to determine if the object belongs to a seen category i or an unseen class.

K-nearest neighbor-based cluster head

Enabling the model to cluster unseen classes provides it with the ability to differentiate among diverse unseen classes. We present a k-nearest neighbor-based cluster head to cluster these unseen classes. Algorithm 1 provides an overview on how the k-nearest neighbor-based cluster head clusters these unseen classes.

Algorithm 1 Algorithm of clustering unseen classes

Input: X - Feature vector of unseen object
$n\_neighbors \leftarrow [1, 2, ..., 20]$, $scores \leftarrow []$
for k in $n\_neighbors$ **do**
    $knn \leftarrow NearestNeighbors(n\_neighbors = k + 1)$
    $knn.fit(X)$
    $distances, \_ \leftarrow knn.kneighbors(X)$
    $score \leftarrow \frac{\sum_{i=1}^{n} distances[i,-1]}{n}$ ▷ Calculate average distance
    $scores.append(score)$
end for
$best\_k \leftarrow n\_neighbors[scores.index(\min(scores))]$
$neigh \leftarrow NearestNeighbors(n\_neighbors = best\_k)$
$neigh.fit(X)$
$neighbors \leftarrow neigh.kneighbors(X, return\_distance = False)$
for i, $neighbor\_indices$ in $enumerate(neighbors)$ **do**
    $cluster\_labels[i] \leftarrow neighbor_indices$
end for
$knn \leftarrow KNeighborsClassifier(n\_neighbors = best\_k)$
$knn.fit(X, cluster\_labels)$
$y\_pred \leftarrow knn.predict(X)$

The search space parameter is defined as n_neighbors, which means that we search for the optimal value of the n_neighbors within a range from 1 to 20 (excluding 20). The purpose of this is to experiment with different values of k (i.e., the number of nearest neighbors) and find the best value to construct the KNN model. Then, the cluster labels are assigned based on the indices of the nearest neighbors. After the prediction is completed, each unseen vector will have a label ID pointing to a specific cluster. Next, these unseen clusters will be divided to ensure that there are only ten clusters in each embedding vector set (we will explain in detail why only ten clusters are retained in an embedding vector set in Section "Patch size"). Then, we will integrate the new embedding vector set with unseen clusters into the collection of the embedding vector sets set to complete the update of seen categories. Simultaneously, we will update the boxes predictor so that the detector can detect the newly added seen categories.

Training

Diffusion detector

The L_2 loss function [55] using by diffusion model can be formulated as follows:

$$L_{\text{train}} = \frac{1}{2}||f_\theta(z_t, t) - z_0||^2, \tag{4}$$

which $t \in \{0,1,...T\}$ and the neural network $f_\theta(z_t, t)$ are trained to predict z_0 from z_t by minimizing the training objective with L_2 loss.

To establish a robust foundation for our object detection framework, we incorporated a pre-trained diffusion model [25] that has been extensively trained on MSCOCO [56] datasets. We specifically employed the weights of a model based on the ResNet50 [51] architecture, which has demonstrated remarkable performance in object detection tasks due to its deep residual learning capabilities. It is noteworthy that the original implementation

of the diffusion model involved a lengthy process with 500 sampling steps, which contributed to precise but computationally intensive inference. Considering the real-time requirements of our zero-shot classification task, we optimized the inference pipeline by reducing the number of sampling steps from 500 to 300. This strategic adjustment enabled us to substantially accelerate the inference speed of our diffusion-based detector while maintaining an acceptable trade-off between accuracy and real-time performance metrics.

Self-distillation model

In order to align the output distributions, the cross-entropy loss concerning the parameters of student network θ_s is minimized by the following:

$$\min_{\theta_s} H(P_t(x), P_s(x)), \tag{5}$$

where $H(a, b) = -alogb$.

In the following, a description is provided on how the problem in Eq. (5) is adapted for self-supervised learning. The initial step involves generating various distorted views or crops of an image using a multi-crop strategy [57]. Specifically, a set V of different views is created from a given image. To capture both global and local information, our model incorporates two global views (x_{1g} and x_{2g}) and multiple local views of smaller resolution. While all crops are processed by the student model, only the global views are utilized by the teacher model. This process promotes "local-to-global" correspondences [53]. The loss function is then minimized:

$$\min_{\theta_s} \sum_{x \in \{x_{1g}, x_{2g}\}} \sum_{\substack{x' \in V \\ x' \neq x}} H(P_t(x), P_s(x')), \tag{6}$$

We use vision transformer (ViT) [18] as the backbone of self-distillation and distance-based classifier. We employed four distinct model configurations with varying sizes and resolutions (ViT-S/16, ViT-S/8, ViT-B/16, and ViT-B/8) [53] to thoroughly investigate their feature extraction efficacy.

Experiment

Preparation

Datasets

We evaluate our model on the set of tasks $T = \{T_1, T_2\}$. Among them, task 1 is the clustering performance testing task. As shown in Table 1, for task 1, we use 10 classes from

Table 1 Datasets for each task

	Task 1			Task 2		
	Split	Samples	Classes	Split	Samples	Classes
STL-10	Train+test	13,000	10	-	0	0
ImageNet-10	Train+unlabeled	13,000	10	-	0	0
CIFAR-10	Train+test	60,000	10	Train+test	60000	10
CIFAR-100	Train+test	60,000	100	Train+test	60000	100
CUB	Train+Test	11788	200			

STL-10 [27], ImageNet-10 [28], and CIFAR-10 [26], 100 classes from CIFAR-100 [56] datasets. For task 2, we use CIFAR-10, CIFAR-100 and CUB [58]. Furthermore, we use pre-trained self-distillation models with two different resolutions and two different model sizes, resulting in four types of models. Therefore, in task 1, we plan to evaluate the performance and practicality of each method and model through thorough evaluation.

Evaluation metrics

In task 1, to assess our approach, we employ three commonly recognized metrics for clustering evaluation: The normalized mutual information (NMI), accuracy (ACC), and adjusted rand index (ARI).

The NMI is a metric that remains consistent regardless of the dataset's size. It effectively measures the extent of information overlap between the true labels and the labels assigned through clustering, indicating the quality of the clustering. This can be formulated as follows:

$$\mathrm{NMI}(U, V) = \frac{2 * I(U; V)}{H(U) + H(V)}, \tag{7}$$

where U and V are two sets of clusters, the shared information content of U and V is quantified by $I(U; V)$ which is the mutual information, while $H(U)$ and $H(V)$ represent the individual uncertainties of U and V.

Accuracy (ACC) measures the proportion of correctly clustered instances by comparing the cluster assignments with the ground truth labels, reflecting the clustering correctness. This can be formulated as follows:

$$\mathrm{ACC} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{n}} 1\{\mathrm{l_i} = \mathrm{m(c_i)}\}}{\mathrm{n}}, \tag{8}$$

where n is the number of samples, c_i is the cluster assignment for sample i, l_i is the true label for sample i, m is the mapping function from clusters to true label s, and l is the indicator function.

Adjusted rand index (ARI) which can adjust the similarity between the true clustering and the predicted clustering with a value that can be compared across different datasets. This can be demonstrated as follows:

$$\mathrm{ARI} = \frac{\mathrm{RI} - \mathrm{Expected\ RI}}{\mathrm{Max\ RI} - \mathrm{Expected\ RI}}, \tag{9}$$

where RI is the rand index, which is calculated as follows:

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}, \tag{10}$$

in this context, TP is the count of true positive pairs, TN is the number of true negative pairs, FP is the count of falsely identified positive pairs, and FN is the count of falsely identified negative pairs. The expected RI depends on the marginal totals of a contingency table (or confusion matrix) of the cluster assignment.

In task 2, we use three evaluation metrics: precision, recall, and F1 scores to assess model performance.

Among them, precision is a measure of the accuracy of a classification model, which indicates the proportion of the true positive predictions in the total predicted positives. The precision metric is computed by dividing the number of true positives by the total number of instances classified as positive, which includes both true positives and false positives. High precision indicates that an algorithm generated a significant number of relevant results compared to irrelevant ones. Precision can be formulated as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

Recall measures the ability of a model to find all the relevant cases within datasets. It represents the fraction of actual positives correctly identified by the classifier out of all actual positives. Mathematically, it is the ratio of correctly detected positive cases to the total actual positive cases. High recall indicates that the class is correctly recognized to a large extent. Recall (sensitivity) can be presented as follows:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

The F1-score is calculated as 2 times the product of precision and recall divided by the sum of precision and recall, thereby balancing the trade-off between false positives and false negatives. It is a measure that combines precision and recall, considering both false positives and false negatives, to provide a single score for model accuracy, providing a single score that weighs both the concerns of finding all relevant instances (recall) and returning only relevant instances (precision). F1-score reaches its best value at 1 (perfect precision and recall) and worst at 0. F1-score can be demonstrated as follows:

$$\text{F1} = \frac{2 * \text{TP}}{2 * \text{TP} + \text{FP} + \text{FN}} \tag{13}$$

Implementation details

The detector of our model is based on diffusion detector [25] with ResNet-50 [51], and Swin-Base [52] backbone. We use the detector to detect both seen and unseen objects. More precisely, we employ a diffusion model with the ResNet-50 [51] architecture as the backbone network to extract objects from images. Additionally, this diffusion model has been pre-trained on the MSCOCO [56] datasets.

In task 1 and task 2, we tested 4 self-distillation models, whose parameter counts and resolutions are shown in Table 2.

It is worth noting that larger model parameters and smaller resolution values indicate better performance of the model. Furthermore, the self-distillation model we use has been pre-trained on the ImageNet datasets [54].

Table 2 Parameters and resolution of each model

	ViT-B/8	ViT-B/16	ViT-S/8	ViT-S/16
#params	85M	85M	21M	21M
resolution	8	16	8	16

Table 3 Cluster comparison for task 1

	CIFAR-10			STL-10			ImageNet-10			CIFAR-100		
	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
K-means	0.759	0.862	0.726	0.945	0.975	0.944	0.972	0.990	0.978	0.623	0.482	0.326
Agglomerative	0.722	0.742	0.632	0.930	0.964	0.924	0.958	0.981	0.958	0.622	0.472	0.308
CC	0.705	0.790	0.637	0.764	0.850	0.726	0.859	0.893	0.822	0.431	0.429	0.266
SKZC	**0.909**	**0.962**	**0.919**	**0.976**	**0.991**	**0.981**	**0.983**	**0.994**	**0.987**	**0.801**	**0.818**	**0.677**

The bold values represent the maximum values in the same row

Table 4 Model comparison for task 1

	CIFAR-10			CIFAR-100		
	NMI	ACC	ARI	NMI	ACC	ARI
ViT-B/8	**0.909**	**0.962**	**0.919**	**0.801**	**0.818**	**0.677**
ViT-B/16	0.897	0.956	0.905	0.800	0.818	0.678
ViT-S/8	0.855	0.934	0.86	0.758	0.775	0.612
ViT-S/16	0.832	0.922	0.837	0.715	0.734	0.552

The bold values represent the maximum values in the same row

For the hyperparameters, we set the value of α to 0.75 and set the number of clusters in each embedding set at 7.

Results and discussion

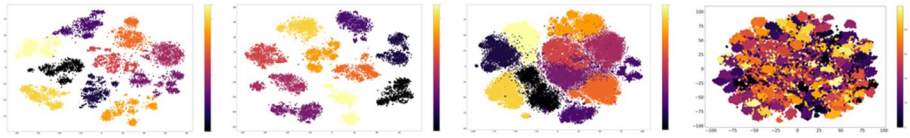
Clustering performance

The quality of clustering directly influences the outcome of the entire classification task; therefore, the model's ability to effectively cluster data is of crucial significance. The clustering performance of our model is shown in Table 3, and we tested the ViT-B/8 self-distillation model on the STL-10, ImageNet-10, CIFAR-10, and CIFAR-100 datasets. Apart from the contrastive clustering algorithms, all the algorithms tested in our study employed feature vectors extracted by a self-distillation model for clustering operations. It is evident that compared to contrastive clustering algorithm, the traditional clustering algorithm also achieved promising performance. This indicates the effectiveness of self-distillation models.

In Table 4, we conducted tests using the CIFAR-10 and CIFAR-100 datasets and concluded that the ViT-B/8 model has the best performance. It can be clearly seen that the model possesses a greater quantity of parameters and enhanced resolution typically demonstrates improved performance outcomes. Therefore, due to the substantial number of model parameters and higher resolution afforded by ViT-B/8, it exhibits the most superior performance. Besides, considering the requirement for real-time classification, we are willing to sacrifice some model performance to enhance the inference speed of the model.

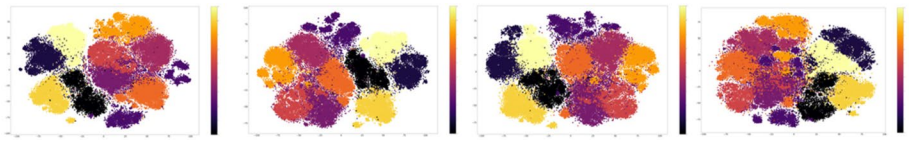
The clustering visualization results of the ViT-B/8 model on STL-10, ImageNet-10, CIFAR-10, and CIFAR-100 are shown in Fig. 4.

The clustering visualization results for the ViT-B/8, ViT-B/16, ViT-S/8, and ViT-S/16 models on CIFAR-10 and CIFAR-100 datasets are shown in Fig. 5.



(a) STL-10 (b) ImageNet-10 (c) CIFAR-10 (d) CIFAR-100

Fig. 4 Different datasets' visualization results of cluster. **a** STL-10. **b** ImageNet-10. **c** CIFAR-10. **d** CIFAR-100



(a) base 8 (b) base 16 (c) small 8 (d) small 16

Fig. 5 Different models' visualization results of cluster. **a** Base 8. **b** Base 16. **c** Small 8. **d** Small 16

Table 5 Model comparison on CIFAR-10 for task 2

	Airplane			**Automobile**			**Bird**		
	Precision	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**
ViT-B/8	**0.913**	**0.846**	**0.878**	0.935	**0.894**	0.914	**0.929**	**0.783**	**0.850**
ViT-B/16	0.879	0.832	0.855	**0.959**	0.881	**0.918**	0.922	0.758	0.832
ViT-S/8	0.866	0.814	0.839	0.916	0.889	0.902	0.903	0.701	0.789
ViT-S/16	0.807	0.741	0.773	0.909	0.869	0.889	0.826	0.644	0.724
	Dog			Frog			Horse		
	Precision	Recall	f1-score	Precision	Recall	f1-score	Precision	Recall	f1-score
ViT-B/8	0.863	**0.795**	**0.828**	**0.950**	0.881	**0.914**	0.928	**0.855**	**0.890**
ViT-B/16	**0.866**	0.771	0.816	0.903	**0.901**	0.902	**0.939**	0.843	0.888
ViT-S/8	0.791	0.746	0.768	0.837	0.842	0.839	0.887	0.803	0.843
ViT-S/16	0.787	0.689	0.735	0.800	0.806	0.803	0.865	0.762	0.810
	Cat			Deer			Ship		
	Precision	Recall	f1-score	Precision	Recall	f1-score	Precision	Recall	f1-score
ViT-B/8	**0.796**	**0.796**	**0.796**	**0.898**	**0.844**	**0.870**	**0.948**	0.860	0.902
ViT-B/16	0.781	0.795	0.788	0.874	0.821	0.847	0.945	**0.865**	**0.903**
ViT-S/8	0.715	0.724	0.720	0.824	0.781	0.802	0.942	0.854	0.896
ViT-S/16	0.644	0.697	0.669	0.786	0.717	0.750	0.886	0.835	0.860
	Truck			Micro avg			Macro avg		
	Precision	Recall	f1-score	Precision	Recall	f1-score	Precision	Recall	f1-score
ViT-B/8	**0.949**	0.862	**0.904**	**0.910**	**0.842**	**0.874**	**0.911**	**0.842**	**0.875**
ViT-B/16	0.905	**0.893**	0.899	0.896	0.836	0.865	0.897	0.836	0.865
ViT-S/8	0.918	0.847	0.881	0.857	0.800	0.828	0.860	0.800	0.828
ViT-S/16	0.873	0.854	0.863	0.816	0.761	0.788	0.816	0.761	0.788

The bold values represent the maximum values in the same row

Classification performance

Based on the results shown in Table 5, we tested 4 self-distillation models on the CIFAR-10 datasets. It is easy to see from the table that in the CIFAR-10 dataset's ten categories, the base-sized model exhibits the best performance, and the model with a resolution of 8 achieves the highest precision and recall scores in 70% of the categories, as well

as the highest f1-scores in 80% of the categories. Therefore, we can consider the ViT-B/8 model, with larger model parameters and higher resolution, has the best model performance.

Similar to the result in Table 5, the test results on the CIFAR-100 datasets show that the ViT-B/8 model achieved the highest scores across all metrics. It indicates that the ViT-B/8 model has the best model performance. The result is shown in Table 6.

Model size

The base model (B) has a greater number of parameters; thus, it is more likely to capture complex image characteristics, which generally leads to better generalization ability and inference accuracy when there is an abundance of training data. Although the small model (S) has fewer parameters and a lower computational cost, making it potentially more suitable for resourse-constrained environments or scenarios sensitive to latency, it might prevent overfitting due to its simplicity especially in cases where training data is not extensively available.

Patch size

Models with a smaller patch size (e.g., ViT-B/8) generate longer sequences and therefore have the capacity to capture finer-grained image information. This can aid in learning more complex image patterns, potentially leading to improved accuracy of the model. However, longer sequences also mean higher computational costs and increased memory demands. In contrast, a larger patch size (e.g., ViT-Base/16) reduces sequence length, lowering computational complexity but potentially at the loss of some detailed information.

Subsequently, we designed an experiment to compare the performance gap between our model and other more recent zero-shot image classification models. As shown in Table 7, our model (based on ViT-B/16) achieved the best overall performance. Compared to SGFR [59], our model demonstrated improvements of 10.9%, 13.3%, and 7.8% on Sacc, Uacc, and H metrics, respectively. We attribute the enhancements in our model to the methodological design and the choice of the number of clusters in each embedding set and the value of α.

Ablation study

We designed ablation experiments to study the contributions of SDDC and KCH in the model (see Table 8). Missing SDDC module means we replace the self-distillation

Table 6 Model comparison on CIFAR-100 for task 2

	Micro avg			Macro avg		
	Precision	**Recall**	**f1-score**	**Precision**	**Recall**	**f1-score**
ViT-B/8	**0.715**	**0.692**	**0.704**	**0.735**	**0.692**	**0.708**
ViT-B/16	0.705	0.680	0.692	0.725	0.680	0.696
ViT-S/8	0.661	0.639	0.649	0.68	0.639	0.652
ViT-S/16	0.615	0.593	0.604	0.634	0.593	0.606

The bold values represent the maximum values in the same row

Table 7 Comparisons in task 2 on the CUB dataset

Method	140:60 splits			100:100 splits		
	Sacc	Uacc	H	Sacc	Uacc	H
DEM	37.9	27.6	27.5	38.6	27.8	28.0
Lisgan	48.2	34.7	36.9	47.3	31.7	34.5
ICCE	48.3	36.7	38.6	47.2	36.5	39.1
AVAEDS	45.2	38.3	41.1	46.2	37.8	41.5
SGFR [59]	41.6	41.1	45.6	40.5	47.0	45.8
SKZC	**52.5**	**54.4**	**53.4**	**55.6**	**62.3**	**58.8**

The bold values represent the maximum values in the same row

Table 8 Ablation experimental results of our model

ID	SDDC	KCH	Precision		Recall		f1-score	
			Micro avg	Macro avg	Micro avg	Macro avg	Micro avg	Macro avg
1	√	×	0.202	0.204	0.156	0.156	0.176	0.176
2	×	√	0.416	0.462	0.341	0.341	0.375	0.377
3	√	√	**0.910**	**0.911**	**0.842**	**0.842**	**0.874**	**0.875**
4	×	×	0.128	0.120	0.112	0.112	0.120	0.096

The bold values represent the maximum values in the same row

network with a standard ResNet backbone, and missing KCH refers to using the K-Means clustering algorithm in place of the KCH module.

When SDDC and KCH are missing (row 4), the model performs the worst. Adding only SDDC (row 1) will improve the model's ability to cluster, and with high-quality clustering, the model is likely to demonstrate enhanced performance in classification. Adding only KCH (row 2) will directly improve the model's ability to classify. When adding both SDDC and KCH, the model performs the best. Therefore, the presence or absence of both SDDC and KCH will affect the performance of the model and the optimal performance is obtained when both components are present and work together.

The calculation of cluster radius

In our experiments, we found that a category's cluster may have several points that are far from the cluster center. If we simply use the Euclidean distance from the furthest point to the cluster center as cluster radius, it could lead to a large number of misjudged unseen objects. Moreover, the choice of the cluster radius can affect the accuracy and false positive rate of unseen object identification. Therefore, in order to calculate the optimal cluster radius, we designed an experiment.

In the experiment, we defined the distance value from all points in the cluster to the cluster center point as d_i. Then, we place this distance d_i into a set D. Afterward, we select the smallest α percentile values from set D and discard any remaining points that fail to meet the specified conditions. Finally, we set the maximum value in set D as the cluster radius, allowing α to increase from 0 to 1 in increments of 0.01 (It is evident that α is positively correlated with the radius of the cluster). We then plotted the curve showing the change in the accuracy rate of unseen identification as α varied, as shown in Fig. 7. Moreover, we also plotted the curve of the harmonic mean of our model on the

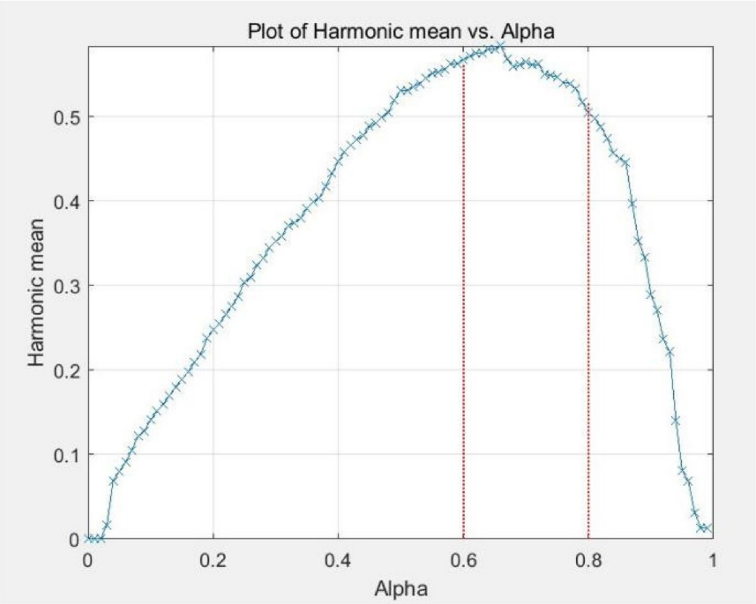


Fig. 6 The curve of harmonic mean as a function of alpha

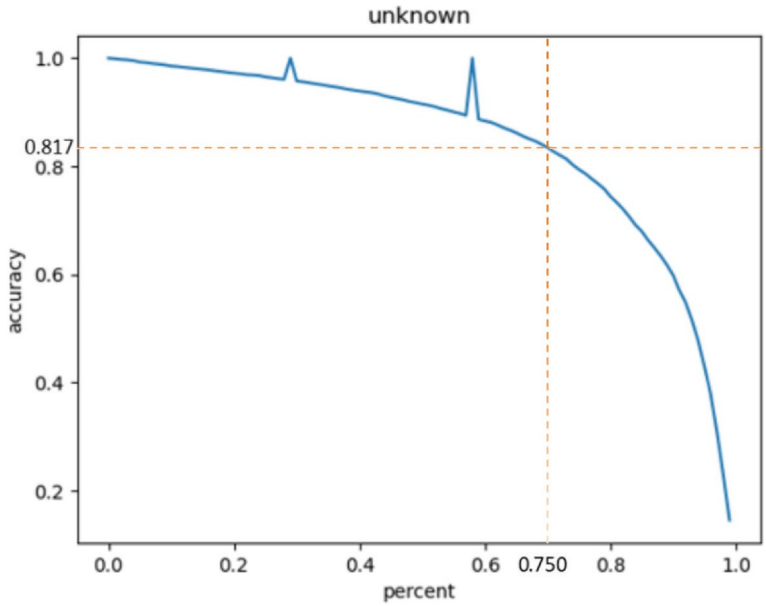


Fig. 7 Change in the accuracy rate of unseen identification with the variation of α value curve

CUB dataset as a function of α, as shown in Fig. 6. The maximum value is reached within the interval from 0.6 to 0.8, strictly speaking at 0.66. However, considering that the optimal value of α might differ across datasets, and with a view to generality, we set α to 0.75. The misjudgment rate of unseen is negatively correlated with the accuracy rate.

Number of clusters in each embedding set

As the task progresses, more and more feature vectors will inevitably appear in the embedding set. It is impractical to perform clustering operations only in one embedding

set, as it would consume a lot of time and computational power. It is obvious that these feature vectors follow the same distribution. Based on the above fact, we decided to place clusters in different embedding sets and perform clustering operations there, rather than just in one embedding set. However, it is worth noting that if the number of clusters in the embedding set is too small, it may lead to excessively small differences in the distances between the input feature vector and the different cluster center vectors, resulting in incorrect judgments of input feature vectors. Therefore, we designed an experiment to explore the optimal number of clusters in each embedding set. The performance metrics NMI, ARI, and ACC indices for clustering as a function of the number of clusters in the embedding set are shown in Fig. 8.

As shown in Fig. 8, once the number of in the same embedding set exceeds 7, the three indicators NMI, ARI, and ACC plummet sharply, indicating a rapid deterioration in clustering performance within that embedding set. A decline in clustering performance can lead to a model propensity for misclassifying unseen classes as seen ones (disorganized vector distribution within each cluster, resulting in an excessively large radius). Moreover, considering the fact that too few clusters may result in seen categories being misidentified as unseen categories, an excessively high number of clusters can lead to feature vectors becoming overly concentrated within the embedding set, thereby causing unseen classes to be erroneously identified as seen classes. Therefore, we decide to fix the number of clusters in each embedding set at 7. Furthermore, we have plotted the visualization of the growth process of the number of clusters in the embedding set, as shown in Fig. 9.

Conclusions

In this work, we propose a novel zero-shot classification model named self-distillation and k-nearest neighbor-based zero-shot classification model. We propose a new method including k-nearest neighbor-based cluster head (KCH) and self-distillation and distance-based classifier (SDDC). Abundant experiments demonstrate the effectiveness

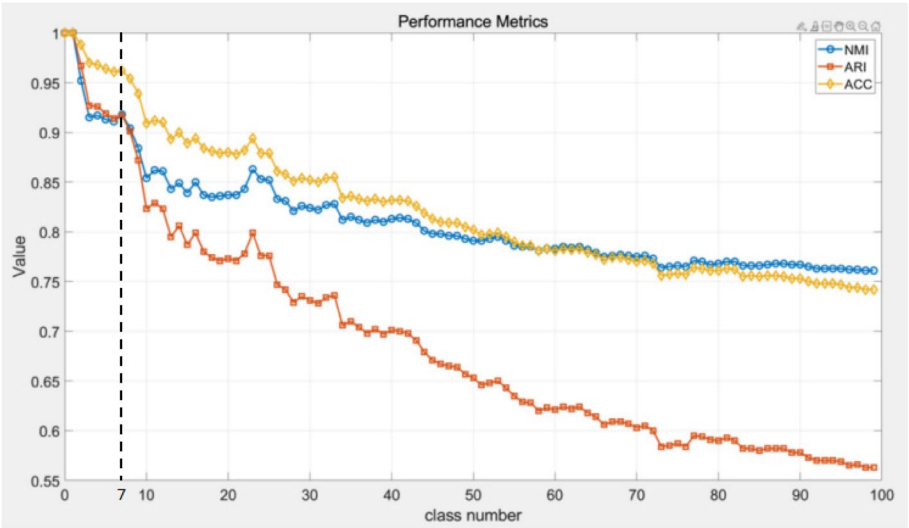


Fig. 8 Clustering performance in the same embedding set as a function of a number of cluster curve

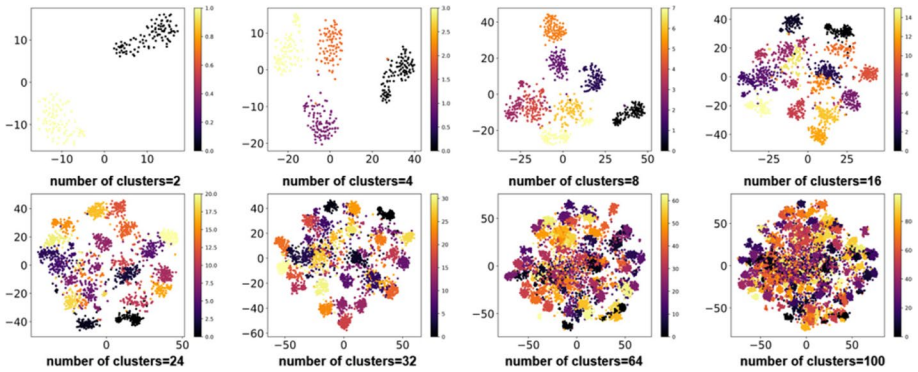


Fig. 9 Visualization of the growth process of the number clusters in the embedding set

of our model on zero-shot classification problems. In cluster performance, on datasets CIFAR-10, CIFAR-100, ImageNet-10, and STL-10, our model outperforms the contrastive clustering model across the board. In the classification task, we achieved a precision of 0.910 and a recall of 0.842 on CIFAR-10 datasets, a precision of 0.737, and a recall of 0.688 on CIFAR-100 datasets for the macro-average.

While our model has shown promising results on certain datasets (dataset CIFAR et al.), it still has limitations (dataset CUB). Real-world objects are incredibly diverse and complex, often exceeding what can be experimentally simulated. Take birds as an example: there are over 9000 known species of birds, each with distinct appearances. Even humans find it challenging to differentiate closely related bird species due to their similar features. Consequently, for several closely related categories, our model may perform poorly in classification tasks because the feature vectors of these categories are too close together within the embedding space. To address this issue, future research will delve deeper into the selection of the number of clusters in each embedding set and optimizing α parameter. By dynamically adjusting these based on the characteristics of feature vectors within the embedding set, we aim to achieve a more reasonable distribution of feature vectors, ultimately enhancing the model's ability to classify categories with similar features.

In future work, we hope to continue improving our model structures and apply it to open-world object detection problems. By exploring these uncharted territories, we aim to bridge the gap between academic experimentation and real-world applicability, making our model more robust and versatile in handling diverse and dynamic environments.

Abbreviations

KCH	K-nearest neighbor-based cluster head
SDDC	Self-distillation and distance-based classifier
ViT-B/8	Vision transform model with base size and 8 resolution
ViT-S/8	Vision transform model with small size and 8 resolution
ViT-B/16	Vision transform model with base size and 16 resolution
ViT-S/16	Vision transform model with small size and 16 resolution

Acknowledgements
The authors thank Jiajie Li from the University of Electronic Science and Technology of China for organizing the image data of this paper and to Yong Sun for providing the graphics card support for the paper experiment.

Authors' contributions
The authors Muyang Sun and Haitao Jia contributed equally to this study.

Funding
This research was funded by the Science and Technology Program of Sichuan (grant number 2022ZDZX0005, 2023ZHCG0013), the central government guides local special fund projects of the Mianyang Municipality Science and Technology Bureau (grant number 2022ZYDF009).

Availability of data and materials
Pretrained models of DiffusionDet are from https://github.com/ShoufaChen/DiffusionDet.
Pretrained models of self-distillation model are from: https://github.com/facebookresearch/dino.

Declarations

Competing interests
The authors declare that they have no competing interests.

Received: 15 February 2024 Accepted: 13 April 2024
Published online: 22 April 2024

References

1. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In *Proceedings of the Neural Information Processing Systems (NIPS).
2. Chang D, Ding Y, Xie J, Bhunia AK, Li X, Ma Z, et al., (2020) The devil is in the channels: mutual-channel loss for fine-grained image classification, TIP
3. Feinerer I, Hornik K (2020) wordnet: WordNet Interface. R package version 0.1-15. [Online]. Available: https://CRAN.R-project.org/package=wordnet
4. Yang X, Deng C, Wei K, Yan J, Liu W (2020) Adversarial learning for robust deep clustering. In *Proceedings of the Neural Information Processing Systems (Neur)*, 2020.
5. Ju Y, Lam KM, Chen Y, Qi L, Dong J (2020) Pay attention to devils: a photometric stereo network for better details. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJI)*.
6. Li H, Wang F, Liu J, Huang J, Zhang T, & Yang S. (2022). Micro-knowledge embedding for zero-shot classification. Computers and Electrical Engineering, 101. https://doi.org/10.1016/j.compeleceng.2022.108068
7. Wang W, Zheng VW, Yu H, Miao C (2019) "A survey of zero-shot learning: settings, methods and applications," in *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1-19.
8. Lampert CH, Nickisch H and Harmeling S. (2009) Learning to detect unseen object classes by between-class attribute transfer, CVPR
9. Palatucci Mark, A. Pomerleau Dean, E. Hinton Geoffrey and Tom Michael Mitchell, (2009) Zero-shot learning with semantic output codes, NIPS
10. Frome A, Corrado GS, Shlens J, Bengio S, Dean J, Ranzato M'A and Mikolov T. (2013) DeViSe: a deep visual-semantic embedding model, NIPS
11. Norouzi M, Mikolov T, Bengio S, Singer Y, Shlens J, Frome A, et al., (2014) Zero-shot learning by convex combination of semantic embeddings, ICLR
12. Wang X, Ye Y, Gupta A (2018) Zero-shot recognition via semantic embeddings and knowledge graphs, CVPR
13. Kampffmeyer M, Chen Y, Liang X, Wang H, Zhang Y, Xing EP (2019) Rethinking knowledge graph propagation for zero-shot learning, CVPR
14. Liu S, Chen J, Pan L, Ngo CW, Chua TS, Jiang YG (2020) Hyperbolic visual embedding learning for zero-shot recognition, CVPR
15. Romera-Paredes B and Torr PHS. (2015) An embarrassingly simple approach to zero-shot learning, ICML
16. Akata Z, Perronnin F, Harchaoui Z, Schmid C (2013)Label-embedding for attribute-based classification, CVPR
17. Akata Z, Reed S, Walter D, Lee H, Schiele B (2015)Evaluation of output embeddings for fine-grained image classification, CVPR
18. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, et al. (2020) An image is worth 16x16 words: transformers for image recognition at scale. preprint arXiv:2010.11929
19. Zhao H, Jia J, Koltun V. (2020) Exploring self-attention for image recognition. In CVPR,
20. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H (2020) Training data-efficient image transformers & distillation through attention. preprint arXiv:2012.12877
21. Hinton G, Vinyals O, Dean J. (2015) Distilling the knowledge in a neural network. preprint arXiv:1503.02531
22. Cheng R, Wu B, Zhang P, Vajda P, Gonzalez JE (2021) Data-efficient language-supervised zero-shot learning with self-distillation, 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Nashville, TN, USA, pp. 3113-3118, https://doi.org/10.1109/CVPRW53098.2021.00348
23. X. Kong , Kong X et al. (2022) En-compactness: self-distillation embedding & contrastive generation for generalized zero-shot learning, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, pp. 9296-9305,https://doi.org/10.1109/CVPR52688.2022.00909
24. Pennington J, Socher R, Manning CD (2014) , EMNLP
25. Chen S, Sun P, Song Y, Luo P (2022). DiffusionDet: diffusion model for object detection. arXiv. https://doi.org/10.48550/arXiv.2211.09788
26. Krizhevsky A, Hinton G. (2009) Learning multiple layers of features from tiny images. Master's thesis, Dept. Comp. Sci., Univ; Toronto

27. Chang J. Wang L, Meng G, Xiang S, and Pan C. (2017) Deep adaptive image clustering. In Proceedings of the IEEE international conference on computer vision, 5879–5887
28. Coates A, Ng, Lee H. (2011) An analysis of single-layer networks in unsupervised feature learning, in Proc. 14th Int. Conf. Artif. Intell. Statist. (AISTATS), pp. 215–223
29. Li Y, Hu P, Liu Z, Peng D, Zhou JT, Peng X (2021). Contrastive clustering. 35h AAAI Conference on Artificial Intelligence, AAAI 2021, 10A, 8547–8555
30. Li J, Jin M, Lu K et al (2019) Leveraging the invariant side of generative zero-shot learning[C]. In: Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, 7402–7411
31. Caixia Y, Chang X, Li Z et al (2021) Zeronas: differentiable generative adversarial networks search for zero-shot learning[J]. IEEE Trans Pattern Anal Mach Intell 2021:1–9
32. Shermin T, Teng SW, Sohel F et al (2021) Bidirectional mapping coupled GAN for generalized zero-shot learning[J]. IEEE Trans Image Process 31:721–733
33. Xian Y, Lorenz T, Schiele B, et al (2018) Feature generating networks for zero-shot learning. In CVPR, 5542–5551
34. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein Gan[C]. ICML
35. Felix R, Kumar VBG, Reid I et al (2018) Multi-modal cycle-consistent generalized zero-shot learning[C]. In: Proceedings of European Conference on Computer Vision, Munich, 21–37
36. Han Z, Fu Z, Li G et al (2020) Inference guided feature generation for generalized zero-shot learning. Neurocomputing 430:150–158
37. Vyas MR, Venkateswara H, Panchanathan S (2020) Leveraging seen and unseen semantic relationships for generative zero-shot learning[C]. European Conference on Computer Vision. Springer, Cham, 70–86
38. Chen Z, Huang Z, Li J et al (2021) Entropy-based uncertainty calibration for generalized zero-shot learning[C]. Australasian Database Conference. Springer, Cham, 139–151
39. Schonfeld E, Ebrahimi S, Sinha S, et al (2019) Generalized zero-and few-shot learning via aligned variational autoencoders[C]. Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition, 8247 – 8255
40. Verma VK, Arora G, Mishra A et al (2018) Generalized zero-shot learning via synthesized examples[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE, 4281–4289
41. Chen SM, Xie GS, Liu Y et al (2021) HSVA: hierarchical semantic-visual adaptation for zero-shot learning[C]. 35th Conference on Neural Information Processing Systems
42. Ma P, Hu X (2020) A variational autoencoder with deep embedding model for generalized zero-shot learning[C]. Proceedings of the AAAI Conference on Artificial Intelligence, 11733–11740
43. Chen L, Zhang H, Xiao J, Liu W, Chang SF (2018)Zero-shot visual recognition using semantics-preserving adversarial embedding networks[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1043–1052
44. Guan J, Lu Z, Xiang T et al (2020) Zero and few shot learning with semantic feature synthesis and competitive learning[J]. IEEE Trans Pattern Anal Mach Intell 43(7):2510–2523
45. Pandey A, Mishra A, Verma VK et al (2020) Stacked adversarial network for zero-shot sketch based image retrieval[C]. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2529–2538
46. Das D, George Lee CS (2019) Zero-shot image recognition using relational matching, adaptation and calibration[C]. 2019 International Joint Conference on Neural Networks (IJCNN). IEEE, 1–8
47. Xie GS, Zhang XY, Yao Y et al (2021) Vman: a virtual mainstay alignment network for transductive zero-shot learning. IEEE Trans Image Process 30:4316–4329
48. Liu Y, Tuytelaars T (2020) A deep multi-modal explanation model for zero-shot learning. IEEE Trans Image Process 29:4788–4803
49. Yang Hu, Wen G, Chapman A et al (2021) Graph-based visual-semantic entanglement network for zero-shot image recognition[J]. IEEE Trans Multimed 24:2473–2487
50. Luo Y, Wang X, Pourpanah F (2021) Dual VAEGAN: a generative model for generalized zero-shot learning. Appl Soft Comput 107:107352
51. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In Proceed- ings of the IEEE conference on computer vision and pattern recognition, pages 770–778
52. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022
53. Caron M, Touvron H, Misra I, Jegou H, Mairal J, Bojanowski P, & Joulin A. (2021). Emerging properties in self-supervised vision transformers. arXiv
54. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC and Fei-Fei L (2015) Imagenet large scale visual recognition challenge. IJCV
55. Ho Jonathan, Jain Ajay, Abbeel Pieter (2020) Denoising diffusion probabilistic models. Adv Neural Information Process Syst 33:6840–6851
56. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In ECCV, pages 740–755. Springer
57. Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A (2020) Unsupervised learning of visual features by contrasting cluster assignments. In NeurIPS
58. Wah C, Branson S, Welinder P, Perona P, Belongie S. "The Caltech-UCSD Birds-200-2011 Dataset," California Institute of Technology, CNS-TR-2010-001, 2010.
59. Li X, Fang M, Li H and Chen B (2024) Selective-generative feature representations for generalized zero-shot open-set classification by learning a tightly clustered space. Expert Syst Appl 245, 123062, 2024. https://doi.org/10.1016/j.eswa.2023.123062. Available: https://www.sciencedirect.com/science/article/pii/S0957417423035649