Study Report on Coresets selection for Open-Set with Fine-Grained task and Self-Supervised machine learning

Wentao Ma[1,*] and Chenyou Ma[2]

[1] Suzhou North America High School, Suzhou, China, Tony.ma.2006@outlook.com
[2] Curtin Singapore, Singapore, mchenyou08@gmail.com

*corresponding author

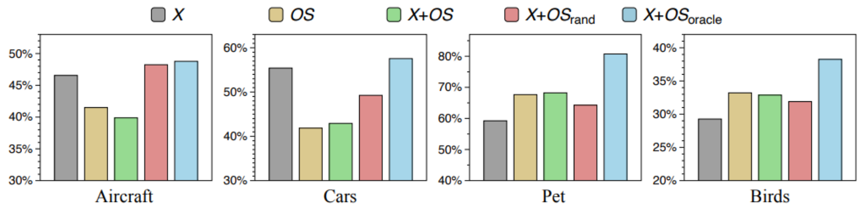
Abstract. The paper we studied introduces concepts about machine learning and algorithms for corsets selection. A group of researchers from KAIST have developed an effective coresets selection algorithm to help Open-set self-supervised learning on image classification tasks. Experiments are being established using big open sets and making it more fine-grained, models will be trained using the fine-grained sets in order to classify and annotate different objects in the picture. In our studies, we have experimented using pictures of aircraft to train the model. An algorithm to select coreset named SimCore is being developed, and the group of researchers had found that by merging the coreset selected from the open set with the target dataset, had made the training process of the model more efficient.

Keywords: Coreset, classification, Self supervised machine learning.

1. Introduction

The big open set contains 1,281,167 random pictures of different objects, then it has been divided into 10 different fine-grained sets of aircraft, pets, cub, dog, flower, action, indoor pictures, texture, faces and food. Due to the lack of computer space, only the aircraft set of 6667 picture is being tested. [3]

In the figure 1, (X) is the target dataset, and tests had been done with (X) and Open Set (OS). Tests are also being done using the core During our research, we wish to experiment on the coresets selection algorithm and find out how effective it is on improving machine learning tasks, and try to study the concepts of Coresets and machine learning.[3]



Target (X)	Classes for OS_{oracle} (#)
Aircraft	airliner, warplane, ... (8)
Cars	convertible, jeep, ... (10)
Pet	Persian cat, beagle, ... (24)
Birds	goldfinch, junco, ... (20)

Figure 1 Training efficiency result

1.1. Coresets

Coresets are beneficial to the effectiveness of machine learning, they are selected terms from a larger dataset that represents and summarize the essential properties and structures of the original dataset. They are smaller in size compared to the larger dataset, therefore it significantly decreases the computational and memory resources required. [4,5] They are usually used in tasks like clustering, regression and optimization, since they can be more efficiently done with a smaller data set size. [4]

Data selected for a Coreset are essential for the learning process. Coresets can help overcome challenges such as mislabelled pictures. [4] Problems like this may happen in large datasets. Coresets also need to not choose data with bad qualities, even when the pictures are all correctly labeled, such samples might cause training to happen in a way that is not preferred. For example in the figure 2 below, selecting such an image might lead to crucial consequences to the result of training.

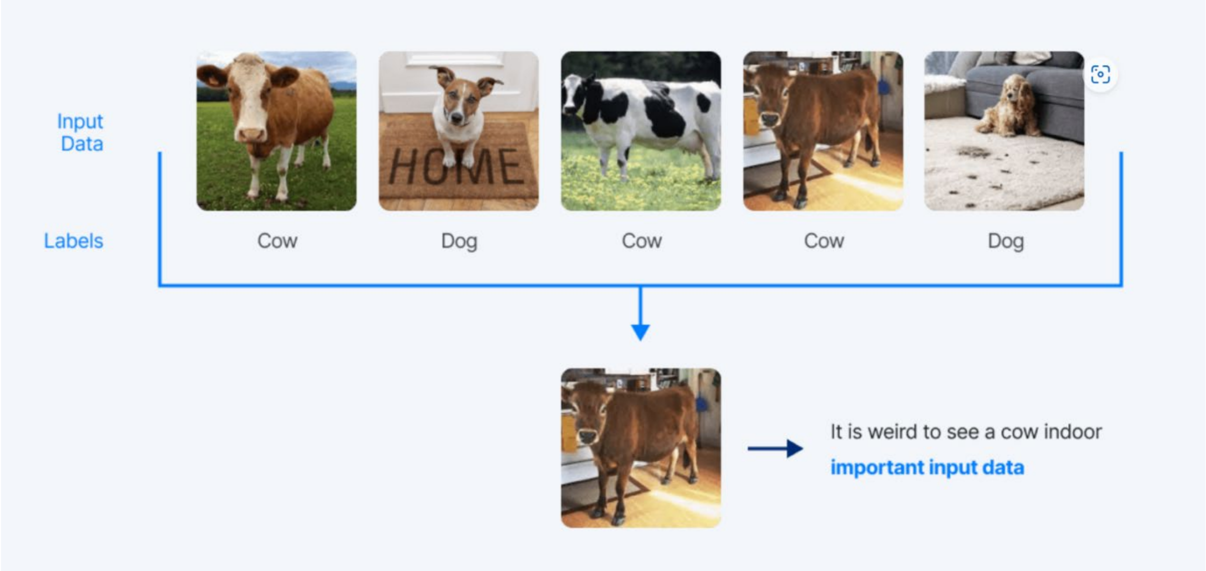


Figure 2 special training data [4]

1.2. How do coresets benefit machine learning?

1.2.1. Training efficiency

Due to the reduction of the dataset size, it makes the training process more efficient, since the model trains and operates with a smaller size. [4,5]

1.2.2. Memory usage

Lower memory requirements since smaller data set size, and it is helpful for resource constraints environments. [4,5,6]

1.2.3. Robustness of the model

Using coresets to train can help to improve the robustness of the model, coresets selection removes the redundant and noisy data in the original data set. [4,5]

The main goal of choosing a coreset for Self-Supervised learning (SSL) task is to use the distribution mismatch of the open set and the target data set, and the objective is to find a coreset that has the minimum distance to the target dataset using minimal spaces.

2. Methdology

2.1. ResNet-50

ResNet-50, a deep Residual Network model, is a prominent member of the ResNet family. Introduced by Kaiming He et al. from Microsoft Research in 2015, it has found widespread application in diverse computer vision tasks including image classification, object detection, and image segmentation. [1]

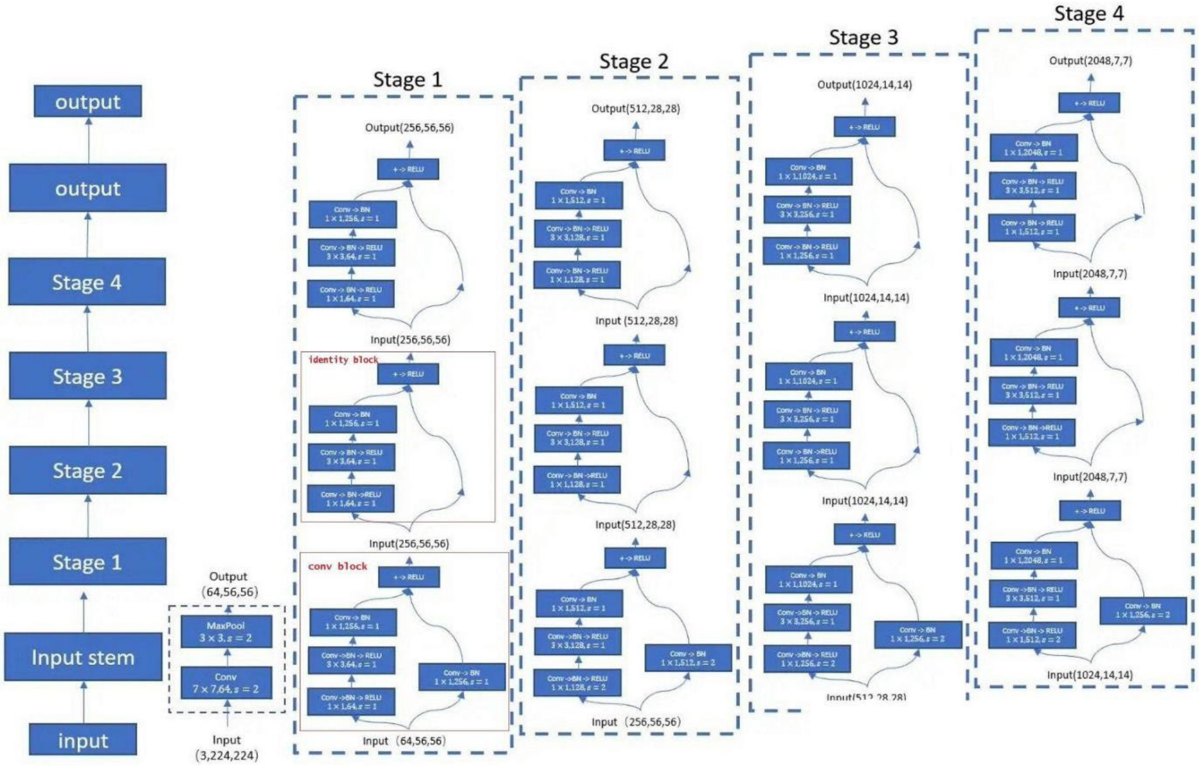


Figure 3 Resnet-50 structure [1]

2.1.1. Convolutional Block and Identity Block

ResNet-50 contains two basic parts: the Convolutional Block and the Identity Block.

Convolutional Block: The Convolutional Block is formed by multiple numbers of convolutional layers in order to learn the features of an image. Its typical structure includes:

- 1x1 Convolutional layer: Used to reduce and restore the number of channels, thereby reducing computational complexity and maintaining the dimensionality of the feature map. [2]
- 3x3 Convolutional layer: Used for feature learning, in the middle of two 1x1 convolutional layers. [2]

It is usually used at the beginning of the network to extract low-level features from the image.

Identity Block: The Identity Block also consists of three convolutional layers, where the first and third layers are 1x1 convolutional layers, with the middle layer being 3x3, similar to the Convolutional Block.

The input and output of the Identity Block maintain consistent dimensions, with a skip connection utilized to directly incorporate the input into the output, thereby preserving the original input information.

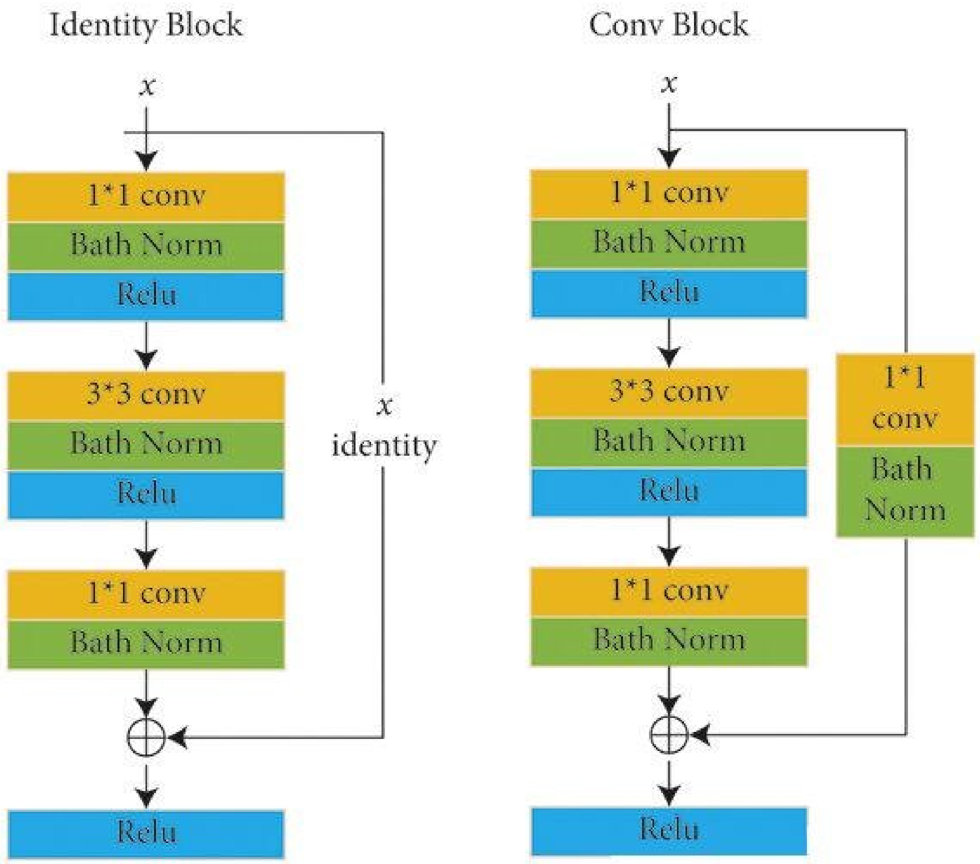


Figure 4 Structure of Identity Block and Convolutional Block

2.1.2. Batch Normalization Layers

Batch Normalization layers integrate with ResNet-50, it is a type of regularization technique that is being commonly used to be a stimulator and accelerate the training process of the neural network and improve the performance of the model. In ResNet-50, such a layer is often used between activation functions and convolutional layers. By normalizing the inputs to each mini batch to result in an expectation of approximately 0 and a variance close to 1. Which improves the stability of the network by effectively solving the problem of abnormal exploding of gradients. [1,2]

- Calculate the mean and variance of the mini-batch inputs for each channel.
- Using the calculated mean and variance to normalize the mini-batch inputs.
- Scale and shift the normalized inputs.
- Finally, apply a non-linear transformation to the resultant inputs using the activation function. [1,2]

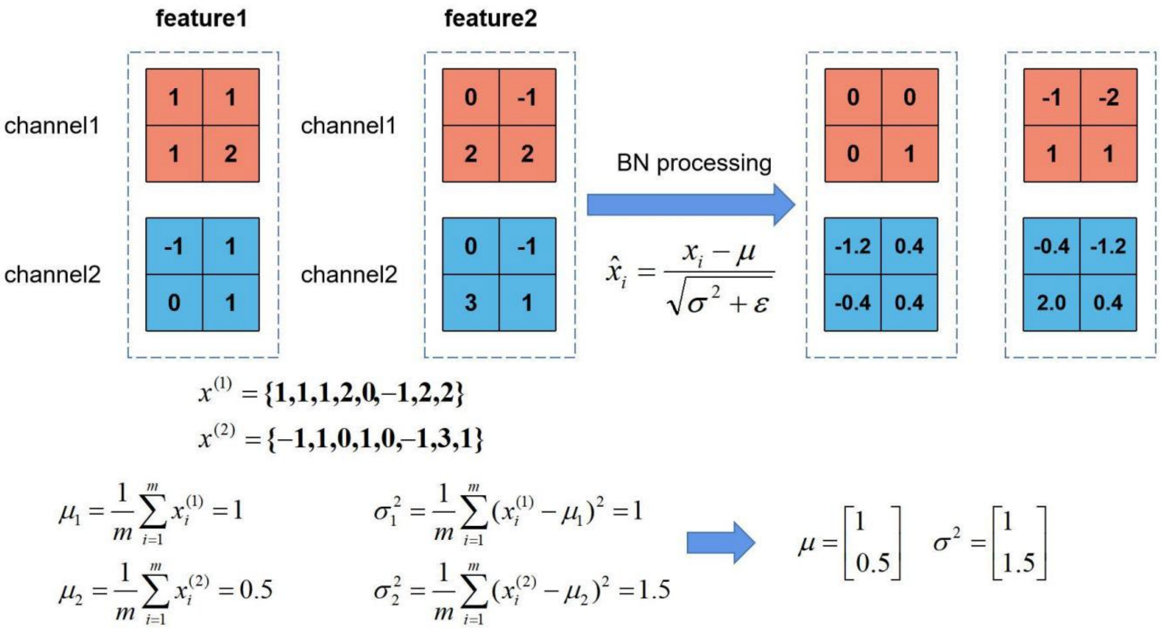

feature1
feature2
channel1
channel2
channel1
channel2
1 1
1 2
-1 1
0 1
0 -1
2 2
0 -1
3 1
BN processing
$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}$
0 0
0 1
-1.2 0.4
-0.4 0.4
-1 -2
1 1
-0.4 -1.2
2.0 0.4
$x^{(1)} = \{1,1,1,2,0,-1,2,2\}$
$x^{(2)} = \{-1,1,0,1,0,-1,3,1\}$
$\mu_1 = \frac{1}{m}\sum_{i=1}^{m} x_i^{(1)} = 1$
$\sigma_1^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i^{(1)} - \mu_1)^2 = 1$
$\mu_2 = \frac{1}{m}\sum_{i=1}^{m} x_i^{(2)} = 0.5$
$\sigma_2^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i^{(2)} - \mu_2)^2 = 1.5$
$\mu = \begin{bmatrix} 1 \\ 0.5 \end{bmatrix}$
$\sigma^2 = \begin{bmatrix} 1 \\ 1.5 \end{bmatrix}$

Figure 5 Logic of Batch Normalization Layers

2.2. *SimCLR Principles*

2.2.1. Contrastive Learning

Contrastive learning entails acquiring knowledge through the comparison of similarities and disparities among diverse instances. Within this framework, input data is categorized into distinct groups (positive and negative sample pairs), and features are either extracted or classifications are conducted by evaluating the distinctions between samples.

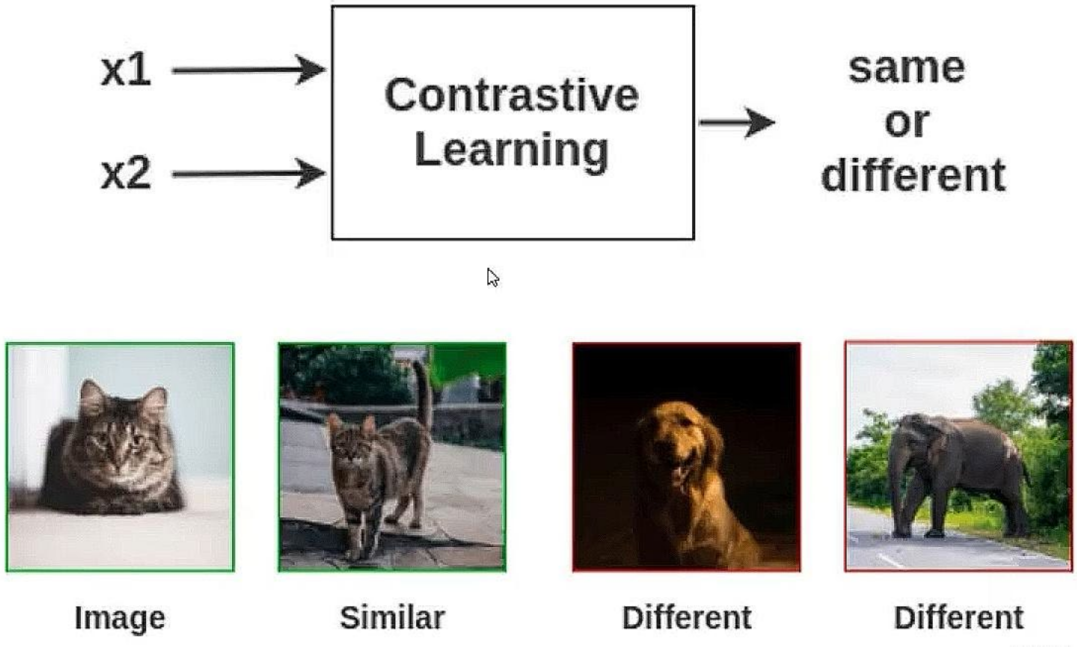


Figure 6 Contrastive learning

2.2.2. Sample Similarity
There are various approaches to contrastive learning, with distance metric-based methods being the most prevalent. These methods utilize distance functions, such as Euclidean distance or cosine similarity, to quantify the similarity between two instances. By computing distances between instances, we can identify the most similar or dissimilar instances and subsequently perform tasks such as feature selection, similarity matching, or classification. (Optimally, positive sample pairs should exhibit high similarity while negative sample pairs should demonstrate low similarity.)

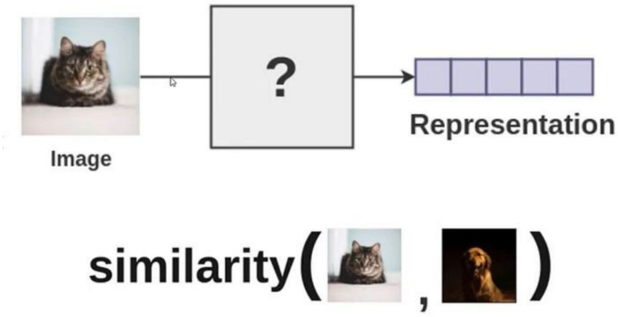


Figure 7 How contrastive learning works

2.2.3. Steps in SimCLR: Contrastive learning, through metric learning, provides the ability to extract features effectively.

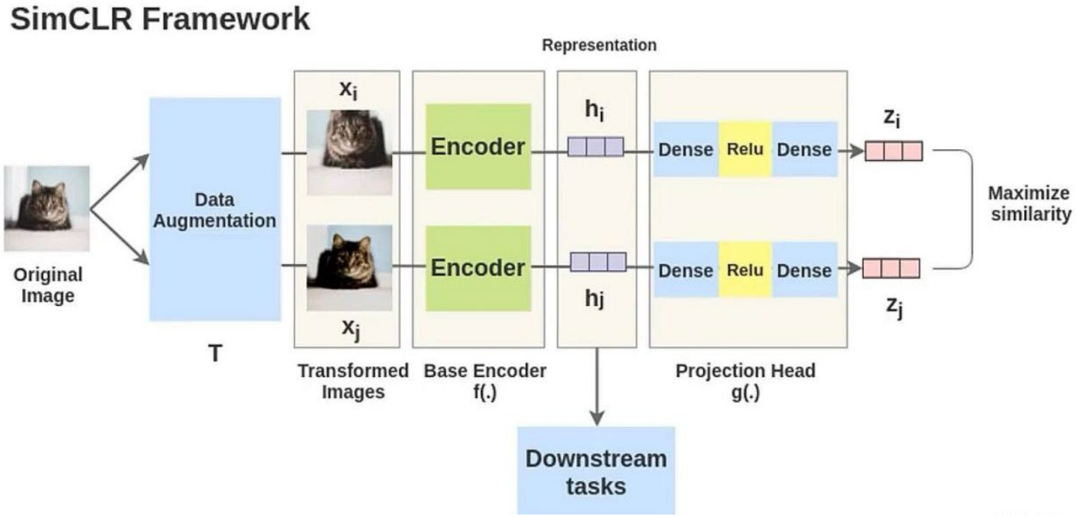


Figure 8 Logic of contrastive learning [4,5]

1. Select an Input Image: Apply two distinct data augmentations to the same image, creating a positive sample pair; different images are considered as negative sample pairs.

2. Prepare Two Random Image Augmentations: Implement augmentations such as rotation, color/saturation/brightness adjustments, scaling, cropping, etc. The paper delves into the spectrum of augmentations in depth and evaluates their effectiveness. (For constructing positive samples: image SimCLR - data augmentation, text SimCSE - dropout, image-text CLIP - image-text pairs)

3. Feature Extraction: Utilize a deep neural network (preferably a convolutional neural network such as ViT, BERT, or ResNet-50) to acquire feature representations (embeddings) of the augmented images.

4. Feature Projection: Use a small fully connected linear neural network to project the embeddings into another vector space.

5. Compute Loss: Calculate contrastive loss and propaga through the two networks. When the projections from the same image are similar, the contrastive loss decreases. The similarity between projections can be arbitrary; here, cosine similarity is used, as in the paper.

6. Downstream Tasks: Use the encoder obtained from contrastive learning as a feature extractor and fine-tune it based on the downstream task dataset.

[8,7]

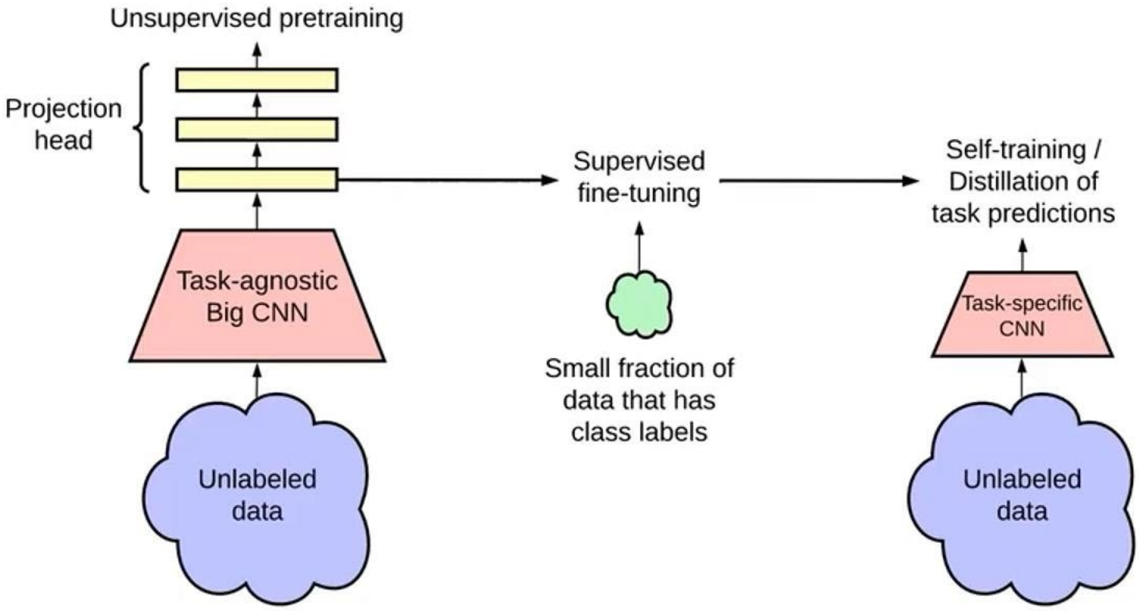


Figure 9 Unsupervised learning, Contrastive learning

2.2.4. Requirements:

- A substantial volume of data and a sizable batch size (8192) are utilized.
- The generation of positive and negative sample pairs does not necessitate labeling.

2.2.5. Designing the Loss Function:
The numerator represents the similarity within the same class (distance between positive samples), while the denominator signifies the similarity across different classes (distance between negative sample pairs). The parameter \(t \) denotes the temperature (scaling < 1) and is utilized to adjust the ratio.

2.3. Coreset selection method

2.3.1. Sampling method
A type of simple but useful coreset sampling algorithm is being used in the paper, the algorithm is named SimCore. SimCore finds a subset that shares the most similar semantics with the target set with the following equation, [3]

$$f(S) \;=\; \underset{x \in X}{\Sigma\, max}\; \omega(x,u), where\ S \subseteq U, U \cap X \;=\; \emptyset(2)$$

Target dataset: ***X***
Open set: ***u***

Here, ***w(x,u)*** represents the similarity between ***x*** and ***u***, their dot product being calculated for their normalized features. Also using K-means clustering to create centroids $\hat{X}$ from the target dataset ***X***. which saves more computational resource [3]

2.3.2. Iterative sampling

- In each iteration, select a subset ***S*** from ***u*** that maximizes the objective function ***$\hat{f}$(S),*** also using $\hat{X}$ instead of ***X*** to decrease the computational complexity.

- Exclude the selected subset $\boldsymbol{S}$ from the open set $\boldsymbol{u}$ after each iteration, and stop the iteration when ideal coreset size is reached. An ideal coreset can be determined by testing the performance of the selected coreset.

2.4. OpenSSL

2.4.1. OpenSSL Problem Formulation

Self-Supervised Learning (SSL) represents a pioneering approach for acquiring the intrinsic properties of data while filtering out irrelevant signals through the discrimination of perturbed samples. Recent research has advocated for the utilization of a contrastive loss function to promote similarity in representations from the same image and dissimilarity in representations from different images [14, 28, 79].

Given an input dataset $X = \{x_i\}_{i=1}^{N}$ $X = \{x\acute{i}\}$,we generate two copies of randomly augmented images,denoted as $A(X) = \{\tilde{x}_i\}_{i=1}^{2N}$, where$\tilde{x}_i$ and $\tilde{x}_{N+i}$form an augmented pair. Let E_θ be an encoder network. Using these augmented pairs, the contrastive loss can be formulated using the formula: $L(X; E_\theta) = \frac{1}{2|X|}$ $\sum_{\tilde{x}_i \in A(X)} l_{ssl} \,(z_i, z_i^+; \{z_i^-\})$, where $z_i = E_\theta(\tilde{x}_i), z_i^+$ $z_i = E_\theta(\tilde{x}_i), z_i^+$ represents the augmented pair of z_i^+ and far from $\{z_i^-\}$.

Projection heads or predictors are frequently used to more accurately assess the similarities of these representations, therefore enhancing the learning process. This method is essential in propelling semi-supervised learning forward, empowering models to acquire meaningful features without reliance on labeled data. In addition to contrastive approaches, non-contrastive techniques have also been investigated, further expanding the breadth and potential of semi-supervised learning.

2.4.2. Non-Contrastive Techniques

In Self-Supervised Learning (SSL), non-contraintive techniques focus on strategies for acquiring valuable representations, and not relying on the comparisons between the negative pairs and positive pairs.

Momentum Contrast (MoCo)

Momentum Contrast (MoCo), it is a type of useful non-contrastive technique, which harnesses a momentum-based encoder to preserve a running average of preceding encoder weights. This stabilizes the training process and refine the quality of the representations. Rather than depending on contrastive loss to build positive instances, MoCo establishes an expansion of negative examples. The momentum encoder yields consistent and stable representations by appending the weights with a moving average, thereby bolstering the model's discriminative power across disparate samples. This method has evinced exceptional performance across diverse benchmarks, affirming that non-contrastive methodologies can secure competitive or even superior outcomes relative to conventional contrastive strategies.

SwAV (Swapping Assignments between Views)

SwAV introduces a clustering-based approach to semi-supervised learning (SSL), aiming to assign each image to a cluster in a codebook. In contrast to contrastive methods relying on positive and negative pairs, SwAV utilizes a memory bank of cluster assignments and learns to predict the cluster assignment of an image under different augmentations. This method effectively captures the data distribution and encourages the model to learn robust and invariant features. Leveraging clustering techniques, SwAV proves particularly effective in handling large datasets with diverse features.

BYOL (Bootstrap Your Own Latent)

BYOL introduces an innovative approach by completely eliminating the necessity for negative samples. Instead of depending on contrastive loss, BYOL utilizes two neural networks: a target network and an online network. The online network generates representations, while the target network offers a slowly updated version of the online network's weights. The loss function in BYOL promotes the similarity between the representations of the online network and the target network without requiring negative samples. This method has demonstrated robust performance across various datasets, indicating that it effectively circumvents the contrastive nature of SSL.

3. Results

3.1. Target Datasets

The evaluation of SimCore utilized 11 target datasets, which are Aircraft, Cars, Pets, Birds, Dogs, Flowers, Action, Indoor, Textures, Faces, and Food.

To measure the quality of the representation of the SimCore, the paper first conducted linear probing, which is a common method of testing the performance of a representation algorithm. By comparing the results of SimCore and random sampling in all eleven fine-grained datasets, the paper reaches the first conclusion that SimCore outperformed random sampling. Furthermore, the paper compared the situation where only one cluster centroid of the target dataset was used and the situation where multiple cluster centroids of the target dataset was used. It proved that multiple cluster centroids are more advantageous than only one cluster centroid. However, this is under the condition that the representation method is the same. Simcore using one cluster centroid is still better than random sampling with multiple cluster centroid.

Notably, the varying trend observed in the target datasets provides insight into the ideal coreset size, which is determined by the degree of mismatch in distribution with the open-set. For instance, in datasets such as Pet and Birds, the pretraining of the OS proved to be highly effective. Additionally, in these datasets, SimCore successfully u-tilized the substantial budget allocation. This suggests that certain target datasets necessitate a larger number of coreset samples compared to others. Nevertheless, in practical terms, it is not feasible to determine the ideal budget size in advance due to our limited understanding of an uncurated open set. Hence, it is imperative that we manage SimCore with a well-defined stopping criterion. Remarkably, the utilization of SimCore with a stopping criterion significantly enhances the accuracy, in contrast to the X pretraining. This represents a significantly higher increase in performance when compared to the extensive OS pretraining and random sampling. The reason for this is that SimCore intelligently selects an appropriate size of coreset, and this size varies depending on the specific dataset being targeted.

SimCore was implemented on various architectures, namely EfficientNet-B0, ResNet18, ResNeXt50, and ResNet101. Irrespective of the size of the encoder architecture, SimCore significantly enhances pretraining on the target dataset. In addition, the paper conducted experiments using different SSL methods. SimCore consistently illustrates the impact of combining the coreset samples, even when using the latest autoencoder-based semi-supervised learning (SSL) techniques.

3.2. Open-Set

The paper has been utilizing the ImageNet-1k benchmark as the open-set. This dataset is carefully curated and encompasses various general domain. Utilizing a coreset extracted from ImageNet has demonstrated significant improvements in the performance of the desired tasks. However, in reality, an open-set is not aligned with our existing knowledge; instead, it is a collection of data that is randomly selected from the web or a database. The paper then demonstrates that SimCore significantly enhances pretraining by effectively identifying a well-matched coreset, even in open-set scenarios using three additional open-sets: MS COCO, iNaturalist 2021-mini, and Places365.

SimCore consistently achieves better results compared to pretraining without OS. However, the performance of SimCore is influenced by the open-set. When it comes to datasets that are not naturally diverse, such as Action and Indoor, iNaturalist is not as efficient as ImageNet. However, it does provide a useful subset of data for Birds. As anticipated, Places365 provides the most superior coreset for indoor settings compared to other open-sets due to its extensive inclusion of scenic semantics.

The Places365 open-set for the Pet target dataset has shown surprising improvements, particularly in identifying animal images. SimCore has successfully identified animal images, despite the assigned labels representing locations. The iNaturalist coreset comprises organisms either domesticated by humans or found indoors, which may be useful for capturing action scenes involving humans engaged in activities such as photography, fishing, or gardening.

To showcase the impact of SimCore in a more authentic setting, tests were conducted using uncurated open-sets. Initially, a merged dataset comprising all four open-sets was used to replicate a larger and more diverse open-set scenario. Web-crawled image datasets obtained through queries based on ImageNet classes and images obtained through queries based on Aircraft+Cars+Birds classes were used. SimCore sampled a smaller portion of the actual crawled set for each target, demonstrating its ability to generate valuable subsets of data even when the original dataset is not well-curated or contains unknown elements.

The analysis of the distribution of features using Gaussian kernel density estimation in R2 was used to examine how the SimCore algorithm selects the coreset in the latent space. The findings showed that SimCore effectively selects instances that are closely intertwined with the target data. By comparing the occupied areas of OS and X, it was confirmed that the open-set distribution is similar to each target dataset.

The SimCore algorithm was used to determine which instances from the open-set were actually sampled and then visualized the results. In the Pet dataset, SimCore with one cluster centroid primarily selected animal images, but also included some data that was not directly related to cats and dogs. In contrast, SimCore with a sample size of 100 with predominantly cat or dog images included up to 20 classes, each representing a specific breed of either a cat or a dog.

In the Birds dataset, SimCore with one cluster centroid collected other images, while SimCore with a hundred cluster centroids only collected data on the top-20 bird species.

3.3. Downstream Tasks

The paper assessed various downstream tasks, including k nearest neighbor (kNN) classification, semi-supervised learning, object detection, and multi-attribute classification. The results showed that SimCore has the highest accuracy in all tasks for most datasets.

4. Conclusion

The paper tested and proved that SimCore, their novel algorithm for extracting a sample from the dataset, makes pre-training more effective. The reasoning behind it and the steps that lead to the conclusion is unfolded through various perspectives. The evaluation is performed via target datasets, open sets, qualitative evaluations, and downstream tasks.

The coreset selection algorithm SimCore developed is successful on improving the efficiency of machine learning. Based on the experiments both we did and data given by the KAIST researchers, it has showed us the level of improvements it had made.

References

[1] Durga, B. K., & Rajesh, V. (2022). A ResNet deep learning based facial recognition design for future multimedia applications. Computers & Structures, 6-8. https://doi.org/10.1016/j.compstruc.2022.106095

[2] Representation learning (no date) Papers With Code. Available at: https://paperswithcode.com/task/representation-learning (Accessed: 21 July 2024).

[3] Papers with code - paper tables with annotated results for Coreset sampling from open-set for fine-grained self-supervised learning (no date) The latest in Machine Learning. Available at: https://paperswithcode.com/paper/coreset-sampling-from-open-set-for-fine/review/(Accessed: 21 July 2024).

[4] Coresets.org. (n.d.). Coresets explained And how does it work? | coresets.org. https://coresets.org/

[5] Jubran, I., Maalouf, A., & Feldman, D. (2019, October 19). Introduction to Coresets: Accurate coresets. arXiv.org. https://arxiv.org/abs/1910.08707

[6] Suslov, E. (2024, February 29). What are resource constraints and how to manage them? PPM Express. https://ppm.express/blog/resource-

constraints/#:~:text=The%20main%20types%20of%20resourcing%20constraints%20are%3A%201,resources%20required%20for%20a%20project.%20. . .%20More%20items

[7] Ikotun, A. M., Ezugwu, A. E., Abualigah, L., Abuhaija, B., & Heming, J. (2023). K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. Information Sciences, 622, 178–210. https://doi.org/10.1016/j.ins.2022.11.139

[8] Papers with Code - Contrastive Learning. (n.d.). https://paperswithcode.com/task/contrastive-learning