ADDRESSING COLD-START IN ACTIVE LEARNING

Ayush Shrivastava, Khush Shah, Satyam Bhardwaj, Vipul Baghel
IIT Gandhinagar
{shrivastavaayush, khush.shah, bhardwajsatyam, baghelvipul}@iitgn.ac.in

ABSTRACT

Active learning (AL) aims to identify valuable samples in an unlabeled pool to build a training dataset with minimal annotation costs [1]. Traditional methods start with partially or initially labeled samples for active selection and query annotations through iterations, known as a "Warm Start". However, the challenge arises in a "Cold Start" when no labeled data is available initially. Cold-start active learning lacks the advantage of leveraging information from annotations or feedback, making sample selection more challenging [2]. This necessitates a distinct sampling strategy.

We investigate a specific approach using the self-supervised methodology to choose initial training points from an unlabeled data pool - "Active Sampling". Active Sampling employs a contrastive technique to derive meaningful representations of data points, followed by clustering with methods like K-means and BIRCH. The most informative samples from these clusters are given to the oracle for labeling. This method provides a more effective starting point than random selection for the initial training set [2]. Our experiments encompass three datasets—MNIST, CIFAR-10, and a subset of ImageNet—and we achieved positive results across all datasets using the approach.

***K*eywords** Machine Learning · Active Learning · Image Classification · Self-supervised Learning

1 Introduction

Using deep neural networks (DNNs) in applications often requires a large amount of labeled samples for effective training [3]. The annotation cost, particularly in domains such as medical imaging, Natural Language Processing (NLP), and Remote Sensing, is often substantial [4]. Active learning (AL) is a promising solution to reduce annotation costs while maintaining DNN performance. Traditional AL methods, such as uncertainty sampling and committee-based algorithms, struggle in deep learning scenarios due to the iterative nature of sample selection and the reliance on DNNs, which perform poorly with a small training set. To make traditional AL methods work in deep learning scenarios, a large number of samples with initial annotations are often needed for the initialization of AL models. In addition, in the iterative way of traditional AL methods, the sample selection process must be suspended several times until newly selected samples are annotated, which is not practical in many real-world applications.[5] To address these issues, the concept of cold-start AL was introduced, where all valuable samples are selected at once without the need for an initial labeled set.

Our main contributions to this project are as follows:

- Implement the active sampling strategy to mitigate the cold-start problem, in which no labeled samples are given. [2]
- Train various neural-net based feature extractors on MNIST, CIFAR10 and ImageNet dataset, using self-supervised learning techniques such as Autoencoders and Contrastive learning.

2 Existing Works

There are various complex supervised learning tasks such as speech recognition, information extraction, classification, filtering, etc, in which data points' label acquisition is difficult, time-consuming, and highly cost-ineffective. Active learning methodologies address the challenge of labeling bottlenecks by soliciting queries in the form of unlabeled

instances, subsequently annotated by an oracle, often a human annotator. This strategic approach seeks to optimize accuracy with a minimal number of labeled instances, effectively mitigating the financial burden associated with acquiring labeled data [1].

There are three types of active learning approaches:

- Membership Query Synthesis
- Stream-Based Selective Sampling
- Pool-based Active Learning

Membership Query Synthesis, an active learning strategy, involves generating queries based on specific membership criteria, with the model creating new queries from scratch rather than relying on existing ones. Stream-Based Selective Sampling, operating in a streaming data scenario, selectively samples data from a continuous stream based on perceived informativeness. Pool-based Active Learning, accessing a pool of unlabeled instances, optimizes the labeling process by selecting instances for enhanced model performance. Various methodologies have been employed to devise effective query strategies. Uncertainty sampling and committee-based algorithms are predominant. Uncertainty sampling relies on heightened uncertainty in samples, utilizing metrics like least-margin, least confidence, and max entropy for gauging uncertainty. Its simplicity and computational efficiency contribute to widespread adoption, especially in the era of deep learning. An alternative strategy involves committee-based algorithms, leveraging the disagreement among multiple classifiers. This methodology assesses the divergence for each unlabeled sample, selecting those with the most substantial divergence scores. The underlying assumption is that querying annotations for samples with inconsistent predictions across classifiers yields valuable information [1].

However, the above-mentioned conventional AL approaches (categorized as warm-start paradigms) require a finite number of labeled instances to begin the AL. Earlier, a few labeled samples were sufficient to initialize warm-start Active Learning (AL) models because feature descriptors are already manually extracted, requiring only the fitting of the inductive bias. In contrast to this, present models rely on deep neural networks that are end-to-end trainable, incorporating a unified framework where feature extraction and inductive bias are seamlessly integrated. In order to mitigate this problem, Jin et. al. [2] proposed an approach named as Cold-start AL for image classification task, which consists of a contrastive self-supervised feature extractor, a hierarchical clustering module, and a sampling module based on information density. They tested their approach on CIFAR-10, CIFAR-100 and CALTECH-256 datasets. In this work, we implement and validate their approach on a much simpler dataset such as MNIST, and a much larger and complex ImageNet dataset.

3 Methodology

3.1 Learning Representation

The effectiveness of Active sampling heavily relies on the quality of feature representation. In cold-start active learning, where initially labeled samples are unavailable for learning feature representation, we employ a contrastive self-supervised learning algorithm. Contrastive self-supervised learning stands out as an advanced unsupervised model for feature representation [12]. The core idea is to enhance consistency between similar samples (positive sample pairs) while minimizing consistency between dissimilar samples (negative sample pairs). To achieve this, an encoder network is employed to extract features that learn a robust representation of images via a "learning comparison" in the feature space. Specifically, we employ various data augmentations on the same image to form positive sample pairs and different images to create negative sample pairs.

We have also explored simpler representation learning strategies such as using Vanilla autoencoders [13] and Encoders with triplet loss[14].

3.2 Clustering Algorithms

3.2.1 K-Means Clustering

K-means [16] is an algorithm used to classify objects into clusters automatically. The algorithm identifies k centroids and then allocates data points to the nearest cluster. First, k distinct instances are chosen randomly from the dataset and used as the initial centroids.

3.2.2 BIRCH Clustering

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) [15]is an algorithm used for hierarchical clustering. It is particularly suited for large datasets due to its incremental and memory-efficient properties. The algorithm builds a tree structure (Clustering Feature tree) with enough information to cluster data without holding all data in memory. BIRCH starts by loading a portion of the data into memory and preprocesses it to remove noise and outliers. BIRCH builds a Clustering Feature (CF) Tree, a hierarchical data structure where each node contains the data summary in its subclusters.

3.3 Active Sampling

We create m clusters, where m represents the number of initial data points to be sampled from the unlabeled dataset. The core concept of Active Sampling is to select the sample with the highest information content. The ideal sample should best represent the potential distribution of a cluster, typically found in the densest area of the latent space. Information content for each sample in a cluster is computed, and the one with the highest information content is chosen from each cluster. The information content is determined using cosine similarity:

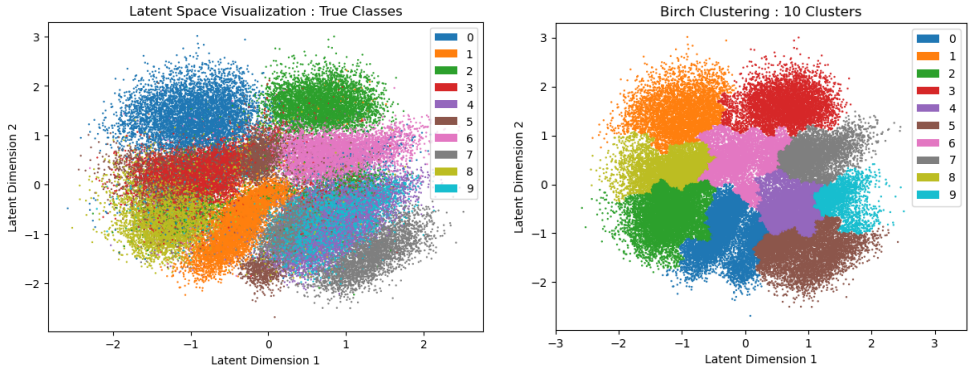
$$\text{sim}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\langle \mathbf{x_1}, \mathbf{x_2} \rangle}{\|\mathbf{x_1}\| \|\mathbf{x_2}\|}$$

The similarity of each point with every other point in the cluster is calculated, and the information content for any point is the mean of its similarity with every other point. A higher information content indicates greater similarity to the rest of the data in the cluster. The most informative point from each cluster is selected to form the initial dataset.

4 Experiments

4.1 MNIST

The MNIST dataset contains 60,000 training and 10,000 testing images, and images are all 28×28 pixels in size and are stored in a standardized format[9]. We used a simple Variational Autoencoder (VAE) having hidden layers of 64 and 32 with an output layer having four nodes. We learned a two-dimensional latent representation of the MNIST dataset. We trained VAE with keeping $\beta = 0$ and added a `torch.nn.BatchNorm1d` at the end in the Encoder to ensure the latent vectors are centered around the origin with unit variance. The batch size was kept at 64 for all training purposes. The representations learned from the AutoEncoder are shown in the Figure 1.

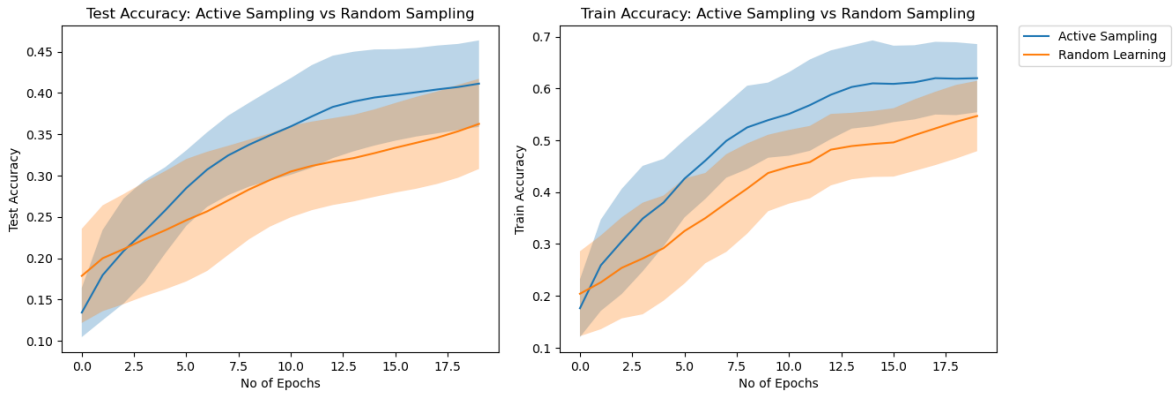


(a) True classes.

(b) BIRCH cluster labels (10 clusters)

Figure 1: $2-$dimensional representations learned on MNIST using $\beta-$VAE with $\beta = 0$ and BatchNorm.

We pick the most informative sample from each cluster and form a train-set. We compare the performance on train-set obtained from the Active sampling algorithm against random train-sets. A random train-set is obtained by sampling random points from the entire data-set. We ran 20 trials where we obtained a new random train-set while the Active Learning train-set was kept fixed. Plot (a) and (b) of Figure 2 display the Test and Train Accuracy as the number epoch increases. We were able to achieve a 5% accuracy boost while using the Active Sampling method over a randomly sampled trainset.



(a) Performance on Test-set (b) Performance on Train-set

Figure 2: Test accuracy vs number of epochs for 100 initial labeled samples on MNIST dataset.

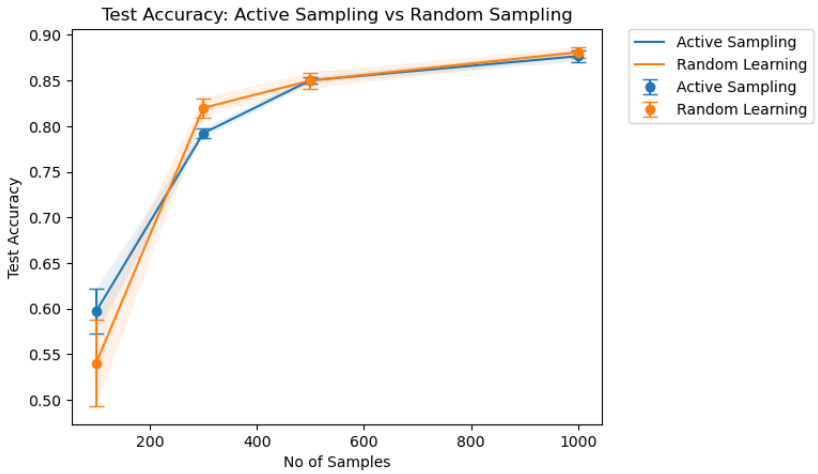


Figure 3: Number of samples vs test accuracy for MNIST dataset

4.2 CIFAR-10

The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 distinct classes, with 6,000 images per class, for a total of 50,000 training images and 10,000 testing images[10].

For this dataset, we explored a simpler self-supervised learning task defined by the triplet loss, which is given by

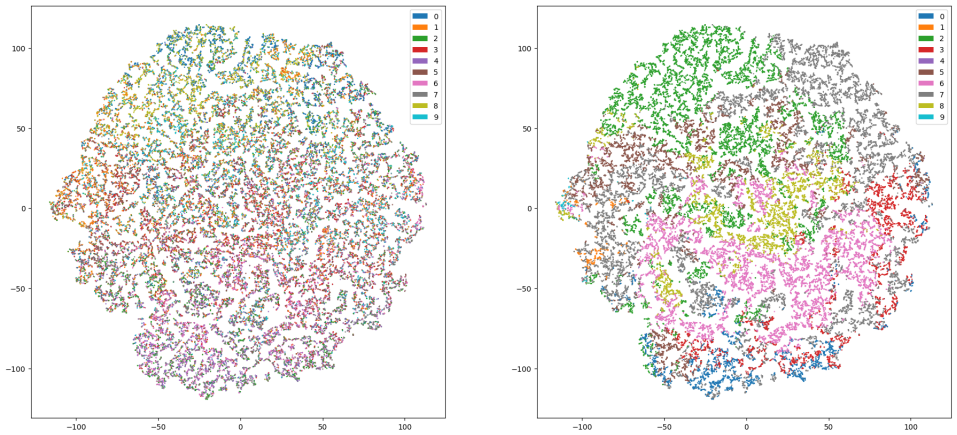
$$\mathcal{L}(A, P, N) = \text{RELU}\left(\|\text{f}(A) - \text{f}(P)\|_2 - \|\text{f}(A) - \text{f}(N)\|_2 + \alpha\right)$$

where A is the anchor input, P is a positive input of the same class obtained by augmenting the anchor input with a suitable augmentation, N is a negative input. In case of self-supervised learning, we take any other data point from the training set as the negative input. α is the margin that defines by how much we want to separate two different classes and f is the Encoder model providing the latent representation for a datapoint. In addition to this loss, we enforced the latents to have unit L_2 norm. This constrains the latents to lie on the surface of a d-dimensional sphere and discourages the latents from either shrinking to zero or blowing up. We kept $d = 96$ as our latent dimension.

We trained the model for 600 epochs with a learning rate of $1e-4$ using the Adam optimizer. From Figure (a) we can see that the model is able to separate the classes to some extent. Due to lack of time, we were not able to test this representation for Active sampling.

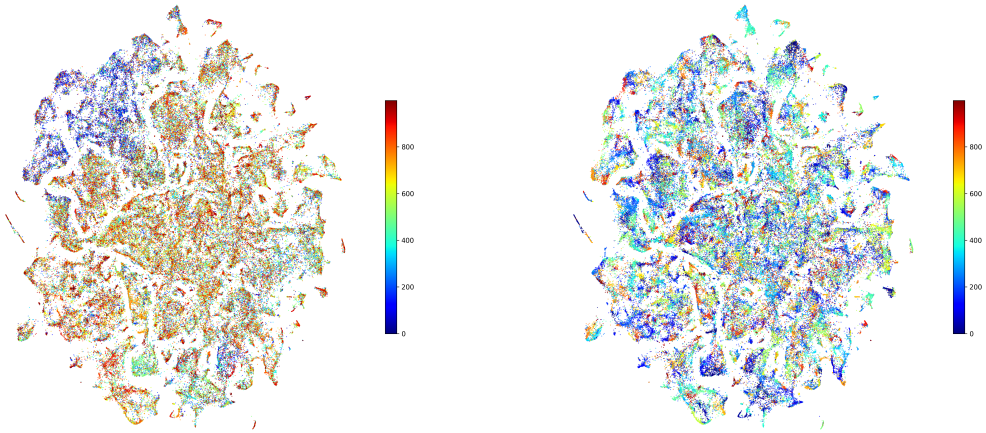
4.3 ImageNet

ImageNet [6] is a large dataset of natural images containing 1000 classes with 1.2M images in the training set. Due to our limited compute, we work on a class-balanced subset of the training set containing 125,000 images (125 images



(a) True labels (b) BIRCH cluster labels

Figure 4: t-SNE visualization of the latents vectors learned for CIFAR-10 using Autoencoder with Triplet loss.



(a) True labels (b) BIRCH cluster labels

Figure 5: t-SNE plot of latent representations in ImageNet-125k pool set.

per class). We consider this as our unlabelled pool set, henceforth referred to as ImageNet-125k. For testing our approach, we create a class-balanced subset of 5,000 images (5 images per class), which we call ImageNetTest-5k. Each image was first resized to 256×256, center-cropped to 224×224 and then standardized using the mean and standard deviation of the entire ImageNet training set. This ensures compatibility with the pre-trained MoCo v2 model, based on a ResNet-50 backbone, used to obtain the representations [11]. The model was trained for 800 epochs on ImageNet-1M. Model weights were provided by the authors on the GitHub repository for the paper [11]. We freeze the weights of the featurizer and modify the final layer of the network to output logits over 1000 classes. This is our classifier network for ImageNet.

To visualize the latent representations learned by MoCo v2 via self-supervised contrastive learning, we perform t-SNE with on the ImageNet-125k as shown in Figure 5(a). We used the `tsnecuda` library with default values i.e. perplexity of 50 and learning rate of 200 running for 1000 iterations. We can see some local structure with some classes grouped quite well, while others are mixed.

To begin the process of Active sampling, we first cluster the latent representations into $N \in \{1000, 2000\}$ clusters and then pick the most informative sample from each cluster as described previously in 3.3. Figure 5(b) shows the t-SNE plot with labels assigned by the BIRCH clustering algorithm.

Figure 6 shows the gain in test accuracy obtained by using Active sampling vs Random sampling for the initial train set of sizes $N \in \{1000, 2000\}$. We repeat the training 20 times to obtain better estimates of the mean and standard

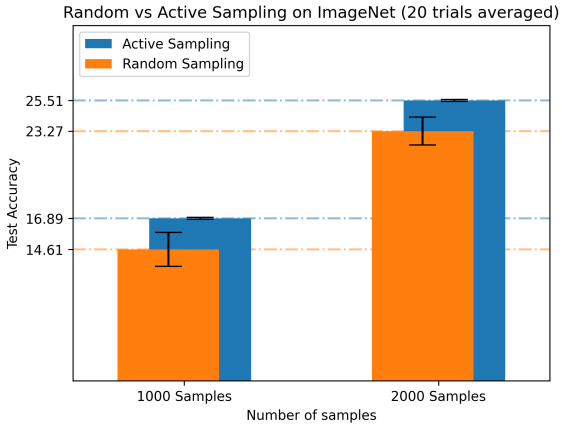


Figure 6: Active Sampling vs Random Sampling

Table 1: Summary of results obtained on the ImageNet-125k dataset

	Number of samples	Mean accuracy	Standard deviation
Random sampling	1000	14.613	0.624
Active sampling	1000	16.895	0.0373
Random sampling	2000	23.274	0.509
Active sampling	2000	25.514	0.0358

deviation in the test accuracy. In case of random sampling, in each iteration we reinitialize the random initial set. However, since the Active sampling is deterministic once the latent vectors are fixed for a fixed N, we use the same initial train set for each iteration.

Table 1 shows the mean and standard deviation of the test accuracy for Active and Random sampling with $N \in \{1000, 2000\}$, trained 20 times. We can observe a gain of $\sim 2.2\%$ improvement in the test accuracy.

5 Conclusion

We have explored the problem of Cold-start in Active Learning and implemented one promising approach for image classification based on representations learned using contrastive self-supervised learning followed by clustering. On the MNIST dataset, we observed a gain of $\sim 5\%$ accuracy on the test set for $N = 100$ samples, while the advantage decreases as we increase the number of samples to $N = 1000$. We hypothesize this is due to the number of samples being drastically larger than the number of true classes, making it highly probable that random sampling obtains a good class-balanced initial set. On the ImageNet dataset, we observe a gain of $\sim 2.2\%$ improvement in the test accuracy. Thus, we have validated the approach of [2] on both small toyish datasets and large real-world datasets.

While there is a modest improvement in the test accuracy, there is a trade-off between the annotation budget and computational budget. Training models on large unlabelled pool of data using contrastive self-supervised learning requires a lot of compute power, and in most cases it might be preferrable to go with random sampling. However, for scenarios in Medical imaging where annotation costs may be extremely high, it becomes feasible. In future work, we can also explore the Bayesian techniques to train effectively a small-scale neural network for discriminative feature representation.

Acknowledgments

We extend our sincere gratitude to the CVIG Lab and CNP Lab for providing essential computational resources that significantly contributed to the successful completion of this project. The availability of high-end GPU machines was pivotal in training such large models and gathering valuable data for this study.

References

[1] Burr Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison. 2009.

[2] Jin, Qiuye, et al. “Cold-Start Active Learning for Image Classification.” Information Sciences, vol. 616, Nov. 2022, pp. 16–36. DOI.org (Crossref), https://doi.org/10.1016/j.ins.2022.10.066.

[3] Sarker, Iqbal H. “Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions.” SN Computer Science, vol. 2, no. 6, Aug. 2021, p. 420. Springer Link, https://doi.org/10.1007/s42979-021-00815-1.

[4] Mall, Pawan Kumar, et al. “A Comprehensive Review of Deep Neural Networks for Medical Image Processing: Recent Developments and Future Opportunities.” Healthcare Analytics, vol. 4, Dec. 2023, p. 100216. ScienceDirect, https://doi.org/10.1016/j.health.2023.100216.

[5] Zheng, Hao, et al. “Biomedical Image Segmentation via Representative Annotation.” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, July 2019, pp. 5901–08. ojs.aaai.org, https://doi.org/10.1609/aaai.v33i01.33015901.

[6] Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

[7] Pedregosa, Fabian, et al. “Scikit-Learn: Machine Learning in Python.” Journal of Machine Learning Research, vol. 12, no. 85, 2011, pp. 2825–30. www.jmlr.org, http://jmlr.org/papers/v12/pedregosa11a.html.

[8] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. https://doi.org/10.48550/ARXIV.1912.01703

[9] Deng, L. (2012). The mnist database of handwritten digit images for machine learning research. IEEE Signal Processing Magazine, 29(6), 141–142.

[10] CIFAR-10 and CIFAR-100 Datasets. https://www.cs.toronto.edu/ kriz/cifar.html. Accessed 20 Nov. 2023.

[11] Chen, X., Fan, H., Girshick, R., He, K. (2020). Improved baselines with momentum contrastive learning. https://doi.org/10.48550/ARXIV.2003.04297

[12] Chen, Ting, et al. Big Self-Supervised Models Are Strong Semi-Supervised Learners. arXiv:2006.10029, arXiv, 25 Oct. 2020. arXiv.org, https://doi.org/10.48550/arXiv.2006.10029.

[13] Hinton, G. E., and R. R. Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks.” Science, vol. 313, no. 5786, July 2006, pp. 504–07. DOI.org (Crossref), https://doi.org/10.1126/science.1127647.

[14] Hoffer, Elad, and Nir Ailon. Deep Metric Learning Using Triplet Network. arXiv:1412.6622, arXiv, 4 Dec. 2018. arXiv.org, https://doi.org/10.48550/arXiv.1412.6622.

[15] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. 1996. BIRCH: an efficient data clustering method for very large databases. SIGMOD Rec. 25, 2 (June 1996), 103–114. https://doi.org/10.1145/235968.233324

[16] Ahmed, Mohiuddin, Raihan Seraj, and Syed Mohammed Shamsul Islam. 2020. "The k-means Algorithm: A Comprehensive Survey and Performance Evaluation" Electronics 9, no. 8: 1295. https://doi.org/10.3390/electronics9081295