Dataset Reduction Algorithm Based on Deep Features Clustering

Jian Tang[1,a,*], Hua Cheng[2,b]

[1]*Information Engineering University, Zhengzhou, China*[*]
[2]*Jiangnan Institute of Computing Technology, Wuxi, China*
[a]*tjjackson@qq.com*, [b]*chenghua@ncic.ac.cn*
Corresponding Author: Jian Tang, tjjackson@qq.com

ABSTRACT

The dataset reduction algorithm is to obtain the simplified dataset through compressing original dataset based on a strategy and rule. It ensures to reduce the training set size as much as possible, without changing generalization ability of the simplified dataset. To study the algorithm with better reduction effect for deep learning image dataset, the paper introduces prototype selection, feature extraction and other concepts in dataset reduction; proposes the dataset reduction algorithm combined with deep feature extraction framework and clustering; and realizes effective reduction for deep learning image dataset. The experimental results indicate that the proposed reduction algorithm can effectively improve generalization of the image dataset with better effect than other similar reduction algorithms. Besides, the paper verifies validity of the algorithm.

Keywords: *Dataset reduction algorithm, Prototype selection, Deep features, Clustering*

1. INTRODUCTION

At present, there are more and more data acquisition equipment of each field, resulting in exponential increase of data size. As a result, higher requirements are proposed for storage capacity and data processing capacity of the computer system, and how to collect, manage and utilize data has been a major topic in the field of information technology. The data analysis technology with representative machine learning and data mining will greatly promote transformation from data to knowledge.

In 2006, the paper [10] firstly proposed the deep learning concept. They tried to provide more essential representation for data through multi-hidden-layer artificial neuron network structure and realize multi-layer abstraction for data through the concept of hierarchical abstraction. As an important branch of machine learning, the deep learning gradually achieves major breakthrough in the field of natural language processing, image recognition, speech recognition, etc. The image classification competition - ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is taken for an example. From 2012, emerging neural network models (e.g. AlexNet [13], GoogleNet [18], ResNet [9], etc.) successively achieved remarkable scores [17] in the competition, which resulted in wide application of the deep learning in the field of image recognition. Besides, the network model can be trained better due to public dataset (e.g. ImageNet [17], CIFAR [12], etc.), while the model trained with massive data has a strong generalization capacity and can better adapt to dataset learning required for actual application to improve the classification effect.

Generally, the deep learning includes training stage and inference stage. The neural network accepts different types of datasets and optimizes own network weight parameters under unsupervised condition to learn features of a mode [1], which is called as “training stage”. The trained model can be used for predicting unknown data or obtaining the target mode from data, which is called as “inference stage”. In the process, the dataset on the one hand provides information source for neural network and on the other hand verifies training effect in the inference test, and therefore plays a foundation role in the deep learning.

As above mentioned, although development of deep learning correspondingly improves the capacity of solving problems in the field of scientific research, the deep learning also brings new challenges while providing opportunities. In the big data era, the demand for deep learning becomes more urgent due to widespread data,

A. El-Hashash et al. (Eds.): IEIT 2022, ACSR 100, pp. 948–955, 2023.
https://doi.org/10.2991/978-94-6463-058-9_148

and rapid increase of data size also results in excessive temporal and spatial consumption of the deep learning algorithm. GPT-3 [22] is taken for an example. In addition to surprising 175 billion parameters, it has training data size even up to 45TB. Under such surprising number of model parameters and training data size, it is conservatively estimated that the single training costs of GPT-3 vary from one million to ten million dollars, including cost due to hardware hashrate and overhead of manual annotation and data check. The larger the dataset size is, the more the training costs will be. In the future, the field of deep learning will be full of competition in training hashrate and data quality.

With increasingly massive dataset, it is extremely important to simplify the training dataset for deep learning. On the one hand, the decrease of dataset size can reduce the demand of training hashrate. On the other hand, removing redundant and noise samples in the dataset can improve inference accuracy of the trained model. In the actual application of deep learning, the efficiency of training or inference algorithm (i.e. spatial and time complexity of deep learning algorithm) may be as important as the final capacity of accuracy prediction of the mathematical model [1]. As a result, the dataset reduction algorithm becomes a research hotspot in the field of machine learning.

2. STATUS QUO OF DOMESTIC AND FOREIGN RESEARCH

In the machine learning, the dataset reduction algorithm is to is to obtain the simplified dataset *S* through compressing the original dataset *T* based on a strategy and rule. It ensures to reduce the training set size as much as possible, without changing generalization ability of the simplified dataset [21]. Before application of the dataset reduction algorithm in the dataset training stage, the dataset is pre-processed to reduce the temporal and spatial complexity of the deep learning algorithm in learning and training. The dataset consists of several instances, where the representative instance is the prototype. Based on the generation of new prototype, the dataset reduction algorithm can be roughly classified into prototype selection and prototype generation [4]. The reduction algorithm of prototype selection is to select a part of instances from original dataset *T* as elements of the packed dataset *S* in order to reduce the size of original dataset. The reduction algorithm of prototype generation starts from the original dataset *T* and generates a small amount of instance data to form the packed dataset *S* in order to simplify the dataset size. It is difficult to identify attributes of the generated prototype from visual effect in the image sample, so that the paper mainly focuses on the prototype selection algorithm.

Under the prototype selection algorithm, CNN (Condensed nearest neighbor) [8] and ENN (Edited nearest neighbor) [20] are early algorithms based on the nearest neighbor rules and many subsequent algorithms are improved based on both algorithms.

CNN randomly selects samples from *T* to initialize *S* firstly; then randomly selects any sample to be learned from *T* and find out and obtain its nearest samples from *S*; if they are same in classification, it indicates that the sample to be learned has representative prototype and will not be included in *S*, otherwise, the sample to be learned is included in *S* as a new prototype, until all samples in *T* are learned, and the algorithm is ended. Although CNN is susceptible to random initialization, reading order, abnormal sample, missing deletion mechanism, no consideration of sample's distribution feature and other defects, it is widely used and improved for exploration because it can be realized easily and support incremental learning. For example, FCNN (Fast condensed nearest neighbor) [2] focuses on reducing sensitivity of reading order of samples and obtaining classification decision boundary prototype; GCNN (Generalized condensed nearest neighbor) [5] introduces the same nearest neighbor and conquers CNN's shortcoming of using same classification of nearest neighbors only; MNV (Mutual neighborhood value) [6] uses mutual nearest neighbor value to reduce the sensitivity of samples reading order; RNN (Reduced nearest neighbor rule) [7] focuses on improving CNN's shortcoming of being incapable of deleting added prototype when constructing the simplified dataset; the classification-boundary sample selection algorithm based on clustering strategy includes IKNN (Improved k-nearest neighbor classification) [23] and PSC (Prototype selection by clustering) [15], etc. The above algorithms are still sensitive to the noise.

The editing nearest neighbor algorithm (also "editing algorithm") aims to select representative points through removing noise points and clearing sample points in different classification of overlay regions. The editing algorithm has a high reduction effect for many training sets with noise. ENN is the earliest editing algorithm, which was proposed based on the idea of removing noise and expanding class boundary.

Processing procedure of ENN: The classification information of any sample in the training set is obtained through KNN firstly to determine that the actual classification of the sample is consistent with the computation classification; if they are different, the sample is deleted. Repeating above steps to obtain deeper boundary clearing effect, which is the Repeated ENN [19]. However, ENN excessively depends on distribution of training set, measurement criterion of nearest neighbor, and the nearest sample, and changes boundary status of the training set. Based on ENN, NearMiss [14] proposed the concept of large and small classification and used it for sample editing on the uneven class boundary.

3. DATASET REDUCTION ALGORITHM BASED ON DEEP FEATURES CLUSTERING

For the image classification problem in deep learning, the reduction algorithm selects a processing method that combines with feature extraction and prototype selection. While removing redundant samples in the dataset as much as possible, the algorithm expects to ensure no significant decrease of generalization capacity of the classification network or even realize improvement. Through describing basic idea of the deep feature clustering-based algorithm, realization of the algorithm itself, design of termination conditions, and analysis of algorithm complexity, the paper proposes a new dataset reduction algorithm and verifies its effectiveness and performance in the subsequent experiment.

3.1. Deep Features

For a long time, the reduction algorithm is a difficult problem, because there is not universal measurement method for various applications and industries. For different data types, various feature extraction and feature selection methods spring up. However, the manually designed feature extraction method undoubtedly will be influenced by vision and perception of the algorithm designer. With the emergence of machine learning, the neural network-based unsupervised learning feature extraction method attracts more and more attentions.

In recent several years, the deep learning technologies with the representative CNN (Convolutional Neural Network) [13] have achieved many achievements far better than common machine learning algorithms in the field of computer vision. For the image data, advanced features in the image can be extracted with CNN and then used for dataset reduction in the next step. Generally, one or more convolutional pooling layers can be seen in CNN, which can help CNN greatly reduce the image data size and effectively utilize the two-dimensional structure of input data.

However, the training difficulty also increases with the number of layers of CNN. When the network is deepened to a certain level, deeper network will result in worse training effect. ResNet can respond to the network degradation problem better through jump connection of internal residual blocks. Therefore, the paper selects ResNet as the deep feature extraction model.

3.2. Clustering Selection

Different algorithms have different opinions with regard to retain and delete different types of data points. Some reduction algorithms of prototype selection consider that internal nodes will not directly influence the class boundary if the position information has any change (e.g. movement or deletion) due to far distance from the class boundary. Some algorithms even delete all internal nodes to obtain extremely high reduction rate.

The paper considers that data points in the dataset shall be scattered appropriately but shall not be deleted completely. In deep learning, the weight change of internal samples on the neural network plays a major role and gives rough functions to the model. If many or even all data points are deleted, training effect of the model will be greatly reduced, which is opposite from the dataset reduction target. Besides, the nodes adjacent to the boundary shall be retained to the largest extent to improve generalization accuracy of the model.

Therefore, the dataset can be processed as per class with the clustering method. From the view of sample distribution, there are more internal samples and few class boundary samples. As a result, there are more internal clusters and few external clusters from the clustering algorithm. For each cluster, internal samples can be selected through retaining samples adjacent to the cluster center, i.e. to scatter the sample space.

In terms of clustering, the paper selects K-means [3] as the clustering algorithm. For the algorithm, the basic idea is to select k number of sample points as initial clustering center and classify all sample points into different clusters through iteration to minimize the distance between objects in the cluster, but the object distance between clusters is large.

The parameter k in the algorithm is defined in advance, and k number of data objects are selected randomly from the sample point as the initial clustering center. Steps of the algorithm:

1) Randomly select k number of objects from n number of data objects as the initial clustering center, i.e. center of each cluster;

2) Calculate the Euclidean distance between each object and above center samples based on the mean value of clustering samples in each cluster; re-classify the above samples based on the minimum distance and give the nearest cluster to each sample again;

3) Re-calculate mean value of the cluster (center object) in case of any change of the cluster;

4) Repeat Step (2) & (3), until the iterations are satisfied, or each cluster will not have any change;

5) Obtain center samples of k number of clusters and attributes of all samples in the k number of clusters after end of the algorithm.

Based on determination for attribution cluster of each sample and the distance from computation sample to the cluster center, it is easy to determine the samples to be retained through quantitative results sorting and estimate the importance of a sample.

3.3. Algorithm Design

As above mentioned, the information that a data point is important (i.e. the datapoint can be replaced by other data points in the same cluster) is determined by its position in the dataset space.

The sample in dataset is denoted as X_{ij}, where i is the sample class in original dataset; j is serial number of the sample in the class. For example, the sample X_{ij} is extracted for deep features, unfolded into one-dimensional sample and then normalized to obtain DX_{ij}. After clustering operation with similar samples, the clustering center C_{ik} can be obtained, where k is the cluster serial number of the clustering center. The importance of the sample point depends on the distance with clustering center C_{ik}, i.e. $E(DX_{ij}, C_{ik})$. If the sample point X_{ij} has the nearest distance with the clustering center C_{ik} in the same cluster k, it is considered that the sample point is necessary; if the number of samples in the cluster is larger than 1 and the sample point X_{ij} does not has the nearest distance with the clustering center C_{ik}, then it is considered that X_{ij} is not important.

It is easy to find that there are more clusters in the sample space and more samples in the cluster; there are few clusters for the boundary and few sample points for the cluster, and there is one sample point only. This is consistent with design concept of the algorithm, i.e. to reduce density of internal samples and training costs, retain boundary samples and generalization capacity of the neural network.

To realize above theories and definitions, the algorithm is described as below:

Process of the algorithm:

1) Select a ResNet model and denote it as Net (ResNet20 hereof) to make preparation for building deep feature extractor; randomly extract subset from T and train Net with applicable subset to obtain the trained model; remove Output layer of the model to obtain the deep feature extractor and denote it as Net.features().

2) Traverse the original dataset T, extract features of each sample X_{ij} to get Net.features(X_{ij}); unfold Net.features(X_{ij}) into one-dimensional form and make normalization, denoted as DX_{ij}.

3) Traverse each class i in the dataset T: The K-means clustering is conducted for all samples of the same class and the number of clusters in clustering is K. After clustering, k number of cluster centers and DX_{ijk} can be obtained, where the subscript k is the cluster serial number of the sample.

4) Traverse all cluster centers generated in clustering: If the number of samples in the cluster is 1, it indicates that there is no simplified sample in the cluster. The sample X_{ij} is included into the simplified dataset S based on DX_{ijk}. If the number of samples in the cluster is larger than 1, the Euclidean distance between all DX_{ijk} and the cluster center in the same cluster is calculated and sorting is conducted, the nearest samples are retained and the rest samples are deleted. The X_{ij} corresponding to the nearest sample DX_{ijk} is included into the simplified dataset S.

5) Traverse all classes in T to obtain the simplified dataset *S*.

The clustering step in the algorithm is an iterative process which will be terminated if the defined iterations are satisfied, or each cluster has no change. Among many performance evaluation indexes for the reduction algorithm, the reduction amplitude and accuracy rate are important parameters. If both are considered, performance of the reduction algorithm can be evaluated roughly. If based on the learning of simplified dataset S, the generalization accuracy after training will be closer to the original training set T, and even the former has higher classification accuracy than the latter, then performance of the reduction algorithm will be higher. For the compression amplitude, if the classification accuracy varies within acceptable range, the smaller the size of compression dataset S is, the better the performance of the reduction algorithm will be. Both are also uniform and satisfied for the reduction algorithm as much as possible. For the reduction rate, the paper provides an equation as below:

$$\mathrm{Cr} = \frac{\mathrm{Count(T)} - \mathrm{Count(S)}}{\mathrm{Count(T)}} = \frac{\sum_{i=1}^{I} K_i}{\sum_{i=1}^{I} \mathrm{Count}(X_i)} (\%) \tag{1}$$

where, K_i is the number of clusters in the i^{th} class sample; X_i is the entirety of the i^{th} class. It can be seen that the reduction rate of the algorithm is closely related to the number of clusters. In actual practice, the common elbow method [11] under the clustering algorithm is used to roughly estimate K_i value. But it is impossible to accurately determine the number of clusters. In addition, the elbow method will consume larger computations than the reduction algorithm itself in drawing the curve, while generalization capacity of the neural network fluctuates up and down within the small interval of K_i value, so that K_i entirety is usually taken with an appropriate estimation value.

Same with some clustering algorithms, the deep feature clustering-based dataset reduction algorithm mainly calculates its attribute to a cluster center at the sample point. It is assumed that n is the number of sample points of the dataset; k is the number of required clusters; t is the number of clustering iterations. Mostly, k can be obtained through multiplying by a coefficient smaller than 1 by n; t is a constant, i.e. time complexity of the algorithm is $O(n^2)$. Besides, the clustering algorithm will cluster each class for the dataset successively instead of acting on the entirety of the dataset every time, so that original calculations can be decreased substantially.

3.4. Performance Evaluation of the Algorithm

The paper selects common color image dataset (e.g. CIFAR-10/100 [12], ImageNet [17], etc.), where ImageNet selects the common ILSVRC2012 subset. CIFAR-10/100 is selected from Tiny images project [16], of which image size is small and image elements are single and can be distinguished easily. Both have same number of samples, but CIFAR-100 has 100 classes of samples and can be trained more difficult. In ILSVRC2012 dataset, images are collected from the network by the ImageNet project and have large size, clear effect and more objects, so that ILSVRC2012 is commonly considered a more challenging image dataset.

TABLE **1** describes summary of the dataset and lists full name of each dataset, sample space, image size, and number of classes.

TABLE 1 DESCRIPTION OF DATASET USED IN THE EXPERIMENT

Full name of dataset	Training set space	Image size	Classes Number
CIFAR-10	50000	32 x 32	10
CIFAR-100	50000	32 x 32	100
ILSVRC2012	1300 x 1000	high resolution	1000

In The experiment, the accuracy verification model uses the ResNet network with 20-layer depth. The experiment selects Intel x86 platform as the hardware environment, GTX1060 as GPU, v3.6 Python, and v1.14.0 TensorFlow. To show the average operating effect of the comparison algorithm, the mean value of 5-fold cross validation of five times as the final experimental result.

First, Fig. 1 to Fig. 3 show the reduction algorithm and comparison effects of random extraction in CIFAR-10, CIFAR-100 and ILSVRC2012 dataset. The experiment starts from the reduction rate 60% (i.e. K =Number of samples ×60%, the random selection is to randomly extract 60% samples) with 5% interval up to the whole set. It can be observed from above three Tables that the accuracy rate of reduction dataset is better than the randomly selected subset in most cases. Besides, it can be found from Fig. 1 to Fig. 3 that if the accuracy rate is 90% for CIFAR-10 and 95% for ILSVRC2012 under the proposed reduction algorithm, the accuracy rate is maximum and even exceeds the whole set; but protruding points cannot be seen on the icon for CIFAR-100 dataset. Based on attributes of the dataset, there are more samples and redundant images of each class for CIFAR-10 and ILSVRC2012, and there are more image classes and few image sample of each class for CIFAR-100, so that there are also few redundant images. The experimental result also fully demonstrates existence of redundant images in the dataset.

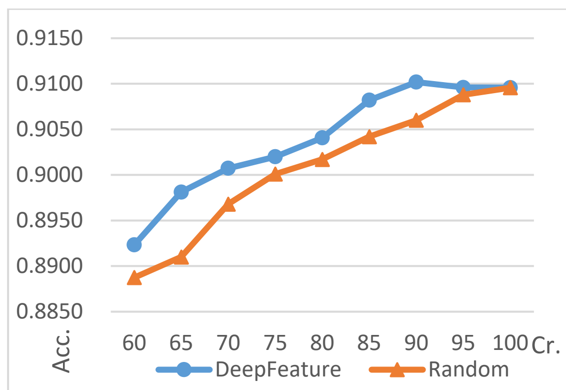

Fig. 1 Experimental result of random selection in CIFAR-10 and the deep feature clustering-based reduction algorithm

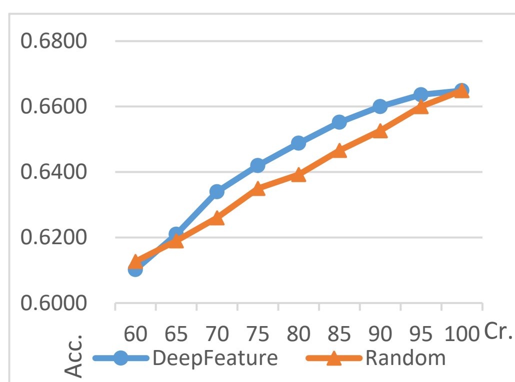

Fig. 2 Experimental result of random selection in CIFAR-100 and the deep feature clustering-based reduction algorithm

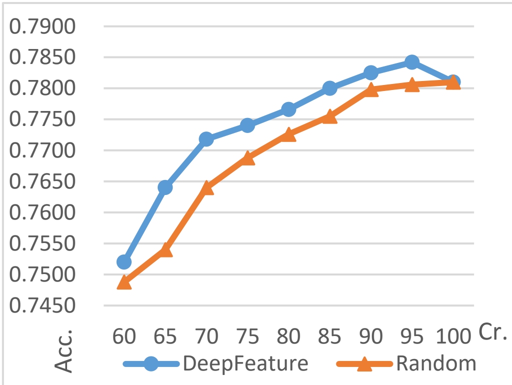

Fig. 3 Experimental result of random selection in ILSVRC2012 and the deep feature clustering-based reduction algorithm

The field of reduction algorithm lacks application of the deep learning image dataset. To evaluate the algorithm, the paper selects conventional CNN and ENN as comparison algorithms. Although CNN and ENN were proposed early, now they are still prevailing due to easy realization and significant effect. Based on both algorithms, a series of improved algorithms are proposed, for example, ReapeatedENN, NearMiss, TomekLinks, NCR and other common algorithms are also concluded in the range of comparison algorithms.

With regard to select K value in three datasets, on the one hand, the optimal values 90% and 95% are measured in CIFAR-10 and ILSVRC2012 in the previous group of experiment and can be considered as K value for calculation; on the other hand, the optimal value obtained with the elbow method under K-means roughly varies within the range of measurements, which demonstrates the experimental result. But K value can confirm the rough range only; if the percentage is further decreased, the accuracy rate tends to fluctuate. Therefore, K can have a better reduction effect within a range.

TABLE 2 EXPERIMENTAL RESULTS OF THE WHOLE SET, CNN, ENN, RepeatedENN, NearMiss, TomekLinks, NCR AND DEEP FEATURE CLUSTERING-BASED reduction ALGORITHM IN THREE GROUPS OF DATASETS. THE LEFT COLUMN SHOWS ACCURACY RATE, AND THE RIGHT COLUMN SHOWS reduction RATE.

	Original		CNN		ENN		RepeatedENN	
	Acc	Cr	Acc	Cr	Acc	Cr	Acc	Cr
CIFAR-10	90.96%	100.00%	77.93%	13.03%	90.70%	89.81%	89.90%	89.90%
CIFAR-100	65.63%	100.00%	51.22%	50.54%	55.57%	39.77%	54.32%	39.82%
ILSVRC2012	77.80%	100.00%	50.33%	60.33%	55.63%	78.43%	54.25%	75.43%

NCR		NearMiss		TomekLinks		DeepFeature		Random	
Acc	Cr	Acc	Cr	Acc	Cr	Acc	Cr	Acc	Cr
90.35%	90.35%	85.84%	98.80%	90.87%	90.87%	91.02%	90.00%	90.60%	90.00%
59.67%	91.28%	64.10%	97.62%	54.99%	91.86%	66.36%	95.00%	66.02%	95.00%
70.44%	89.34%	76.21%	98.80%	70.98%	90.77%	78.42%	95.00%	78.06%	95.00%

First, it can be seen from

TABLE **2** that the deep feature clustering algorithm is better in maintaining the classification accuracy of generalization of the reduction dataset than reduction algorithms (e.g. CNN, ENN, RepeatedENN, NearMiss, TomekLinks, NCR) and even original dataset, and maintains higher generalization level than original dataset in the CIFAR-10 and ILSVRC2012 dataset.

Besides, ENN, Repeated ENN and TomekLinks have better behaviors in CIFAR-10 dataset, and the simplified dataset achieves the generalization effect approximate to the original dataset. NearMiss has better reduction effect in CIFAR-100 dataset. Although CNN has a large reduction percentage, the calculation time and costs also exceed the acceptable range of reduction algorithm. It indicates that different algorithms have different application effect on different datasets. The deep feature clustering algorithm achieves better effect in three datasets, so that the method can be better applicable to the deep learning image dataset.

The algorithm also has limitations as below: First, the step of obtaining deep feature extractor itself will incur training costs. Based on different convolutional network of the extractor, the training costs may influence practicability of the algorithm, so that the existing trained convolutional network can be searched in actual practice. Second, the number of clusters K can be selected based on different datasets only and the hyperparameter is adjusted with related clustering algorithm. Different datasets are applicable to different K value. Third, according to the experimental measurement, if K value is further decreased to the range of reduction rate, accuracy rate of the model will fluctuate severely around the protruding points. It is inferred that the random initialization results in result fluctuation in the training process of neural network. Therefore, K value can bring good effect for the algorithm within a range. The future study can further explore rapid calculation algorithm of K value based on the clustering algorithm.

4. CONCLUSIONS

The paper proposes a dataset reduction algorithm based on deep features clustering and selects important samples through building deep feature extraction framework and prototype clustering. The experimental results indicate that the proposed algorithm can effectively extract redundant samples in the image dataset, save operation costs, and improve generalization capacity of the model. However, it is not convenient for the deep feature extraction framework to calculate costs and clustering parameters. Therefore, future research will consider other estimation methods for clustering parameters and explore more feature extraction methods to improve availability of the algorithm.

REFERENCES

[1] Alpaydin, E. (2020). Introduction to machine learning: MIT press.

[2] Angiulli, F. (2007). Fast nearest neighbor condensation for large data sets classification. IEEE Transactions on Knowledge and Data Engineering, 19(11), 1450-1464.

[3] Beckmann, N., Kriegel, H.-P., Schneider, R., & Seeger, B. (1990). The R*-tree: An efficient and robust access method for points and rectangles. Paper presented at the Proceedings of the 1990 ACM SIGMOD international conference on Management of data.

[4] Bezdek, J. C., & Kuncheva, L. I. (2001). Nearest prototype classifier designs: An experimental study. International journal of Intelligent systems, 16(12), 1445-1473.

[5] Chang, F., Lin, C.-C., Lu, C.-J., & Servedio, R. (2006). Adaptive Prototype Learning Algorithms: Theoretical and Experimental Studies. Journal of Machine Learning Research, 7(10).

[6] Garcia, S., Derrac, J., Cano, J., & Herrera, F. (2012). Prototype selection for nearest neighbor classification: Taxonomy and empirical study. IEEE transactions on pattern analysis and machine intelligence, 34(3), 417-435.

[7] Gates, G. (1972). The reduced nearest neighbor rule (corresp.). IEEE Transactions on Information Theory, 18(3), 431-433.

[8] Hart, P. (1968). The condensed nearest neighbor rule (corresp.). IEEE Transactions on Information Theory, 14(3), 515-516.

[9] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[10] Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. science, 313(5786), 504-507.

[11] Kolesnichenko, P. V., Zhang, Q., Zheng, C., Fuhrer, M. S., & Davis, J. A. (2021). Multidimensional analysis of excitonic spectra of monolayers of tungsten disulphide: toward computer-aided identification of structural and environmental perturbations of 2D materials. Machine Learning: Science and Technology, 2(2), 025021.

[12] Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.

[13] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 1097-1105.

[14] Mani, I., & Zhang, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. Paper presented at the Proceedings of workshop on learning from imbalanced datasets.

[15] Olvera-López, J. A., Carrasco-Ochoa, J. A., & Martínez-Trinidad, J. (2010). A new fast prototype selection method based on clustering. Pattern Analysis and Applications, 13(2), 131-141.

[16] Prabhu, V. U., & Birhane, A. (2020). Large image datasets: A pyrrhic win for computer vision? arXiv preprint arXiv:2006.16923.

[17] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Bernstein, M. (2015). Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3), 211-252.

[18] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. Paper presented at the Proceedings of the IEEE conference on computer vision and pattern recognition.

[19] Tomek, I. (1976). An Experiment with the Edited Nearest-Neighbor Rule.

[20] Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics(3), 408-421.

[21] Wilson, D. R., & Martinez, T. R. (2000). Reduction techniques for instance-based learning algorithms. Machine learning, 38(3), 257-286.

[22] Winata, G. I., Madotto, A., Lin, Z., Liu, R., Yosinski, J., & Fung, P. (2021). Language Models are Few-shot Multilingual Learners.

[23] Wu, Y., Ianakiev, K., & Govindaraju, V. (2002). Improved k-nearest neighbor classification. Pattern recognition, 35(10), 2311-2318.