The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM)
ISSN: 2602-3199

The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 2023

Volume 26, Pages 262-271

IConTES 2023: International Conference on Technology, Engineering and Science

Improve Image Classification Using Data Optimization

Djamel Berrabah
Djillali Liabes University

Yacine Gafour
Ibn Khaldoun University

Abstract: Image classification is a fundamental task in machine learning that involves assigning labels or classes to images based on their content. It is often performed using convolutional neural networks (CNNs). These networks are capable of learning and generalizing patterns from large amounts of data. However, if the data is not sufficiently voluminous, overfitting can occur. In such cases, it is recommended to turn to classical machine learning techniques. Moreover, the data that was insufficient for deep learning may exceed the processing capacity of the machine. This can pose significant challenges in terms of storage, memory availability, and computational power required to perform the learning operations. Our proposed approach involves addressing these challenges by optimizing the content of the dataset. This optimization is performed while preserving the essential information necessary for classification. Indeed, identical or highly similar are identified, grouped together and represented by the most representative one among them. At the same time, their sizes can be reduced. Furthermore, another significant challenge in our proposed approach revolves around managing class imbalances within the dataset. Our approach has been evaluated and the results are promising.

Keywords: Unsupervised linear/non-linear dimensionality reduction, data visualization technique unsupervised learning algorithm, dataset optimization

Introduction

In the contemporary digital age that defines our times, we find ourselves amidst a vast wealth of visual data, comprising a diverse array of images and graphical information. At the core of this visual content lies a fundamental challenge: how to accurately and efficiently extract meaningful information from these images? The answer to this question is pivotal for a multitude of applications, ranging from facial recognition to disease detection in medical X-rays, as well as the classification of objects within complex scenes. Over the past few decades, machine learning and in particular deep learning have revolutionized the way we approach image classification. Deep neural networks, such as convolutional networks (CNN), have achieved remarkable performance in image recognition. However, beyond the power of the algorithms, a crucial element has emerged: the quality and composition of the dataset on which these models are trained. Moreover image classification using deep learning models on small datasets can be a difficult task, since these models typically require large amounts of data to generalize effectively. Although there are solutions to overcome this problem, such as data augmentation, transfer learning, etc., the performance of the image classifier may not be as high as with sets of larger data. However, there are several techniques and strategies that can be used to improve image classifier performance even with limited data. By carefully applying these techniques, it is still possible to create a reasonably effective image classifier with a small dataset. Imagine you are

training a machine learning model using data. You can compare it to a student learning from examples. The more examples (data) this student has, and the more diverse they are, the better they can generalize and make accurate decisions. That's why the quality and variety of data are essential. To optimize the model, it's necessary to carefully select the data, eliminate low-quality data, and ensure that the training dataset is representative of the real-world situations.

In this paper, we will delve into an essential yet often overlooked aspect of image classification. We will explore why the composition of the dataset can impact the model's performance, as well as the efforts made by researchers and practitioners to enhance the accuracy, reliability, and generalization of results. We will discuss improving the quality of datasets and the tangible consequences of their optimization on model performance (Hinton & Salakhutdinov, 2006). We will demonstrate how subtle adjustments to the dataset can significantly affect an algorithm's ability to discern and categorize images with unparalleled precision. Beyond pixel nuances, dataset optimization also delves into semantic and contextual dimensions. This article will showcase sophisticated methods for preprocessing, data augmentation, and dataset cleaning deployed to address these challenges. This paper is organized as follow: In section 2, some dimensionality reduction algorithms, are briefly reviewed. The detailed description of the proposed approach can be found in Section 3. Section 4 presents experiments conducted to validate the efficacy of the proposed method, while Section 5 presents the conclusions and outlines future work.

Related Works

Keeping only the most important images or data points is a crucial step in dataset reduction. It helps to focus on the information that is relevant to the specific task, reducing noise and improving model efficiency. Representing data in a feature space that enhances class separability is fundamental in classification tasks. Feature engineering aims to transform the data into a format where different classes are distinguishable, making it easier for machine learning algorithms to make accurate predictions. The decision to keep or discard data should be a balance between data reduction and maintaining the essential information. It's essential to avoid overfitting (using too much data) and underfitting (using too little data) to ensure the model's generalization. The approach for dataset reduction and feature engineering can vary depending on the specific domain and problem. Different tasks may require different strategies. Overall, the statement emphasizes the importance of thoughtful data preprocessing, which can significantly impact the success of machine learning models, especially in classification tasks.

The field of dimensionality reduction offers a range of techniques, each with its strengths and applications (Dutta & Ghosh, 2016). t-distributed stochastic neighbor embedding (t-SNE) (van der Maaten & Hinton, 2008), independent component analysis (ICA) (Comon, 1994), and multidimensional scaling (MDS) (Cox & Cox, 1994) are indeed some of the well-established and widely used methods, but there are many others, including Isomap (Tenenbaum et al., 2000), LLE (Roweis & Saul, 2000), and autoencoders (Wang et al., 2020), to name a few. The choice of dimensionality reduction method often depends on the specific problem, the nature of the data, and the goals of the analysis. For instance, some methods are excellent for linear dimensionality reduction, while others are often preferred for visualizing high-dimensional data in a lower-dimensional space. In t-SNE, the main idea is to visualize and cluster high-dimensional data while preserving local similarities, making it effective for revealing data structure in lower dimensions (Zhang & Izquierdo, 2006). However, it can be computationally expensive and slow for very large datasets, limiting its practicality in such cases. Independent Component Analysis is commonly used in blind source separation, but it may not perform well when the underlying assumptions about source independence are violated in the data. One of its disadvantages is its sensitivity to these underlying assumptions, which may not always hold in real-world data, potentially leading to inaccurate results. Multidimensional Scaling is a technique for visualizing high-dimensional data while preserving data point distances, but it can be computationally demanding for large datasets. This computational intensity makes it less practical for very large datasets, potentially leading to longer processing times. Isomap, Locally Linear Embedding (LLE), and autoencoders are all dimensionality reduction techniques used to map high-dimensional data into lower-dimensional spaces. Isomap and LLE emphasize preserving local data structures, while autoencoders utilize neural networks to learn efficient representations. However, these methods may require careful tuning and can be sensitive to the choice of hyperparameters and the specific characteristics of the dataset, making them less straightforward for some applications. Some of these methods have been combined to create new techniques, including Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018). UMAP combines elements from Isomap and t-SNE and is recognized for its capability to capture both local and global structures within data. However, it is computationally intensive and

sensitive to hyperparameter choices, which can affect result quality and introduce non-deterministic behavior. Dimensionality reduction techniques discussed and so on, such as Isomap, LLE, t-SNE, and UMAP, have been used effectively for data visualization (Yousaf et al., 2020; Hajibabaee et al., 2021) and classification (Milošević et al., 2022; Shi et al., 2019) tasks. These techniques can provide encouraging results when applied to well-structured and low-noise data. However, as the statement correctly points out, their performance can be less reliable when confronted with noisy or real-world data, where the data may be less well-behaved or exhibit more complex structures. In such cases, additional data preprocessing or specialized methods may be required to mitigate the impact of noise and improve performance.

Proposed Approach

Large image datasets have become increasingly common across various fields, necessitating the reduction of image datasets to improve the efficiency and computational performance of image processing tasks. These tasks encompass activities like classification, object detection, and image retrieval. The primary objective when working with these datasets is to optimize dimensionality while retaining crucial information (as depicted in Figure 1). By reducing dimensionality, algorithms become more computationally efficient, resulting in faster and more manageable processing, all while preserving essential image details. A multitude of techniques for dimensionality reduction has been developed, and the choice of the most suitable method depends on the specific characteristics of the dataset and the desired outcome of the image processing task. Our approach serves as a preprocessing step, conducted prior to commencing the image classification process (Figure 1). In this paper, we introduce two distinct optimization approaches for reducing the image dataset: Horizontal and Vertical optimizations. These strategies are tailored to address the challenges posed by the dataset's volume

Figure 1. Image classification process using an optimized dataset

Vertical Optimization

Principal Component Analysis (PCA) and clustering are two distinct techniques commonly employed in data analysis and machine learning. They can be effectively used in conjunction to comprehend and structure data. PCA, as elucidated by Jolliffe and Cadima (2002), stands as a widely adopted linear dimensionality reduction method celebrated for its ease of implementation and versatile applications across various domains. PCA operates by harnessing eigenvectors to capture linear variations within high-dimensional data. It adeptly unveils the underlying lower-dimensional structure of data points distributed along or in proximity to a linear subspace. This is achieved by identifying linear combinations of the original variables, referred to as principal components, which encapsulate a significant portion of the data's variance (as depicted in Figure 1). As a result, PCA emerges as a potent technique for dimensionality reduction and the extraction of pivotal insights from extensive datasets."

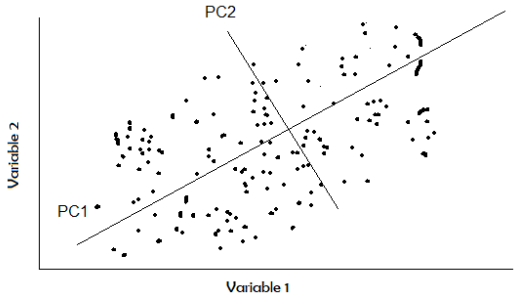

Figure 1. Generating PCA dimensions

In the context of image processing, an image can be regarded as a high-dimensional dataset, where each pixel represents a feature. When applying PCA to images, we treat them as matrices, with pixel intensity values corresponding to the matrix elements. The PCA algorithm then analyzes the covariance matrix of the image data and computes the associated eigenvectors and eigenvalues. These eigenvectors, often referred to as principal components, represent the directions of maximum variance in the image data, effectively forming a set of basis vectors spanning the image space. The corresponding eigenvalues indicate the significance or weight of each eigenvector in representing the image data.

By selecting a subset of principal components with the largest eigenvalues, we can effectively reduce the dimensionality of the image data while preserving the most significant information. In summary, PCA in image processing offers a way to reduce dimensionality while retaining essential information. The fundamental concept behind this technique is to decrease dataset dimensionality while preserving as much variability, both statistical and strategic information, as possible (Jolliffe & Cadima, 2002).

PCA can be used as a preprocessing step before applying clustering algorithms to data. A lower number of principal components that explain a significant portion of the variance are typically selected (e.g., 95% of the variance). The clustering algorithm is applied to the transformed data after reducing its dimensionality. Thus, the clustering algorithm will group similar data points based on the principal components instead of the original features (Figure 2). As a result, the results of clustering can be interpreted in the lower-dimensional space, which make it easier to understand the structure of data and the relationships between clusters. By combining PCA and clustering, it is possible to potentially improve the quality of clusters, reduce the impact of noise, and gain a better understanding of data.

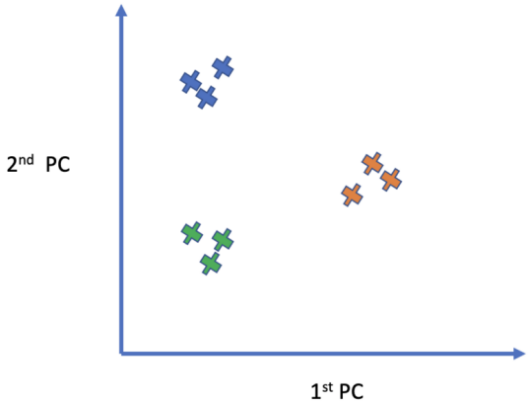

Figure 2. PCA clustering

Horizontal Optimization

K-means (Garbade & Michael, 2018) stands as a widely used unsupervised clustering technique with applications spanning various domains, including image processing and computer vision. Its significance is prominently visible in tasks like image segmentation and classification. The central objective of K-means is to partition a dataset into 'k' distinct groups or clusters, ensuring that images within the same cluster share similarities while those across different clusters exhibit dissimilarities (Figure 3). In the realm of K-means, each observation is represented as a point in an n-dimensional space, with 'n' signifying the number of descriptive variables or features. The algorithm's mission is clear: group similar data points and unveil concealed patterns in the dataset. This is achieved by identifying a fixed number of clusters ('k') within the dataset. The K-means algorithm embarks on this journey with an initial set of centroids, randomly selected as the starting points for each cluster. It then iteratively calculates to optimize the positions of these centroids.

Utilizing K-means clustering to eliminate redundant images presents a pragmatic method for mitigating data redundancy within an image dataset. First, relevant features are extracted from each image using Ho-LBP (Gafour et al., 2020). These features will serve as the basis for comparing and clustering images. Then, the K-means clustering algorithm is applied to the feature vectors extracted from the images. The number of clusters (K) chosen will

determine the level of image redundancy reduction. Within each cluster created by K-means, images that are similar to each other are considered to be potentially redundant or near-duplicates. Determining which images are redundant depend on the similarity threshold. Images with feature vector similarities that exceed this threshold are considered duplicates. Once duplicates are identified within clusters, redundant images are eliminated. Reducing the number of images in a dataset using K-means can be a challenging task. In practice, our objective is not the complete removal of redundant images but rather the reduction of dataset size. It's worth noting that eliminating all redundancy entails the deletion of a portion of the data, which could result in valuable information loss. This is especially critical when the deleted data plays a significant role in analysis or addresses class balancing concerns.

K-means Algorithm for Dataset optimization:

- **Feature Extraction**: Initially, we extract meaningful features from input images by employing the Ho-LBP descriptor.
- **Vectorize Images**: The extracted features are transformed into a suitable format for K-means clustering. Typically, each image is represented as a feature vector in a high-dimensional space based on these extracted features.
- **K-means Clustering**: Apply the K-means algorithm to cluster the feature vectors associated to images into 'k' clusters. The choice of 'k' depends on how many images are to be retained in the reduced dataset (or to eliminate from the input dataset).
- **Cluster Centroids**: After K-means clustering, 'k' cluster centroids are obtained. These centroids represent the most representative images in each cluster.
- **Cluster Labeling:** After clustering, each image is assigned to one of the 'K' clusters based on its similarity to the cluster's centroid.
- **Thresholding:** A similarity threshold is set to determine when two images are considered duplicates. Images with feature vector similarities exceeding this threshold are marked as duplicates.
- **Duplicate Detection:** Within each cluster, images that are very similar are likely duplicates or near-duplicates. Images within the same cluster are compared to identify and flag potential duplicates.
- **Selecting Representative Images**: To reduce the dataset, one (or more) image(s) can be chosen from each cluster to represent that cluster. These selected images will form the reduced dataset.
- **Eliminate Duplicates:** The elimination process concerns images having probable duplicates or near-duplicates. This elimination is done according to the degree of similarity. the images with the greatest degree of similarity will be the first candidates to be eliminated.

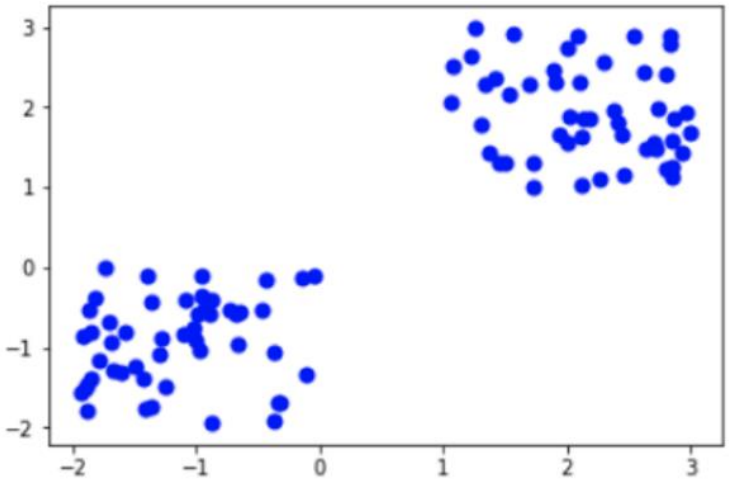

Figure 3. How data is displayed on two-dimensional space

It is necessary to know that the K-means algorithm is applied in our approach in a somewhat specific way. In its original version, the algorithm attempts to organize the dataset into a specified number of clusters. But Image Datasets are already organized into clusters (classes). So what is the role of this algorithm in our proposal approach?

As said above, our aim is the deduplication of similar images, more precisely, those containing the same information or almost. In this case, k-means is not applied globally but in a particular way. This means that the same class is divided into clusters to eliminate redundant images.

Hybrid Solution

In the preceding subsections, we explored two approaches to dataset optimization: horizontal and vertical. As previously discussed, the central challenge lies in optimizing the dataset while preserving vital strategic information. Both Principal Component Analysis (PCA) and K-means are powerful techniques for dataset optimization, each with its own unique strengths and applications.

In horizontal optimization, PCA takes center stage. Its primary role is to reduce data dimensionality, effectively reducing the number of variables in a dataset while retaining critical information. This can be particularly valuable for mitigating noise, enhancing model performance, and simplifying complexity. However, a key consideration is the potential loss of strategic image information if the number of principal components becomes excessively small.
In contrast, vertical optimization involves the use of K-means, a clustering algorithm aimed at partitioning data into coherent groups or clusters based on similarity. This approach is instrumental in uncovering underlying patterns and trends within the data, facilitating a deeper understanding of dataset characteristics. Nevertheless, it carries the risk of losing the images themselves, which are deemed strategic information.

Recognizing these challenges and trade-offs, we have explored the concept of hybridization as a viable solution. Our goal is to synergize the advantages of both approaches, leveraging PCA's proficiency in reducing dimensionality and K-means' expertise in cluster analysis. This hybrid solution strives to produce cleaner datasets with diminished noise levels, offering the potential for a well-organized cluster framework, ultimately augmenting modeling effectiveness.

Experimental Studies

In the forthcoming section, we will showcase the results of our experiments, in which we assess the performance of our approaches. We utilized the Ho-LBP descriptor and employed the SVM classification algorithm. These methods were implemented using The Extended Yale Face Database B. The experiments were executed on a system with a 2.5 GHz Intel Core i5 CPU, 8GB of RAM, and a 64-bit Windows operating system.

Dataset

The Extended Yale Face Database B (EYFB) is a renowned resource in the realm of computer vision and facial recognition. It features grayscale images of individuals captured under diverse lighting conditions, offering a challenging dataset for testing and advancing face recognition algorithms. With images of numerous subjects, each portrayed in varying poses and lighting scenarios, EYFB presents a real-world complexity that makes it a benchmark for evaluating the robustness and accuracy of facial recognition systems. Researchers frequently employ this dataset to assess and enhance the performance of algorithms for face detection, feature extraction, and identity verification. The Extended Yale Face Database B comprises a collection of 16,128 grayscale face images featuring 28 human subjects. Each subject is portrayed in 9 distinct poses and under 64 varying illumination conditions. These images are presented at a resolution of 192x168 pixels.

The lighting conditions in the database vary from frontal lighting to extreme side lighting, including cast shadows. This wide range of lighting variations makes the database challenging and suitable for evaluating the robustness of face recognition algorithms. The Extended Yale Face Database B is often used by researchers and developers in the field of computer vision and face recognition to evaluate and compare the performance of their algorithms. It provides a standardized benchmark for testing face recognition methods under varying lighting conditions. It's worth noting that while the Extended Yale Face Database B is a useful resource for research and development, it's important to ensure that any use of the database complies with the licensing terms and conditions associated with its distribution.

Experimental Results and Discussion

In practice, principal components are intricate combinations of features that adeptly represent data while preserving its unique characteristics, all while ensuring that information overlap is minimized. Original features often exhibit substantial redundancy, underscoring the effectiveness of Principal Component Analysis (PCA) in dimensionality reduction (Granato et al., 2018). When data is projected into a lower-dimensional subspace using principal components as axes, similar data points tend to naturally cluster together (as illustrated in Figure 2). This clustering phenomenon arises because the data representation explicitly aligns with axes that maximize variance (Granato et al., 2018). This approach facilitates the reduction of image sizes within the dataset, all while retaining representative samples. It proves particularly invaluable when dealing with large datasets, offering the means to create smaller, more manageable subsets for experimentation or the training of machine learning models. In a similar vein, K-means clustering, another widely used technique, groups similar data points together, contributing to efficient data organization and analysis in various domains (Garbade & Michael, 2018).

Scenario No. 1: Classification without Optimization

This experimentation (classification without optimization) will be used to compare the results in Table 1 with the other experiments in which the dataset is optimized. The goal is to assess the effectiveness of each optimization type.

Table 1. Classification result without dataset optimization

Precision	Time
61.56	248.44

Scenario No. 2: Classification after Applying PCA

This time, we applied PCA to the dataset before classification. As shown in Table 2 describing the results, the parameter 'C' represents the number of components to describe the training data. With C = 2, the accuracy dropped to 19%. However, with C = 10, accuracy increased to 56%, though it remained lower than the results in Table 2 for classification without optimization. When using C = 20, we achieved the same accuracy as the unoptimized classification, which proves that strategic information has been well preserved. Simultaneously, the execution time has undergone noticeable changes.

Table 2 Classification results after applying PCA

C	Precision (%)	Time (sec)
2	19.23	245.39
10	56.40	244.56
20	61.56	246.11

Scenario No. 3: Classification after Applying K-Means Algorithm

When applying K-means clustering for dataset reduction, it's crucial to make thoughtful choices regarding the features you use and the value of 'k,' which represents the number of clusters or groups in which the data will be partitioned. These choices can significantly impact the quality of the reduced dataset and its effectiveness for downstream tasks.

In this scenario, we utilize K-means for dataset optimization. As depicted in Table 3, we observed a substantial reduction in computation time, amounting to 30 seconds and 40 seconds, respectively, for each experiment, while maintaining the same number of images to be eliminated. Specifically, with K = 9, we achieved a more significant reduction in computation time. Furthermore, it's worth noting that the accuracy remained unchanged, and the number of rows (images) was unaltered, even though the number of clusters was modified. It's essential to highlight

that the accuracy consistently remained at 59.67%. In each experiment, 290 rows (images) were removed from the final dataset (see Table 3).

Table 3. Classification after applying K-means algorithm

S	K	Nb of eliminated Img	Precision (%)	Time(sec)	Clustering time
0.2	9	290	59.67	211.23	231.35
0.2	18	290	59.67	220.50	230.11

Scenario No. 4: Classification after Applying PCA then K-Means Algorithm

In this scenario, we considered re-running scenario 3 by incorporating PCA before implementing the K-means algorithm. We observed that the results were nearly identical, except for the clustering time, which was reduced by more than half. The significant advantage of this approach lies in the substantial reduction in the overall execution time of the process, as the clustering time accounts for a significant portion of the total duration, without affecting the other performance metrics.

Table 4. Classification after applying PCA then K-means algorithm

S	C	K	Eliminated Img	Precision (%)	Time(sec)	Clustering time
0.2	2	9	290	59.67	212.55	75.13
0.1	2	18	290	59.67	220.50	80.22

Getting similar or nearly identical results in various scenarios is a positive indicator. This demonstrates that dataset reduction or optimization was carried out while preserving crucial information.

Scenario No. 5: Classification after Applying PCA then K-Means Algorithm and then PCA

In this final scenario, we incorporated PCA both before calculating the distances between images to identify duplicates and prior to the image classification process. The results, as depicted in Table 5, demonstrate a minimum calculation time of 82 seconds with a 51% precision rate for parameters s = 0.1, c = 2, and k = 18. When using s = 0.05, c = 2, and k = 9, we achieved a precision of 56% with a calculation time of 121 seconds. It's important to note that outcomes vary depending on the parameter settings. In Figure 4, we visualize the optimization process for a dataset class. Each color corresponds to a cluster, with Y representing the centroids and X representing the redundant images. Figure 4 makes it evident that as images get closer in distance, the need for elimination becomes more pronounced.

Table 5. Classification after applying PCA then K-means algorithm and then PCA

S	C	K	Eliminated Img	Precision (%)	Time(sec)	Clustering time
0.1	2	4	3994	52.08	85.00	72.30
0.5	2	5	3660	56.07	100.42	94.17
0.5	2	9	2994	56.25	121.99	96.08
0.5	2	18	2101	57.85	154.11	120.15
0.5	2	18	4070	51.67	82.47	117.43

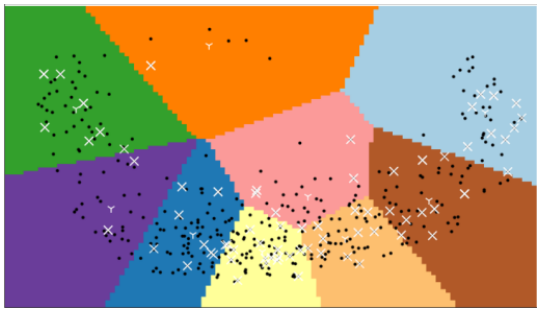

Figure 4. Kmeans clustering and images to be eliminated

In the context of our experiments, we observed the interplay between precision, computation time, and various optimization scenarios. Precision exhibited a slight, nearly negligible decrease, and computation time remained consistently variable, contingent on the specific test and its corresponding parameters. Among the scenarios tested, Scenario N°5 stood out as particularly intriguing due to its varying precision and computation time across different parameter values. In Scenario N°5, the distinction from Scenario N°4 lies in the reduction of image size within clusters through PCA. We then computed distances between these reduced images to eliminate duplicates. The introduction of PCA prior to clustering and distance calculation yielded a significant reduction in computation time. Furthermore, we noted that Scenarios N°4 and N°3 yielded nearly identical results since both involved the elimination of the same number of images. There was no discernible difference between the outcomes of Scenario N°2 and Scenario N°1.

Conclusion and Future Works

The combined use of PCA and K-means in dataset reduction presents a powerful approach for enhancing efficiency and maintaining data quality. PCA effectively reduces dimensionality by preserving essential information, and K-means clustering refines dataset organization. This synergy optimizes data while minimizing redundancy. However, the choice of parameters and feature selection plays a pivotal role in achieving the desired results. When thoughtfully applied, this method streamlines data processing, accelerates computations, and streamlines machine learning models, making it a valuable asset in the realm of data optimization.

Through these experiments, we have been able to observe the potential of eliminating redundant data. By using PCA thoughtfully, we achieved satisfactory results. It's worth noting that the success of this experiment could vary depending on the choice of dataset, algorithms, and implementations. Therefore, conducting a variety of experiments is essential to solidify our assessment of the proposed dataset optimization technique.

The challenge of class imbalance is significant. It occurs when one or more classes have considerably fewer data samples than others, leading to an unequal distribution. This can result in models performing well on the majority class but poorly on the minority class, causing inaccurate and biased predictions. Addressing class imbalance is crucial for model fairness and accuracy.

Notice that K-means clustering is just one of many techniques for identifying duplicate images. Depending on the size and characteristics of the dataset, we think using other methods like perceptual hashing (e.g., pHash) method may also be suitable for duplicate image detection. It's important to choose the right feature representation and similarity metric for comparing images, as this will affect the clustering results and the threshold for identifying duplicates. Also, the choice of 'K' should be guided by the diversity of the dataset and the expected number of distinct duplicate groups.

Scientific Ethics Declaration

The authors declare that the scientific ethical and legal responsibility of this article published in EPSTEM journal belongs to the authors.

Acknowledgements or Notes

* This article was presented as an oral presentation at the International Conference on Technology, Engineering and Science (www.icontes.net) held in Antalya/Turkey on November 16-19, 2023.

References

Comon, P. (1994) Independent component analysis: A new concept?. *Signal Processing*, *36*(3), 287-314.
Cox, T. & Cox M., (1994). *Multidimensional scaling.* London: Chapman & Hall.

Dutta, S., & Ghosh, A. K. (2016) On some transformations of high dimension, low sample size data for nearest neighbor classification. *Mach Learn, 102*, 57–83.

Gafour, Y., Berrabah, D., & Gafour, A. (2020). A novel approach to improve face recognition process using automatic learning. *International Journal of Computer Vision and Image Processing (IJCVIP), 10*(1), 42-66.

Garbade, Dr., & Michael, J. (2018) Understanding K-means clustering in machine learning. Medium, towards data science. Retrieved from https://towardsdatascience.com/understanding-kmeans-clustering-in-machine-learning

Granato, D., Santos, J. S., Escher, G. B., Ferreira, B. L., & Maggio, R. M. (2018). Use of principal component analysis (PCA) and hierarchical cluster analysis (HCA) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective. *Trends in Food Science & Technology, 72*, 83–90.

Hajibabaee, P., Pourkamali-Anaraki, F., & Hariri-Ardebili, M. A. (2021). An empirical valuation of the t-SNE algorithm for data visualization in structural engineering. *20th IEEE International Conference on Machine Learning and Applications (ICMLA*), 1674-1680. Pasadena, CA, USA.

Hinton, G. E., & Salakhutdinov, R. R. (2006) Reducing the dimensionality of data with neural networks. *Science, 313* (5786), 504–507.

Jolliffe, I.T., & Cadima, J. (2016) Principal component analysis: a review and recent developments. *Philos Trans A Math Phys Eng Sci., 374*(2065).

McInnes, L., Healy, J., Saul, N., & Großberger, L. (2018). UMAP: Uniform manifold approximation and projection. *Journal of Open Source Software, 3*(29), 861.

Milošević, D., Medeiros, A. S., Piperac, M. S., Cvijanović, D., Soininen, J., Milosavljević, A., & Predić, B. (2022). The application of uniform manifold approximation and projection (UMAP) for unconstrained ordination and classification of biological indicators in aquatic ecology. *Science of The Total Environment, 815*.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by local linear embedding. *Science, 290* (5500), 2323-2326.

Shi, X., Yang, H. U. I., Xu, Z., Zhang, X., & Farahani, M. R. (2019). An independent component analysis classification for complex power quality disturbances with sparse auto encoder features. *IEEE Access, 7*, 20961-20966.

Tenenbaum, J. B., de Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290* (5500), 2319-2323.

Van der Maaten, L.J.P., & Hinton, G.E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research, 9*(86), 2579-2605.

Wang, K., Su, G., Liu, L., & Wang, S. (2020). Wavelet packet analysis for speaker-independent emotion recognition, *Neurocomputing, 398*(4), 257-264.

Yousaf, M., Rehman, T.U. & Jing, L. (2020) An extended isomap approach for nonlinear dimension reduction. *Sn Computer Science, 1*, 160.

Zhang, Q., & Izquierdo, E. (2006). A multi-feature optimization approach to object-based image classification. In: H. Sundaram, M. Naphade, J. R., Smith, Y. Rui (Eds)., *Image and video retrieval. CIVR 2006. Lecture notes in computer science* (*p.*4071). Springer, Berlin, Heidelberg.

Author Information

Djamel Berrabah
Djillali Liabes University
Sidi Bel Abbes, Algeria
Contact e-mail: *berrabdjamel@gmail.com*

Yacine Gafour
Ibn Khaldoun University
Tiaret, Algeria

To cite this article:

Berrabah, D., & Gafour, Y. (2023). Improve image classification using data optimization. *The Eurasia Proceedings of Science, Technology, Engineering & Mathematics (EPSTEM), 26,* 262-271.