ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

IMPROVING IMAGE RETRIEVAL PERFORMANCE WITH SCS AND MCS CLUSTERING TECHNIQUES

Ulul Fikri[1], Rahmat Prakoso[2], Yufis Azhar[3]

[1,2,3]Universitas Muhammadiyah Malang

email: ulülfikri4@gmail.com[1], prakosorahmat24@gmail.com[2], yufis@umm.ac.id[3]

Abstract

This paper presents two methods, Single Cluster Search (SCS) and Multiple Cluster Search (MCS), aimed at enhancing image retrieval performance on the Corel1k, Corel5k, and Corel10k datasets, which has a wide variation of images. The Multi Texton Co-Occurrence Descriptor (MTCD) method is used for feature extraction, and the K-Medoids and DBSCAN methods are used for dataset clustering. The clusters are then ranked based on the distance of their medoids to the query image. The most relevant images are retrieved from the highest-ranking clusters. SCS selects the cluster with the highest ranking as the search area and expands the search area to the next ranking cluster if the number of images is less than 6, which is the desired number of retrieval results. MCS merges several clusters with the highest ranking and combines clusters as the search area. Both methods are evaluated using several metrics, such as AP, MRR, and retrieval time. The results are also compared with the original method, which does not use clustering (the query image and the dataset are only extracted with MTCD, and their distance is calculated). The findings indicate that both methods improve the retrieval time. In Corel1k, the SCS method reduces the time complexity by 0.001s, while the MCS method, although not surpassing the original method, still shows potential. In Corel5k, both methods reduce the time complexity by 0.052s in the SCS method and 0.015s in the MCS method. In Corel10k, both methods reduce the time complexity by 0.122s in the SCS method and 0.058s in the MCS method, compared to the original method. These results have practical implications for improving image retrieval efficiency. The paper discusses the reasons behind these results and suggests possible directions for future research.

Keywords : image retrieval, Corel, MTCD, clustering, K-Medoids, DBSCAN.

Received: 16-02-2024 | **Revised:** 03-04-2024 | **Accepted:** 12-04-2024
DOI: https://doi.org/10.23887/janapati.v13i2.75643

INTRODUCTION

Image retrieval is the act of finding and retrieving images from a collection based on their content or features. It has many applications in various fields, such as computer vision, multimedia, education, health, and art [1–5]. However, image retrieval is also challenging because different images may have other characteristics or variations, such as color, texture, shape, size, orientation, illumination, noise, and occlusion.

The dataset used for the retrieval significantly impacts its performance [6]. A good dataset should have a sufficient number and variety of images that relate to the user's queries and interests. However, some datasets may have a wide variation of images, which means that the dataset's images have different or various features or categories. This may make it difficult for the image retrieval system to find and rank the most relevant images for the user's queries.

This paper focuses on the Corel1k, Corel5k, and Corel10k datasets, some of the most widely used datasets for image retrieval research. The Corel1k dataset has 1000 images categorized into 10 groups, the Corel5k dataset has 5000 images categorized into 50 groups, and the Corel10k dataset has 10,000 images categorized into 100 groups, each category containing 100 images. The wide variation in categories and features from thex Corel dataset poses a challenge for image retrieval, it requires the image retrieval system to handle different types of features and categories and deal with the ambiguity or similarity of images across various categories.

This research aims to improve the image retrieval performance on the Corel1k, Corel5k, and Corel10k datasets using clustering techniques. Clustering is the method of grouping or partitioning a collection of data points toward subsets or clusters based on some similarity or distance measure [7]. Clustering can help improve image retrieval performance by reducing the search space and increasing the relevance of the retrieval results by focusing on the clusters that are most similar or closest to the user's query [8]. We also used Multi Texton Co Occurrence Descriptor (MTCD) as a feature extraction. MTCD is used to increase image retrieval performance by extracting color, texture, and shape features simultaneously using textons and then calculating the global representations of the image [9].

This paper proposes two methods: Single Cluster Search (SCS) and Multiple Cluster Search (MCS). Both methods use MTCD to identify the features from the images, proceeding by clustering the images according to the features that have been extracted. Both methods then rank the clusters based on the distance of their medoids to the query image and search for the most relevant images from the highest-ranking clusters. SCS takes the cluster with the highest ranking as the search area and expands the search area to the next ranking cluster if the number of images in the search area is less than 6, which is the desired number of retrieval results. MCS merges several clusters with the highest ranking and combines clusters as the search area. We evaluate both methods using metrics such as Average Precision (AP), Mean Reciprocal Rank (MRR), and retrieval time. We discuss the reasons behind the obtained results and suggest possible future research improvements.

Many approaches have been given to improve image retrieval performance, such as [8]. This research uses K-Means and Particle Swarm Optimization (PSO) on the Corel1k dataset, intending to reduce the computational complexity in image searching and retrieval by limiting the search space. K-Means groups images into relevant subsets, so in the search phase, the features of the query image are compared to the cluster centers to identify the most suitable cluster set. The K-Means clustering algorithm uses PSO to specify the optimum number of clusters and centroids. This research yielded an average precision value of 0.805% with low execution time. Another study by [10] applied K-Means and Moth Flame Optimization (MFO). The research was conducted on the Corel1k and COIL datasets, implementing MFO to specify the optimum number of clusters and cluster centroids in K-Means. This research provides an average precision value of 0.853% and an average recall value of 0.3% on the COIL dataset, an average precision value of 0.813%, and an average recall value of 0.169% on the Corel1k dataset with low execution time for both datasets.

Another research related to improving image retrieval performance [9] using the suggested approach developed from the MTH method in research [11], namely the Multi Texton Co-Occurrence Descriptor (MTCD). This method is the result of adding two new textons to MTH, resulting in extraction in the form of shape, color, and texture represented globally using GLCM. 15,000 Corel images and 300 Batik images are used as the dataset, with Canberra as the distance matrix. The conducted research provides results in an increase in precision by 2.86% and recall by 3.12% on the Batik dataset. There is a performance improvement for the Corel dataset with a precision increase of 3.41% and a recall increase of 0.41% on the Corel 5000 dataset. In contrast, on the Corel 10000 dataset, there is a performance improvement with a precision increase of 3.06% and a recall increase of 0.37% compared to MTH.

The results from previous research indicate that implementing clustering methods in image retrieval provides efficiency in execution time, while MTCD can improve the precision and recall values. Issues in earlier studies are related to computational speed and time evaluation, where the use of a significant number of datasets impacts execution time, and the time evaluation is restricted to small datasets, thus not providing a comprehensive overview of the proposed alternative performance.

This research aims to apply clustering methods in image retrieval through the proposed methods (SCS) and (MCS) with MTCD as feature extraction. This is done to reduce the required execution time while simultaneously improving precision and recall values. Thus, the evaluation can provide a more comprehensive understanding of the proposed alternatives. The difference between this study and previous research was implementing clustering methods, including DBSCAN and K-Medoid, before the retrieval process was executed. Additionally, the implementation of the proposed method is in the form of (SCS) and (MCS).

The rest of this paper is organized as follows. Section 2 describes the proposed methods in detail. Section 3 presents the experimental setup and results. This section also discusses the findings and limitations of the paper. Section 4 concludes the paper and gives some directions for future work.

METHOD

The proposed method uses clustering techniques to enhance image retrieval performance on various dataset sizes: Corel1k, 5k, and 10k. This method consists of several stages: dataset preparation, feature extraction, clustering, clustering ranking, image retrieval, and evaluation of retrieval results. Figure 1 provides a general overview of the research methodologies.

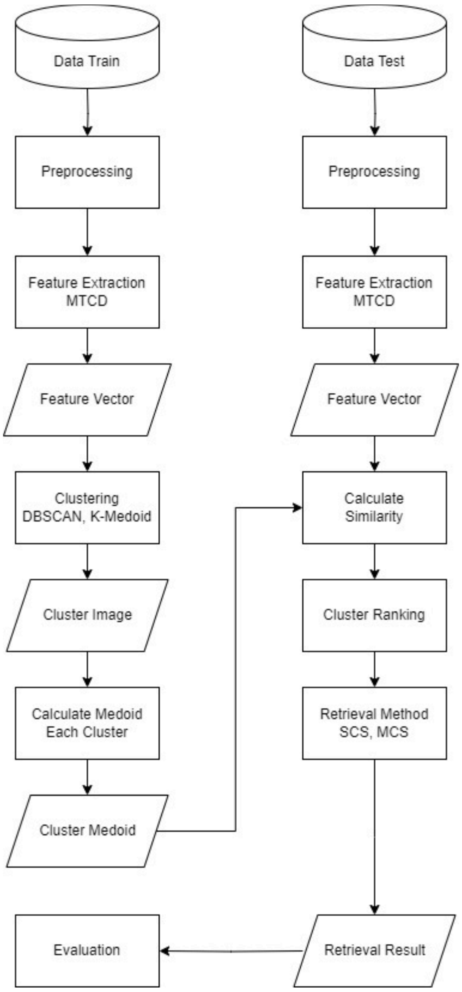

Figure 1. Research Methodology

The initial stage involves preparing the dataset into two subsets: the training and the test data. The Multi Texton Co-occurrence Descriptor (MTCD) is used to extract features from images in the training and test datasets. In the training data stage, the extracted features of each image are clustered, and the medoid points of each cluster are calculated. In the test data stage, the feature extraction results calculate similarities and distances between the test data features and the medoid of each cluster. This is done to determine which cluster has the closest similarity to each test data image, sorted in ascending order based on the calculated distances. Then, image retrieval is performed using the method proposed by the researcher, the SCS or MCS method. The final step of the research involves evaluating the retrieval results.

Dataset

This study used the Corel1k, Corel5k, and Corel10k datasets. The Corel10k dataset was obtained from the Kaggle website with the title 'Corel-10K,' while the Corel1k and 5K data were taken from the Corel10k dataset. The details of each dataset can be found in Table 1.

Table 1. Total Data Per Subset and Category Per Dataset

Dataset	Subsets	Total Data	Total Category
Corel1k	Train	990	10
	Test	10	
Corel5k	Train	4950	50
	Test	50	
Corel10k	Train	9900	100
	Test	100	

The test data was selected by taking one representative image from each category within the dataset. Therefore, the entire quantity of query images corresponds to the number of categories in each dataset. Each category consists of 100 images. Corel1k consists of 1,000 data in 10 categories, Corel5k consists of 5,000 data in 50 categories, and Corel10k consists of 10,000 data in 100 categories.

Preprocessing

The preprocessing in this study involves data collection and resizing. This stage uses the OpenCV library. Initially, the pixels in the image are converted into an array format. Then, the data undergoes the process of resizing. During this stage, the array dimensions are changed to 64 by 64.

Feature Extraction

This step aims to uncover various information from images. In the context of image retrieval based on feature similarity, this study uses a feature extraction method called MTCD, developed in the research [12]. This method combines feature extraction techniques (Multi Texton Histogram and Gray Level Co-Occurrence Matrix) [9]. The MTCD feature extraction process consists of three stages. The first stage involves detecting local features using MTH, while the second stage focuses on detecting global features using GLCM. In the last stage, all detected features are merged and represented as a feature vector [12].

Multi Texton Histogram

In the first step, edge detection is performed on the image to obtain more detailed information. In this study, we used the Sobel Edge Detection method [3], which was selected for its advantages in handling large datasets and achieving higher speed than other methods [13].

The Sobel method produces vectors and magnitudes, then quantized into 18 bins. This approach avoids gradient calculations around the interpolated point between pixels using a 3x3 pixel matrix. This method generates a histogram representing edge information from the image with 18 features.

Several CBIR schemes have been designed using color information derived from histograms [14]. Color histograms are commonly used as features to extract images. In this study, colors are divided into RGB color components. Subsequently, each color component R, G, and B is quantized into four bins, resulting in a total of 64 bins [15]. This method produces a histogram that represents color information from the image with 64 features.

The next step involves texton detection within the quantized results of color and edge orientation [15]. Unlike MTH, which uses four textons for detection, this study uses a developed method named MTCD, using six textons for detection. This refinement aims to address the limitations of MTH, which may result in the loss of crucial information in images that plays a significant role in explaining images more comprehensively. Two new textons were added: the bottom horizontal and the right vertical [16]. The texton identification process used in this study begins by creating a zero matrix of the original image's size. Every kind of texton is then convolved from the image's top-left corner to the bottom-right corner. If any pixels are present with values matching the texton pattern, those values are transferred to the prepared zero matrix. The last outcome of this step is six matrices containing the identified texton values, which will be utilized in the following stages [12].

Gray Level Co-Occurrence Matrix

Statistical methods can describe texture in an image by quantitatively measuring intensity [17]. From the thirteen Haralick features, the proposed approach extracts four main features: Angular Second Moment, Contrast, Correlation, and Entropy.

In this study, GLCM is implemented through several steps [9]. The first stage involves converting the RGB image into a grayscale image. The next step includes making a co-occurrence matrix and determining the spatial relationships between the reference pixel and its neighbors. The subsequent step involves generating a symmetric matrix by adding the co-occurrence matrix with its transpose. Afterward, matrix normalization is performed by computing the probabilities for each matrix element. The last stage involves computing this study's four main GLCM features. This method produces a histogram representing texture information from the image with 16 features.

MTCD Feature Representation

The final step of this method is to combine all extraction results, consisting of a histogram representing color and edge information from MTH extraction with six textons and texture information from GLCM extraction. Combining all detected features is described as a global feature in a feature vector. The total number of features resulting from the MTCD method is 98, consisting of 64 color features, 18 edge features, and 16 GLCM features [12].

Clustering

In this stage, the clustering method is implemented based on the feature extraction results of the training data. The clustering methods used in this research are DBSCAN and K-medoids, each using the Canberra distance metric.

DBSCAN

DBSCAN is a clustering based on a density algorithm that separates data points into core points and outlier points based on the density level of points in their neighborhood. This algorithm effectively handles datasets of various sizes or shapes and is particularly adept at identifying outliers [18]. This makes it suitable for use in Corel datasets which have a wide variety of images. The algorithm uses density differences to identify regions of different densities and marks clustering results [19]. In this study, the procedure for determining the number of clusters is based on the hyperparameters of DBSCAN, namely epsilon (ε) and minimum samples, which are set to approximate the total number of categories in each dataset as closely as possible. Additionally, clusters identified as outliers are removed.

K-Medoids

K-Medoids is a technique that produces representative clusters by assigning other objects to the representative object items that closely match the chosen objects, minimizing the amount of data dissimilarity in the cluster because this method is not influenced by outliers or other extreme variables [20]. K-medoids operate by finding the center point of the existing data without computing averages, as done by the K-means method [21]. K-Medoids produces more stable clusters because the medoids, as the center point of each cluster, are less sensitive to outliers than the K-Means method, which uses

Mean [22]. Unlike the DBSCAN method, the amount of clusters in K-Medoids can be determined by setting the value of K. In this study, the value of K follows the number of categories in each dataset.

The next step involves calculating the medoid for each cluster in the training data using the median. Calculating the cluster medoid using the median is considered more optimal than the mean. The median method was chosen because it is less sensitive to outliers [22]. In the final step, the distance between the query image and the medoid of each cluster is calculated, and the clusters are ranked based on the ascending order of the calculated distances.

We compare the performance of these two algorithms in dealing with the Corel dataset, in which DBSCAN can identify outliers so that they are not included, and K-Medoids, which includes outliers but still has stable cluster.

Distance Measure

In this study, distance measurement is used to calculate the similarity between the query image and the medoid of every cluster and the images within each cluster. The distance metric used in this research is Canberra [9].

SCS & MCS

After obtaining the cluster order from the nearest to the farthest from the query, the following steps involve implementing two approaches to retrieve six images.

Figure 2 illustrates the process of Single Cluster Search (SCS). This method selects the top-ranked cluster as the search area. The next step is to measure the similarity distance between the query and the images within the search area based on the ascending order of the calculated distances. If the number of images in the search area exceeds 6, SCS will expand the search area to the next-ranked cluster. For example, if Cluster A is the nearest cluster to the query, followed by Clusters B, C, D, E, and F, SCS will first search for images in Cluster A. Then, each image in Cluster A, such as A1, A2, and A3, will be retrieved. In this case, if the image taken does not reach the required number of retrieved six images, SCS will continue the search in Cluster B. Images in Cluster B, such as B1, B2, B3, and B4, will have similarity to the query image calculated, the three images closest to the query will be retrieved. Then, SCS will stop the search because the required number of retrieved images is reached.

Figure 3 illustrates the process of Multi-Cluster Search (MCS). This method involves determining the number of clusters to be taken. In this research, we determined five clusters as the search area. After that, MCS will combine all images in each cluster with the highest ranking into one group. The results of the merged cluster are used as the search area. Next, MCS calculates the similarity distance between the query image and the image in the search area based on the ascending order of the calculated distance. Suppose the number of images taken still does not meet the criteria for taking six images. In that case, MCS will add one next ranking cluster to be merged and recalculate the similarity distance between the query image and the image in the additional merged cluster. Adding the next ranking cluster will continue until the required retrieved images are reached. For example, cluster A is closest to the query, followed by Clusters B, C, D, E, and F. In the first stage, MCS will combine clusters A, B, C, D, and E. The result of merging the five clusters will be an area search. If after calculating the similarity distance between each image from the clusters that have been incorporated and the image query, images A1, B1, C1, C2, A2, and A3 are the six images with the highest ranking, then MCS will retrieve those six images and then stop the search. Meanwhile, if the number of images taken still does not reach the required number for taking six images, MCS will add the next ranking cluster to the search area by combining the six closest clusters. For example, MCS will combine clusters A, B, C, D, E, and F, then repeat the retrieval process from the results of combining the six clusters into one group.

The results of retrieving six images using the method proposed by the researcher will undergo a comprehensive evaluation process. The evaluation process of retrieving six images using the researcher's proposed methods involves a thorough examination to understand each method's performance better. Various metrics will be used to assess the effectiveness and efficiency of the proposed methods approaches. Additionally, the proposed methods will be compared with the original method, which doesn't use clustering, to highlight their strengths and weaknesses in image retrieval. By analyzing these comparisons, this research aims to provide valuable information and also make a significant contribution to the development of image retrieval technology.

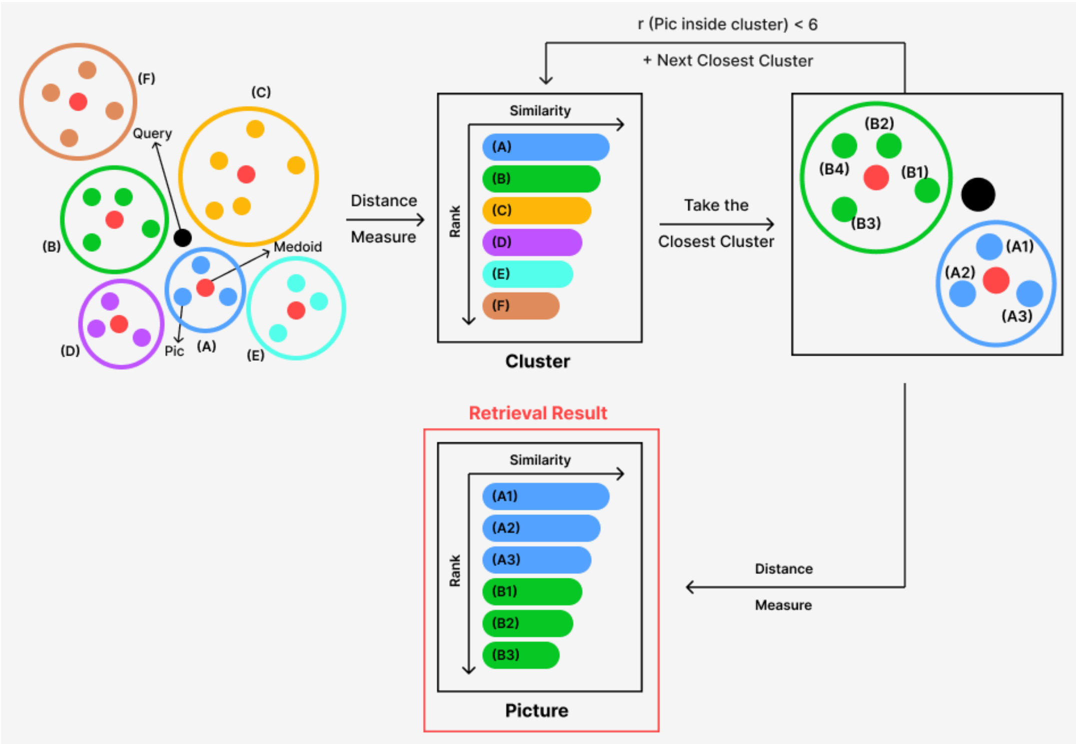

Figure 2. Illustration of SCS

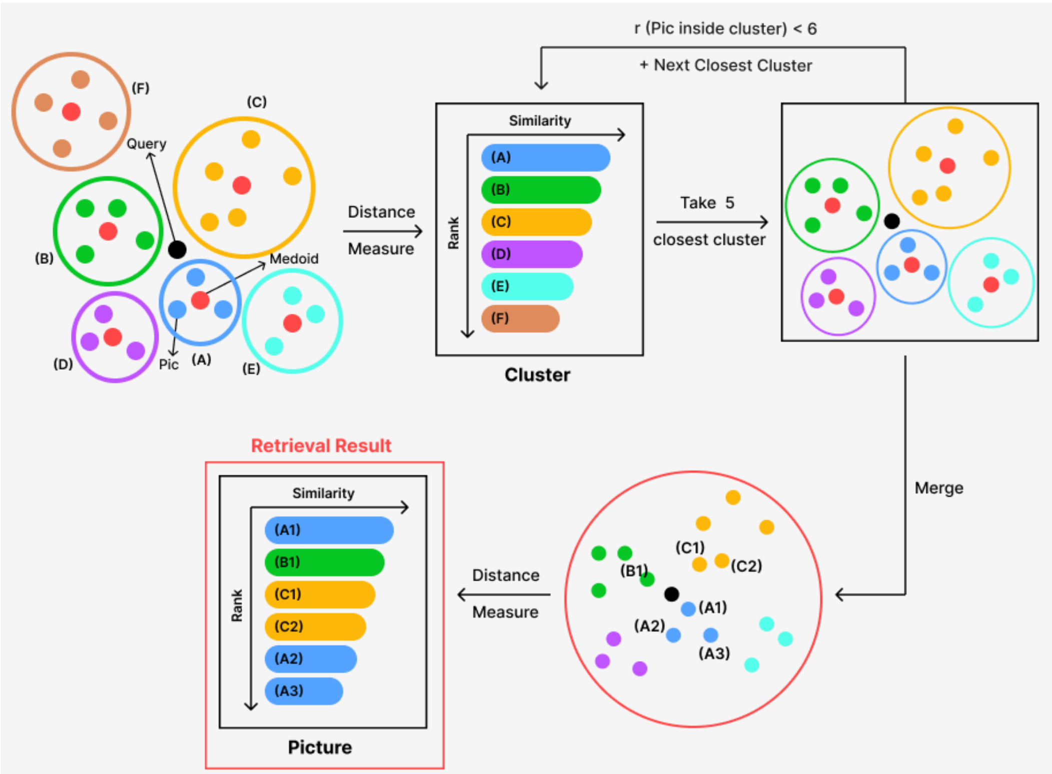

Figure 3. Illustration of MCS

Evaluation

Based on the retrieval results obtained for each query, we used average precision (AP) and mean reciprocal rank (MRR) to analyze the Image Retrieval system performance we created. Equation 1 calculates Average Precision, and Equation 2 calculates Mean Reciprocal Rank.

$$AP = \frac{1}{|Q|}\sum_{i=1}^{|Q|} P_i \tag{1}$$

Where:

P_i : The precision value for each query.
Q : Total number of queries.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|} \frac{1}{r_i} \tag{2}$$

Where:

r_i : Position of first relevant image for each query.
Q : Total number of queries.

Additionally, the Average Retrieval Time required for each query is considered a performance evaluation of the proposed system. Time calculation starts from the beginning of the process for each method proposed by the researcher until the method reaches the number of images to retrieve for each query.

RESULT AND DISCUSSION

The research stages obtained various results with the following details:

Cluster Performance Comparison

The method proposed by researchers aims to improve image retrieval performance. To evaluate this method's system performance, it will be implemented with various dataset sizes and several different clustering methods.

Table 2 shows that the K-medoids clustering method performs better than DBSCAN. Based on the overall experimental results, the SCS method outperforms the MCS method, although the MCS method can show superiority over the AP parameter on smaller datasets. However, when faced with datasets with a wide variation of images and ambiguity or similarity among image categories, the performance of the MCS method decreases. In terms of AP, which is a parameter that indicates the quality of the retrieval system, MRR evaluation is another parameter to measure the extent to which the method proposed by researchers provides relevant image results at the top of the ranking. This is shown in the MRR results. The closer the number is to 1, the better it is. Based on the experiments, MRR performance tends to decrease, while the difference in performance between the two methods proposed by researchers is insignificant. However, both methods are consistent in terms of Average retrieval time.

The experimental result shows that SCS is superior to MCS. This is due to the higher execution time of the MCS method, which results from merging five clusters into a new cluster. This process involves searching the dataset to find data with cluster labels that match the five merged clusters, introducing additional complexity and time for MCS.

Table 2. Each Dataset's Performance was Compared to Compare The K-Medoids and DBSCAN with The Image Retrieval Method Proposed by Researchers

Dataset	Clustering Method	AVG Precision		MRR		AVG TIME (s)	
		SCS	MCS	SCS	MCS	SCS	MCS
Corel1k	K-Medoids	0.78	0.93	1	1	0.014	0.073
	DBSCAN	0.45	0.4	0.5	0.46	0.005	0.020
Corel5k	K-Medoids	0.49	0.43	0.67	0.64	0.021	0.058
	DBSCAN	0.34	0.39	0.41	0.49	0.027	0.115
Corel10k	K-Medoids	0.26	0.24	0.42	0.43	0.024	0.088
	DBSCAN	0.33	0.37	0.48	0.56	0.665	1.018

We also analyzed the clustering process's results on the data train, examining the number of data (n) in each formed cluster. Table 3 compares the clustering methods used, presenting the number of data (n) in each cluster from the Corel1k dataset. These results show differences in the clusters formed by each clustering method.

ISSN 2089-8673 (Print) | ISSN 2548-4265 (Online)
Volume 13, Issue 2, July 2024

Table 3. Corel1k Number of Data From Each Clustering Results

K-Medoids		DBSCAN	
Cluster	n	Cluster	n
0	140	0	71
1	129	1	3
2	83	2	15
3	113	3	17
4	82	4	43
5	109	5	5
6	109	6	4
7	106	7	3
8	64	8	23
9	55		

The DBSCAN method cannot create 10 clusters according to the number of categories in Corel1k. This happens because the number of clusters formed by the DBSCAN method depends on the determination of hyperparameters. After conducting various parameter experiments, no parameters could form 10 clusters. Therefore, the parameters were set to be close to the number of clusters. In this case, DBSCAN formed 9 clusters. Unlike DBSCAN, determining the number of clusters from the K-Medoids method can be determined by the K parameter (number of clusters), so this parameter is set by equating the K value with the number of categories in the dataset. The previous analysis of the retrieval performance of K-Medoids provides superior and consistent results compared to DBSCAN. This is due to the clustering process in DBSCAN removing data identified as outliers, resulting in a lack of data for retrieval. Additionally, the number of data (n) in each cluster in DBSCAN is not evenly distributed. Some clusters have significantly large data, while others have very few. This pattern also occurs in the Corel5k and 10K datasets.

Retrieval Performance Comparison Between K-Medoids and Original Method.

Based on the results of the previous analysis, K-Medoids is a clustering method that performs better than DBSCAN. Therefore, we compare the performance of K-Medoids with the original method. The results of the original method were implemented without using the clustering and retrieval methods proposed by the researchers. However, this method still uses the same feature extraction.

Table 4 indicates a decreased performance of each evaluation metric in the original method (WC). This happens because of a common challenge in the original method, where performance tends to decrease when faced with wide variations of images and continuously growing data. The comparison between the K-Medoids and the original method indicates that the K-Medoids using the proposed method exhibit an improvement in execution time performance. Especially in the SCS method, this method outperforms other methods with the average time required for each query in the image retrieval process. Compared to different methods, this method tends not to have a significant decrease in time when the dataset continues to increase. This happens because the search area of images in the SCS method is not as wide as other methods. The MCS method merges five clusters as the first search area process, which will continue to increase if the number of images taken does not reach six images while the original method takes all data in the data train as its search area. However, the retrieval performance results have not yet surpassed the original method.

Table 4. Performance Comparison of K-Medoids using SCS and MCS with The Original Method (without Clustering)

Dataset	AVG Precision			MRR			AVG TIME (s)		
	SCS	MCS	WC	SCS	MCS	WC	SCS	MCS	WC
Corel1k	0.78	0.93	0.98	1	1	1	0.014	0.073	0.015
Corel5k	0.49	0.43	0.67	0.67	0.64	0.8	0.021	0.058	0.073
Corel10k	0.26	0.24	0.42	0.42	0.43	0.58	0.024	0.088	0.146

The findings of this research indicate that the choice of clustering method and image retrieval method can influence image retrieval performance. Additionally, the amount of data in each cluster can also affect image retrieval performance.

CONCLUSION

The research aims to improve the performance of image retrieval on the Corel1k, Corel5k, and Corel10k datasets using clustering techniques. The implementation of clustering is intended to reduce the search space in image retrieval and enhance the relevance of the retrieval results. The clustering process works by grouping images in the dataset into clusters based on relevant features. So, the retrieval process can be reduced to only clusters highly relevant to the query. Thus, the obtained image retrieval results have high relevance to the query, and the required computational time is reduced.

The results of the conducted research indicate that the implementation of both proposed methods, namely SCS and MCS, failed to outperform the measurement metrics in both AP and MRR but succeeded in outperforming the AVG time execution compared to the original method (without clustering). This suggests that the clustering techniques used are ineffective in improving the accuracy or relevance of retrieval results but effective in reducing the computational complexity. Several factors contributed to the failure of MCS and SCS to outperform the original methods on AP and MRR measurement metrics, including the selection of clustering methods such as K-Medoids and DBSCAN, also fail to produce optimal or stable clusters, as K-Medoids rely on the initial medoid selection and parameter k (number of clusters). At the same time, in DBSCAN, removing data identified as outliers reduces the retrievable data. Additionally, the choice of image retrieval methods, SCS and MCS, are not suitable for the characteristics of the Corel1k, Corel5k, and Corel10k datasets, which have a wide variation of images and ambiguity or similarity among image categories.

The limitations of this study include the proposed method, which only uses MTCD as its feature extraction, based on other research that used SE-ResNet-50 feature extraction to extract image information with a more complex background [23]. A representative feature vector can be produced with more features, while the lack of image features represented by MTCD is less representative of Corel data. Additionally, MTCD imposes limitations on the choice of distance measure metrics. MTCD performs optimally with the Canberra distance measure, restricting the parameters used in both the clustering distance parameter implementation and the SCS or MCS implementations. The choice of clustering method and parameter that depends on the number of categories used also influences research results.

Therefore, this study also provides several suggestions or recommendations for further research, including the use of other feature descriptor methods or combining multiple feature descriptor methods to capture more features from images, the exploration of alternative clustering methods, or optimization of clustering parameters to generate better clusters, the consideration of other features besides medoid of each cluster and using another different distance measure when using SCS and MCS methods in image retrieval, and the consideration of utilizing another dataset with greater or lesser variation, ambiguity, or similarity.

REFERENCES

[1] S. R. Dubey and S. Mukherjee, “LDOP: local directional order pattern for robust face retrieval,” *Multimed. Tools Appl.*, vol. 79, no. 9–10, pp. 6363–6382, 2020, doi: 10.1007/s11042-019-08370-x.

[2] R. Ashraf, M. Ahmed, U. Ahmad, M. A. Habib, S. Jabbar, and K. Naseer, “MDCBIR-MF: multimedia data for content-based image retrieval by using multiple features,” *Multimed. Tools Appl.*, vol. 79, no. 13–14, pp. 8553–8579, 2020, doi: 10.1007/s11042-018-5961-1.

[3] A. E. Minarno, F. D. S. Sumadi, Y. Munarko, W. Y. Alviansyah and Y. Azhar, "Image Retrieval using Multi Texton Co-occurrence Descriptor and Discrete Wavelet Transform," 2020 8th International Conference on Information and Communication Technology (ICoICT), Yogyakarta, Indonesia, 2020, pp. 1-5, doi: 10.1109/ICoICT49345.2020.9166361.

[4] A. Zhong *et al.*, “Deep metric learning-based image retrieval system for chest radiograph and its clinical applications in COVID-19,” *Med. Image Anal.*, vol. 70, p. 101993, 2021, doi: 10.1016/j.media.2021.101993.

[5] S. Lee, D. Kim, and B. Han, “COSMO: Content-Style Modulation for Image Retrieval with Text Feedback,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 802–812, 2021, doi: 10.1109/CVPR46437.2021.00086.

[6] X. Li, J. Yang, and J. Ma, “Recent developments of content-based image retrieval (CBIR),” *Neurocomputing*, vol. 452, pp. 675–689, 2021, doi: 10.1016/j.neucom.2020.07.139.

[7] K. P. Sinaga and M. S. Yang, “Unsupervised K-means clustering algorithm,” *IEEE Access*, vol. 8, pp.

80716–80727, 2020, doi: 10.1109/ACCESS.2020.2988796.

[8] Z. S. Younus *et al.*, "Content-based image retrieval using PSO and k-means clustering algorithm," *Arab. J. Geosci.*, vol. 8, no. 8, pp. 6211–6224, 2015, doi: 10.1007/s12517-014-1584-7.

[9] A. E. Minarno and N. Suciati, "Image retrieval using multi texton co-occurrence descriptor," *J. Theor. Appl. Inf. Technol.*, vol. 67, no. 1, pp. 103–110, 2014.

[10] A. Joseph, E. S. Rex, S. Christopher, and J. Jose, "Content-based image retrieval using hybrid k-means moth flame optimization algorithm," *Arab. J. Geosci.*, vol. 14, no. 8, 2021, doi: 10.1007/s12517-021-06990-y.

[11] G. H. Liu, L. Zhang, Y. K. Hou, Z. Y. Li, and J. Y. Yang, "Image retrieval based on multi-texton histogram," *Pattern Recognit.*, vol. 43, no. 7, pp. 2380–2389, 2010, doi: 10.1016/j.patcog.2010.02.012.

[12] Y. Azhar, A. E. Minarno, Y. Munarko, and Z. Ibrahim, "Image Retrieval Based on Texton Frequency-Inverse Image Frequency," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 5, pp. 87–94, 2020, doi: 10.22219/kinetik.v5i2.1026.

[13] G. Ravivarma, K. Gavaskar, D. Malathi, K. G. Asha, B. Ashok, and S. Aarthi, "Implementation of Sobel operator based image edge detection on FPGA," *Mater. Today Proc.*, vol. 45, no. xxxx, pp. 2401–2407, 2021, doi: 10.1016/j.matpr.2020.10.825.

[14] N. Varish *et al.*, "Image Retrieval Scheme Using Quantized Bins of Color Image Components and Adaptive Tetrolet Transform," *IEEE Access*, vol. 8, pp. 117639–117665, 2020, doi: 10.1109/ACCESS.2020.3003911.

[15] A. E. Minarno, Y. Munarko, A. Kurniawardhani, and F. Bimantoro, "Classification of Texture Using Multi Texton Histogram and Probabilistic Neural Network," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 105, no. 1, pp. 0–7, 2016, doi: 10.1088/1757-899X/105/1/012022.

[16] A. E. Minarno, Y. Azhar, F. D. Setiawan Sumadi, and Y. Munarko, "A Robust Batik Image Classification using Multi Texton Co-Occurrence Descriptor and Support Vector Machine," *2020 3rd Int. Conf. Intell. Auton. Syst. ICoIAS 2020*, pp. 51–55, 2020, doi: 10.1109/ICoIAS49312.2020.9081833.

[17] J. H. Dewan and S. D. Thepade, "Image Retrieval using Weighted Fusion of GLCM and TSBTC Features," *2021 6th Int. Conf. Converg. Technol. I2CT 2021*, pp. 1–7, 2021, doi: 10.1109/I2CT51068.2021.9418131.

[18] D. Deng, "DBSCAN Clustering Algorithm Based on Density," *Proc. - 2020 7th Int. Forum Electr. Eng. Autom. IFEEA 2020*, pp. 949–953, 2020, doi: 10.1109/IFEEA51475.2020.00199.

[19] M. Li, X. Bi, L. Wang, and X. Han, "A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm," *Comput. Commun.*, vol. 167, no. November 2020, pp. 75–84, 2021, doi: 10.1016/j.comcom.2020.12.019.

[20] S. Y. Tan, M. F. Akbar, N. H. M. M. Shrifan, G. N. Jawad, M. Nadhir, and A. Wahab, "Clustering Algorithm-Based Microwave Nondestructive Testing," 2022.

[21] E. H. S. Atmaja, "Implementation of k-Medoids Clustering Algorithm to Cluster Crime Patterns in Yogyakarta," *Int. J. Appl. Sci. Smart Technol.*, vol. 1, no. 1, pp. 33–44, 2019, doi: 10.24071/ijasst.v1i1.1859.

[22] E. Herman, K. E. Zsido, and V. Fenyves, "Cluster Analysis with K-Mean versus K-Medoid in Financial Performance Evaluation," *Appl. Sci.*, vol. 12, no. 16, 2022, doi: 10.3390/app12167985.

[23] Y. Wu, K. Song, F. Zhao, J. Chen, and H. Ma, "Distribution Aligned Feature Clustering for Zero-Shot Sketch-Based Image Retrieval," 2023, [Online]. Available: http://arxiv.org/abs/2301.06685