Software Impacts 15 (2023) 100486

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

CBIR-ANR: A content-based image retrieval with accuracy noise reduction

Gabriel S. Vieira [a,*], Afonso U. Fonseca [b], Fabrizzio Soares [b]

[a] Federal Institute Goiano, Computer Vision Lab, Urutaí, 75790-000, GO, Brazil
[b] Federal University of Goias, Institute of Informatics, Goiânia, 74690-900, GO, Brazil

ARTICLE INFO

Keywords:
Image retrieval
Image descriptor
Microstructures
Features fusion
Local binary pattern
Low-level features combination

ABSTRACT

Due to the expansion of multimedia data leveraged by social networks and smartphone devices, large sets of information are available and increasing daily. In this context, information retrieval is crucial to open new opportunities to individuals, governments, and businesses. Therefore, we present the CBIR-ANR software in which the content-based image retrieval (CBIR) is followed by an accuracy noise reduction (ANR) strategy that adjusts query responses and increases assertiveness in image retrieval. Also, the software combines three low-level features to form a 187-dimensional feature vector, which is size efficient for large-scale data sets and competitive with related work.

Code metadata

Current code version	*v1.0*
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2022-294
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/3022749/tree/v1
Legal Code License	GNU GPL-2.0 License
Code versioning system used	git
Software code languages, tools, and services used	MATLAB
Compilation requirements, operating environments & dependencies	Linux, Microsoft Windows, MATLAB R2014a
If available Link to developer documentation/manual	https://github.com/gabrieldgf4/cbir-anr
Support email for questions	gabriel.vieira@ifgoiano.edu.br

1. Introduction

Among computer systems, web search engines marked the transition between the previous century and the current one. The fast and wide distribution of content, corroborated by data's immediate and accessible presentation, characterized a moment in history generally referred to as the information age. In the current context, large sets of data are available and increasing exponentially every day in such a way that everyday life, business, and economy are directly and indirectly affected, positively and negatively influenced, by the amount of information that can be obtained from the mass of data of existing systems.

Advances in information and communication technologies, especially in developing computational equipment such as efficient hardware and large data storage devices, have driven the construction of solutions for information retrieval. As a result, text-based information retrieval systems gained notoriety, and web search engines became widespread and indispensable. In this sense, networked information systems began to offer advantages to governments, businesses, and individuals in terms of opportunity, knowledge, and decision-making with the consolidation of different sources of information. However, the expansion of multimedia data (images and video) leveraged by social networks and smartphone devices began to demand new information retrieval strategies in addition to the textual search approach. Two other propositions have emerged in this context: tag-based information retrieval (TBIR) and content-based image retrieval (CBIR).

In tag-based information retrieval, the elements that describe the multimedia content are previously annotated, where the information retrieval process consists of searching relevant text documents from a

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author.
E-mail addresses: gabriel.vieira@ifgoiano.edu.br (G.S. Vieira), afonso@inf.ufg.br (A.U. Fonseca), fabrizzio@ufg.br (F. Soares).

https://doi.org/10.1016/j.simpa.2023.100486
Received 5 December 2022; Received in revised form 20 January 2023; Accepted 27 February 2023

set of tags [1]. On the other hand, in content-based image retrieval, the descriptors are automatically obtained by image extractors that describe the multimedia content in the form of feature vectors [2]. In both approaches, the retrieved information is ranked, and the required content is presented according to the evaluated similarity between the query image and the entries in the data set. However, while the first approach evaluates the similarity between portions of text, the second one compares the virtual representation of images through the signatures representing multimedia attributes. In this regard, CBIR systems process images directly and evaluate image features as critical components to unambiguously identify an image and generically group sets of images by the similarity between their virtual representations. Unlike TBIR systems, CBIR systems do not need manual annotation of descriptors, which can be costly to obtain when expert opinion is required to label the data or unreliable when unskilled or uncommitted people do the annotation.

In this sense, it is easier to describe events or objects by capturing images rather than writing words for the moments experienced by people. Following that, image descriptors play a vital role in this task as they capture scene structuring elements to dispense textual descriptions and numerically represent images as one-dimensional vectors. To perform the virtual image representation, image descriptors capture low-level features such as color, shape, and image texture by calculating histograms and pixel association or by computing binary patterns, region-based clustering, and edge-based segmentation. As images carry content information, feature extractors identify points of interest, compute target areas, and encode relevant information into previously established-sized vectors. However, the dimensionality of the feature vectors varies according to the algorithms used. Consequently, design issues such as integrating different features or using computational learning models influence the processing and response time.

It is improbable that a single feature descriptor encodes all major image signatures. As descriptors operate on some low-level attribute, they perform specific tasks that involve cohesive processing of color, shape, or texture. Therefore, not all potential image features are encoded in a single descriptor, making it difficult to retrieve the information correctly in image categories that share identical elements but have different semantic contexts. The commonly applied strategy involves merging different image descriptors to address this limitation. As a result, the different descriptors cooperate, and superior assertiveness is achieved over individual descriptors. Another strategy is to apply learning models to recognize unique features of image categories through training. Deep learning architectures use multiple layers to extract relevant features for image classification and retrieval in this context. Then, low-level features and high-level concepts can be encapsulated to describe images. Although most deep learning approaches show superior results compared to shallow learning models and information retrieval methods that do not use training steps, some conditions limit deep learning approaches. Among these limitations is the dependence on annotated data required by supervised learning models, the processing time spent during training epochs, the hardware configurations required to optimize learning processes, and the limited ability of models to handle unexpected scenarios [2–4].

Although computational learning models and non-learning methods for information retrieval have different particularities, it can be observed that they have a regular behavior. CBIR methods return a predefined number of similar images for a given image query. For example, the number of retrieved images can be 5, 10, 12, 20, or any other arbitrary number. When the results obtained by the CBIR methods are presented, it is noted that the assertiveness in image retrieval decreases as the number of expected responses increases. Thus, there is an inverse relationship between the number of images retrieved and the accuracy of the CBIR models. Besides that, regardless of the query image used, the resulting accuracy is linked to the number of expected responses in which smaller amounts of retrieved images may increase the assertiveness of the methods. In this sense, there is a regular pattern, i.e., a repetitive design between different CBIR methods and between different query images related to the number of expected responses. Ali et al. [5] showed that their method reduced assertiveness as the number of retrieved images increased (Fig. 1(a)). Yu et al. [6] observed that adding more retrieved images decreased the accuracy of their method (Fig. 1(b)). Verma and Raman [7] showed that the average precision of their method was decremented by one percentage point with the addition of another image to be retrieved (Fig. 1(c)). Likewise, Pradhan et al. [8] showed decreasing assertiveness curves concerning the number of retrieved images (Fig. 1(d)).

To the best of our knowledge, the pattern mentioned above was not investigated to improve the results obtained by CBIR methods. Therefore, unlike related works, we present a computer program named CBIR-ANR, in which information retrieval is followed by an accuracy noise reduction strategy that adjusts query responses and consequently increases assertiveness in image retrieval. This strategy consists of grouping the outputs of the top images related to a query image until the required number of retrieved images is reached. In other words, the accuracy noise reduction approach is a loop in which retrieved images in the top positions of a given query image become query images on new searches. Then, the new-found images also become query images. This procedure is repeated until the required number of retrieved images is reached. In our experiments, half of the expected responses represent the number of top images required in the accuracy noise reduction approach. When query images have identical output, duplicate responses are deleted.

Furthermore, another novelty of the CBIR-ANR software is integrating a new combination of image descriptors. The software integrates local binary patterns (LBP) [9], local directional pattern variance (LDiPv) [10], and color micro-structure descriptor (cl-MSD) [11] descriptors for forming a 187-dimensional feature vector, which is size efficient for large-scale data sets. Besides, the software does not use training steps, is easy to use, as explained in Section 2, and is competitive with other content-based image retrieval methods, as presented in Section 3.

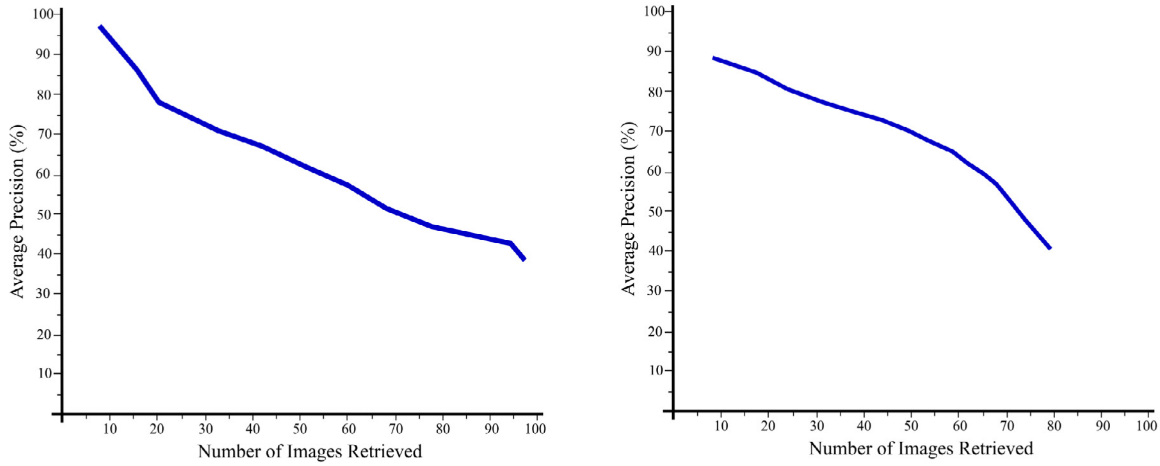
The main highlights of the software are:

- Coding of the relationship between the top positions of query images to reduce the accuracy error and increase the number of hits in image retrieval.
- Implementation and integration of LBP, LDiPv, and cl-MSD image descriptors.
- Performance evaluation with assertiveness measurement using precision and recall metrics.
- Visual presentation of the content search engine results.

2. Description of software

The software was implemented using MATLAB, a coding tool focused, among others, on developing mathematical solutions, statistical evaluations, image processing, and computer vision algorithms. Then, we use MATLAB to implement image loading functions, image descriptors, similarity evaluation between query and data set images, accuracy noise reduction, assertiveness analysis, and visualization of the results obtained by the software.

Initially, the user indicates the path to the data set, and the software calculates the image features according to the LBP, LDiPv, and cl-MSD image descriptors, resulting in a feature vector for each sample of the data set. Then, the user presents the query image, and the software also encodes the attributes of the query image in the form of a feature vector. In the next step, the software calculates the similarity between the query image's feature vector and the data set's feature vectors using L_1 distance. Then, the data set images with a minor numerical difference from the query images are returned in ascending order. After that, the software performs the accuracy noise reduction approach, replaces repeated image results, and calculates the assertiveness in precision and recall. Finally, statistical and visual results are presented.



(a) Adapted from [1]
(b) Adapted from [21]
(c) Adapted from [17]
(d) Adapted from [15]

Fig. 1. Pattern recognition in CBIR methods. As the number of retrieved images increases, assertiveness decreases proportionally.

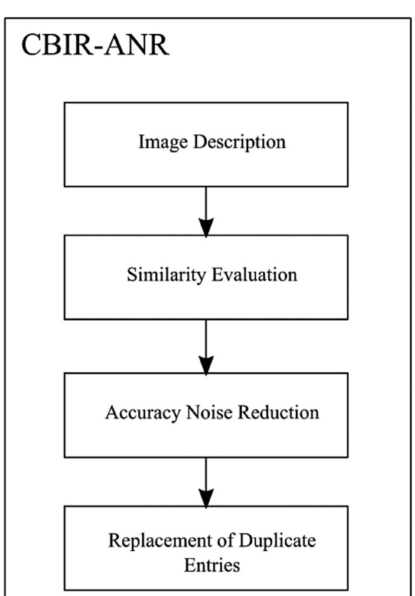


Fig. 2. Architecture of the CBIR-ANR software.

2.1. Software components

The CBIR-ANR is organized into four components corresponding to the software architecture, as presented in Fig. 2. The `Image Description` encodes the visual image attributes and prepares feature vectors according to the computation of image descriptors algorithms. The `Similarity Evaluation` compares the numerical information of the elementary characteristics of the image contents and sorts the results to point out the data set images that are more related to a query image. The `Accuracy Noise Reduction` updates the initial retrieved images and presents a new image arrangement considering the correlation between the top retrieved images. Then, the `Replacement of Duplicate Entries` identifies repeated images resulting from updating the initially retrieved images and replaces them with images ranked by the previous component but not referenced in the cluster of retrieved images.

The cl-MSD, LBP, and LDiPv are implemented in the `Image Description` component. The software receives the image data set path, computes the description of the images, and returns a 187-dimensional feature vector for each image sample. After that, the user can choose to use only one, two, or three descriptors. The 72-first position of the vector encodes the cl-MSD descriptor. From position 73 to 131, the LBP descriptor is encoded. Then, the remaining position

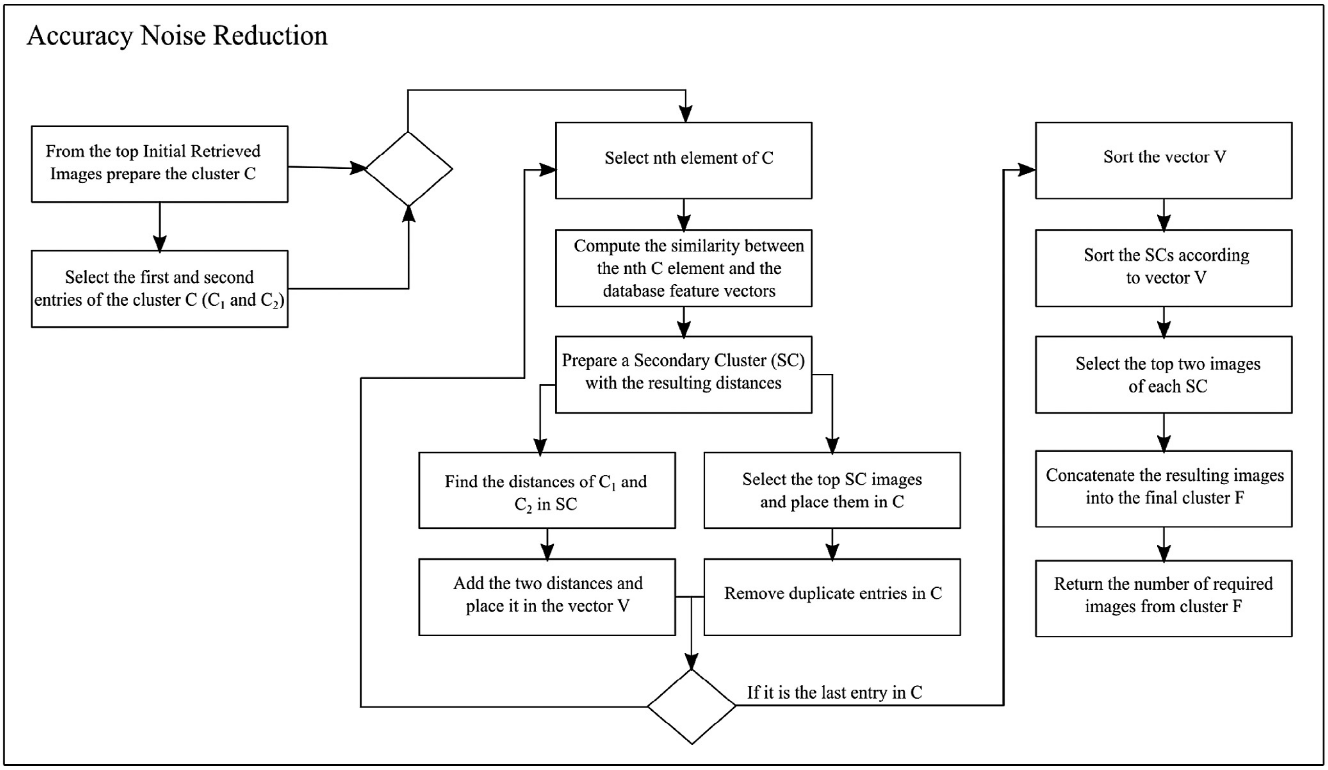


Fig. 3. Accuracy noise reduction strategy workflow.

from 132 to 187 encodes the LDiPv descriptor. The user can select the first 72 positions of the feature vector and perform the program using only the cl-MSD descriptor, or select the other positions, from 73 to 187, and execute the program using the LBP and LDiPv, or use the entire feature vector and use all the three descriptors, or try other combinations. Optionally, the code can be updated to call other image descriptors. The function responsible for the feature space construction can be easily adapted to call other image description functions. As the software is modular, the components are independent and cohesive to handle custom configurations and bespoke scenarios.

In the `Similarity Evaluation` component, the feature vector corresponding to the query image is compared to each sample of the image data set. First, the feature vectors are partitioned according to the number of image descriptors. The evaluation is performed individually between the descriptors, and the results are summed up and normalized. In other words, the process consists of comparing the cl-MSD descriptor of the query image with the cl-MSD descriptor of the images in the data set, the LBP descriptor of the query image with the LBP descriptor of the images in the data set, and the LDiPv descriptor of the query image with the LDiPv descriptor of the images in the data set. Then, the comparison between the cl-MSD descriptors is normalized by considering only the results of the cl-MSD. The same is applied to the LBP and LDiPv descriptors. After that, the outcomes of the cl-MSD, LBP, and LDiPv are added, and the summation results are divided by the number of image descriptors. This approach inhibits the influence of one image descriptor on the other so that the contribution of each of them has the same value. Then, the images are sorted and presented as the output of the process.

In the `Accuracy Noise Reduction` component, the retrieved images obtained with the similarity evaluation between the query image and the data set are updated based on the first top images of the initial retrieved images. In this context, 'top' means the number of images in the database most similar to a query image defined by a user parameter, which must be between 1 and the number of images in the data set. For example, this number could be 5, 10, 12, 20, or any other arbitrary number. The similarity between each top image and the data set is computed, resulting in other image clusters, each with its respective top images. The process is repeated until all images listed in the first positions of the clusters are treated as query images. Then, the clusters are sorted according to their similarity with the initial retrieved images, and the first two positions of each are concatenated to form an updated version of the retrieved images. Due to the potential similarity between the first positions of the image clusters, this strategy may present repeated images in the updated version of the initial retrieved images. For this reason, the `Replacement of Duplicate Entries` component is responsible for identifying the occurrence of repeated images and replacing them with other images from the image clusters. It is an extension of component `Accuracy Noise Reduction`, detailed in Section 2.2.

2.2. Accuracy noise reduction strategy

In a CBIR system, the feature space construction and the similarity evaluation are well-known steps that result in a list of retrieved images ranked by the verisimilitude between image contents. In addition to these steps, our software rearranges the initial retrieved images into a new step to increase assertiveness in the image retrieval process. The software performs the accuracy noise reduction strategy, which consists of updating the position of the retrieved images based on the images with the highest affinity to the query image. Fig. 3 shows the workflow corresponding to updating the initially retrieved images.

First, a new image cluster C is obtained from the top initial retrieved images. In this case, the software selects only half of the number of images required by the user as top images. In the final steps, there is a concatenation between the first two images of secondary clusters so that a final cluster of the same size as required by the user is obtained. For this reason, only half of the first images are used in the processing steps of the accuracy noise reduction strategy. After preparing cluster C, the first two images are selected (C_1 and C_2). Following that, a loop starts where each element of C is treated as a query image, and

Table 1
Comparative results between the CBIR-ANR software and related work in different image databases. In the experimental test, the number of retrieved images is 12.

Corel-1K							
	Wang and Wang [12]	Liu et al. [11]	Feng et al. [13]	Dawood et al. [14]	Raza et al. [15]	Niu et al. [3]	CBIR-ANR
ARP	61.58	72.17	72.85	78.54	80.79	82.24	**84.38**
ARR	7.39	8.66	9.39	9.42	9.69	9.87	**10.13**
Corel-5K							
	Hua et al. [16]	Chu and Liu [2]	Dawood et al. [14]	Liu and Wei [17]	Wei and Liu [4]	Niu et al. [3]	CBIR-ANR
ARP	60.13	60.16	63.14	63.50	66.91	67.93	**72.86**
ARR	7.21	7.21	7.54	7.62	8.03	8.15	**8.74**
Corel-10K							
	Hua et al. [16]	Dawood et al. [14]	Chu and Liu [2]	Liu and Wei [17]	Wei and Liu [4]	Niu et al. [3]	CBIR-ANR
ARP	48.58	50.2	52.96	53.19	56.88	58.52	**63.31**
ARR	5.83	6.03	6.36	6.38	6.83	7.02	**7.60**
GHIM-10K							
	Chu and Liu [2]	Liu [18]	Liu [19]	Liu et al. [20]	Zhou et al. [21]	Niu et al. [3]	CBIR-ANR
ARP	56.48	56.68	57.51	61.16	67.10	68.50	**71.23**
ARR	1.36	1.36	1.38	1.47	1.61	1.65	**1.71**

Fig. 4. An example of the CBIR-ANR software where all retrieved images are from the same category as the query image. In retrieved image captions, 'class' means the image query category, and 'y' is the category of the retrieved image.

its content is compared to the image data set. Then, the similarity evaluation returns a retrieved image list named Secondary Cluster (SC). The images C_1 and C_2 are located in SC, and their distances are added and placed in vector V. Also, the top images in SC are concatenated to cluster C, which dynamically increases the size of C. Repeated images in C are removed, and the loop continues to run until all entries in C are processed. When the loop stops, the vector V is sorted, and the Secondary Clusters (SCs) are reorganized according to V. Then, the first two images of each Secondary Cluster are selected and concatenated to form the final cluster F.

2.3. Visual results

Fig. 4 presents image retrieval results in which all outcomes obtained with the CBIR-ANR software are from the same category as the query image. On the other hand, Fig. 5 describes a case where one output diverges from the query image category, resulting in a precision of 0.92. Images not fitting the expected category are highlighted with a red bounding box. In these examples, the number of images to retrieve is 12.

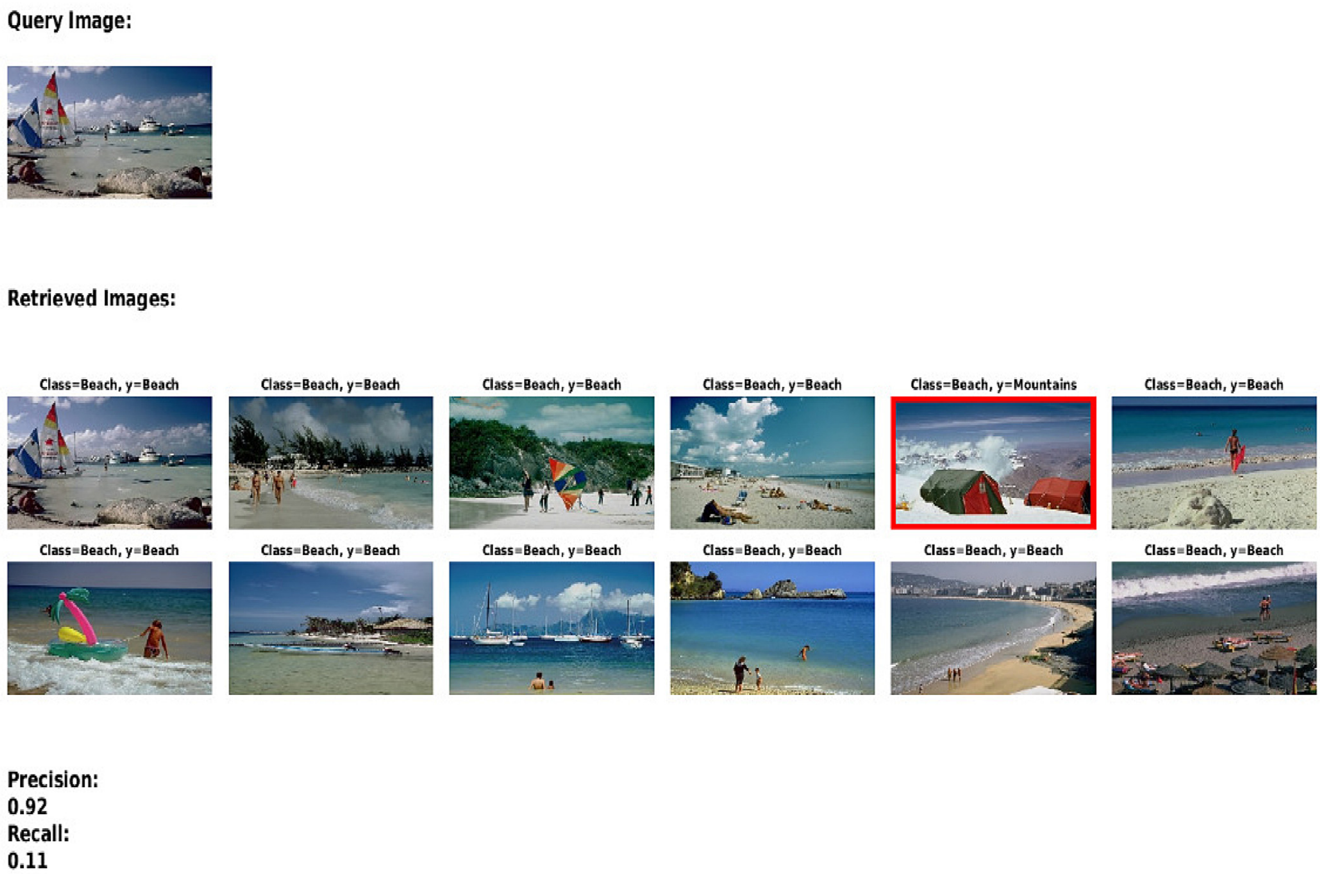


Fig. 5. An example of the CBIR-ANR software in which one retrieved image differs from the query image category. In retrieved image captions, 'class' means the image query category, and 'y' is the category of the retrieved image.

Table 2
Image descriptors integration and the accuracy noise reduction (ANR) strategy. In the experimental test, the number of retrieved images is 12.

	Corel-1K		Corel-5K		Corel-10K		GHIM-10K	
	ARP	ARR	ARP	ARR	ARP	ARR	ARP	ARR
LBP	63.53	7.62	39.57	4.75	31.66	3.80	45.90	1.10
LBP+ANR	**66.87**	**8.02**	**41.77**	**5.01**	**33.76**	**4.05**	**47.16**	**1.13**
LDiPv	58.67	7.04	35.20	4.22	27.80	3.34	39.52	0.95
LDiPv+ANR	**60.45**	**7.25**	**36.83**	**4.42**	**29.03**	**3.48**	**39.87**	**0.96**
cl-MSD	73.78	8.85	55.35	6.64	46.12	5.53	53.17	1.28
cl-MSD+ANR	**77.17**	**9.26**	**61.53**	**7.38**	**51.41**	**6.17**	**58.17**	**1.40**
LBP+LDiPv	69.03	8.28	48.23	5.79	38.61	4.63	52.03	1.25
LBP+LDiPv+ANR	**71.70**	**8.60**	**51.07**	**6.13**	**41.38**	**4.97**	**54.06**	**1.30**
LBP+cl-MSD	78.35	9.40	63.63	7.64	53.89	6.47	63.24	1.52
LBP+cl-MSD+ANR	**82.88**	**9.95**	**69.91**	**8.39**	**59.92**	**7.19**	**68.74**	**1.65**
LDiPv+cl-MSD	78.68	9.44	61.17	7.34	51.88	6.23	60.25	1.45
LDiPv+cl-MSD+ANR	**81.51**	**9.78**	**66.97**	**8.04**	**57.67**	**6.92**	**65.14**	**1.56**
LBP+LDiPv+cl-MSD	80.62	9.67	66.36	7.96	57.00	6.84	66.16	1.59
LPB+LDiPv+cl-MSD+ANR	**84.38**	**10.13**	**72.86**	**8.74**	**63.30**	**7.60**	**71.23**	**1.71**

3. Impact overview

The CBIR-ANR software is competitive with other information retrieval methods in different assessment contexts. In this sense, we use precision and recall metrics to measure the assertiveness of the software. The first is the ratio between the hits achieved by the number of expected correct answers. The second one consists of the ratio between the hits achieved by the total number of images in the data set that belong to the same category as the query image. Then, the average retrieval precision (ARP) and average retrieval recall (ARR) are obtained by considering the average results of all image categories in the data set. Additionally, we use four data sets for evaluation purposes: Corel-1K [22], Corel-5K [11], Corel-10K [11], and GHIM-10K [20]. The CBIR-ANR is superior to several recently published works in the comparative analysis, as shown in Table 1.

Furthermore, we present the potential of the software considering the use of image descriptors individually and in combinations to compare their results with the subsequent application of the accuracy noise reduction (ANR) approach. Table 2 shows the increase in assertiveness when applying the ANR strategy for all cases presented. It is worth mentioning that the CBIR-ANR software uses the LPB+LDiPv+cl-MSD+ANR combination. Also, Table 2 shows the importance of descriptor integration, such as in the arrangement of the LBP, LDiPv, and cl-MSD descriptors that achieved the best result.

As presented in Section 2, the CBIR-ANR software is organized into four components responsible for feature space construction, similarity evaluation, accuracy noise reduction, and replacement of duplicate entries. Execution time may vary depending on the software component under observation. Some components have more processing steps and require more execution time than others. For example, encoding visual image attributes into feature vectors involves steps that consume more resources than other processes. On the other hand, accuracy noise reduction and image replacement are lightweight processes that are performed in a short time. In this sense, we measured the temporal performance of the four steps separately. Table 3 presents the average CPU time (μ) and the standard deviation (σ) for each of the steps in the image retrieval process. As can be seen, the execution time (in seconds) was stable considering different databases where the construction of image descriptors consumed most of the time. The longer processing

Table 3
Temporal performance (in seconds) of the CBIR-ANR software. In the experimental test, we use the LPB+LDiPv+cl-MSD+ANR combination.

		Image description	Similarity evaluation	Accuracy noise reduction	Replacement of duplicate entries	Total
Corel-1K	μ	1.7175	0.0988	0.0004	0.0003	**1.8170**
	σ	0.1642	0.0086	0.0004	0.0005	**0.1737**
Corel-5K	μ	0.4548	0.5252	0.0034	0.0015	**0.9849**
	σ	0.0615	0.0167	0.0010	0.0006	**0.0798**
Corel-10K	μ	0.4269	1.0245	0.0073	0.0031	**1.4618**
	σ	0.0336	0.0230	0.0021	0.0012	**0.0599**
GHIM-10K	μ	2.0767	1.0146	0.0077	0.0031	**3.1020**
	σ	0.0519	0.0250	0.0020	0.0012	**0.0818**

time of the GHIM-10K data set is justified by the size of the target images, which are larger than in the other data sets. A notebook with Core i7-9750H (2.6 GHz; 12 MB Cache) and 16 GB RAM was used to perform the experimental tests.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank the Brazilian agency CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior [CAPES Financial Code #001] for supporting this work. We acknowledge the support of the Universidade Federal de Goiás (Brazil) and Instituto Federal Goiano (Brazil). We also thank Dr. Guang-Hai Liu, who shared the Corel-10K dataset.

References

[1] S. Lee, M. Masoud, J. Balaji, S. Belkasim, R. Sunderraman, S.-J. Moon, A survey of tag-based information retrieval, Int. J. Multimedia Inf. Retr. 6 (2) (2017) 99–113.
[2] K. Chu, G.-H. Liu, Image retrieval based on a multi-integration features model, Math. Probl. Eng. 2020 (2020).
[3] D. Niu, X. Zhao, X. Lin, C. Zhang, A novel image retrieval method based on multi-features fusion, Signal Process., Image Commun. 87 (2020) 115911, http://dx.doi.org/10.1016/j.image.2020.115911.
[4] Z. Wei, G.-H. Liu, Image retrieval using the intensity variation descriptor, Math. Probl. Eng. 2020 (2020).
[5] N. Ali, K.B. Bajwa, R. Sablatnig, S.A. Chatzichristofis, Z. Iqbal, M. Rashid, H.A. Habib, A novel image retrieval based on visual words integration of SIFT and SURF, PLoS One 11 (6) (2016) e0157428.
[6] L. Yu, N. Liu, W. Zhou, S. Dong, Y. Fan, K. Abbas, Weber's law based multi-level convolution correlation features for image retrieval, Multimedia Tools Appl. 80 (13) (2021) 19157–19177.
[7] M. Verma, B. Raman, Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval, Multimedia Tools Appl. 77 (10) (2018) 11843–11866.
[8] J. Pradhan, A.K. Pal, H. Banka, A CBIR system based on saliency driven local image features and multi orientation texture features, J. Vis. Commun. Image Represent. 83 (2022) 103396, http://dx.doi.org/10.1016/j.jvcir.2021.103396.
[9] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[10] M.H. Kabir, T. Jabid, O. Chae, A local directional pattern variance (LDPv) based face descriptor for human facial expression recognition, in: 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, 2010, pp. 526–532.
[11] G.-H. Liu, Z.-Y. Li, L. Zhang, Y. Xu, Image retrieval based on micro-structure descriptor, Pattern Recognit. 44 (9) (2011) 2123–2133.
[12] X. Wang, Z. Wang, A novel method for image retrieval based on structure elements' descriptor, J. Vis. Commun. Image Represent. 24 (1) (2013) 63–74.
[13] L. Feng, J. Wu, S. Liu, H. Zhang, Global correlation descriptor: a novel image representation for image retrieval, J. Vis. Commun. Image Represent. 33 (2015) 104–114.
[14] H. Dawood, M.H. Alkinani, A. Raza, H. Dawood, R. Mehboob, S. Shabbir, Correlated microstructure descriptor for image retrieval, IEEE Access 7 (2019) 55206–55228.
[15] A. Raza, H. Dawood, H. Dawood, S. Shabbir, R. Mehboob, A. Banjar, Correlated primary visual texton histogram features for content base image retrieval, IEEE Access 6 (2018) 46595–46616.
[16] J.-Z. Hua, G.-H. Liu, S.-X. Song, Content-based image retrieval using color volume histograms, Int. J. Pattern Recognit. Artif. Intell. 33 (11) (2019) 1940010.
[17] G.-H. Liu, Z. Wei, Image retrieval using the fused perceptual color histogram, Comput. Intell. Neurosci. 2020 (2020).
[18] G.-H. Liu, Content-based image retrieval based on cauchy density function histogram, in: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery, ICNC-FSKD, IEEE, 2016, pp. 506–510.
[19] G.-H. Liu, Content-based image retrieval based on visual attention and the conditional probability, in: International Conference on Chemical, Material and Food Engineering, Atlantis Press, 2015, pp. 843–847.
[20] G.-H. Liu, J.-Y. Yang, Z. Li, Content-based image retrieval using computational visual attention model, Pattern Recognit. 48 (8) (2015) 2554–2566.
[21] J. Zhou, X. Liu, W. Liu, J. Gan, Image retrieval based on effective feature extraction and diffusion process, Multimedia Tools Appl. 78 (5) (2019) 6163–6190.
[22] J.Z. Wang, J. Li, G. Wiederhold, SIMPLIcity: Semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Mach. Intell. 23 (9) (2001) 947–963.