big data and cognitive computing

MDPI

Article

A New Comparative Study of Dimensionality Reduction Methods in Large-Scale Image Retrieval

Mohammed Amin Belarbi [1,*], Saïd Mahmoudi [1], Ghalem Belalem [2] and Sidi Ahmed Mahmoudi [1] and Aurélie Cools [1]

[1] Faculty of Engineering, University of Mons, 7000 Mons, Belgium; said.mahmoudi@umons.ac.be (S.M.); sidi.mahmoudi@umons.ac.be (S.A.M.); aurelie.cools@umons.ac.be (A.C.)

[2] Department of Computer Science, Faculty of Exact and Applied Sciences, University of Oran1 Ahmed Ben Bella, Oran 31000, Algeria; ghalem1dz@gmail.com

* Correspondence: belarbi.mohammed.amin@gmail.com

Abstract: Indexing images by content is one of the most used computer vision methods, where various techniques are used to extract visual characteristics from images. The deluge of data surrounding us, due the high use of social and diverse media acquisition systems, has created a major challenge for classical multimedia processing systems. This problem is referred to as the 'curse of dimensionality'. In the literature, several methods have been used to decrease the high dimension of features, including principal component analysis (PCA) and locality sensitive hashing (LSH). Some methods, such as VA-File or binary tree, can be used to accelerate the search phase. In this paper, we propose an efficient approach that exploits three particular methods, those being PCA and LSH for dimensionality reduction, and the VA-File method to accelerate the search phase. This combined approach is fast and can be used for high dimensionality features. Indeed, our method consists of three phases: (1) image indexing within SIFT and SURF algorithms, (2) compressing the data using LSH and PCA, and (3) finally launching the image retrieval process, which is accelerated by using a VA-File approach.

Keywords: indexing image; VA-File; PCA; LSH; binary tree; SIFT; SURF; image retrieval

check for updates

Citation: Belarbi, M.A.; Mahmoudi, S.; Ghalem B.; Mahmoudi, S.A.; Cools, A. A New Comparative Study of Dimensionality Reduction Methods in Large-Scale Image Retrieval. *Big Data Cogn. Comput.* **2022**, *6*, 54. https://doi.org/10.3390/bdcc6020054

Academic Editors: Michael Alexander Riegler and Pål Halvorsen

Received: 19 April 2022
Accepted: 4 May 2022
Published: 13 May 2022

1. Introduction

In recent years, many applications have emerged quickly due to the high use of social media, such as Facebook and Twitter. Several multimedia applications contain an extremely high volume of multimedia data (sound, video, and images). These applications are used in different fields, including video surveillance, pattern recognition, and medical applications. The main challenge is to have quick access to this huge dataset in order to index the data in a reasonable time. In the field of image processing, the most used methods for navigating large databases are the famous content-based image retrieval (CBIR) methods [1]. These methods are generally based on three phases: feature extraction, similarity measurement, and the search phase, which must be in real time. CBIR methods consist of automatically detecting and extracting visual features from images such as global and local features by applying some means of image processing algorithms [2]. CBIR systems extract the features of the users image (query) [3], which are then compared to the database's image features (indexing phase). The result of this process is a list of the most similar images to the users' image (query).

In this context, reducing the dimension of the features is a critical step when using high dimension data [4]. The idea is to minimize the dimension in order to speed up the search phase, as well as to be able to read this data with a normal computer. For that reason, dimensionality reduction methods are essential to overcoming the 'curse of dimensionality' [5–7].

We propose in this paper a new approach for image retrieval systems, achieving a fast CBIR system with dimensionality reduction and a fast search approach. First, we use

SIFT and SURF algorithms to extract features. The result of these algorithms is a matrix with high dimension. Then, two methods, PCA and LSH, are exploited to minimize the dimensions of these features. Finally, the VA-File algorithm is used to speed up the search phase. This paper is organized as follows: some related works in the field of large-scale image retrieval and dimensionality reduction are shown in Section 2. Our approach is explained in Section 3, while Section 4 analyzes the performance of our results. Finally, we present the conclusions in the last Section 5.

2. Related Work

In this section, we present the related works linked to the different methods used. Since our approach is related to three main domains, those being CBIR, the 'curse of dimensionality', and indexing, we propose a related work section divided into three parts, treating each of these domains in turn.

2.1. CBIR (Content-Based Image Retrieval) System

The CBIR framework is a computer system used for visualizing, finding through, and retrieving images from a huge database of digital images. In this domain, diverse works have been proposed for the domains of commerce, government, academia, and hospitals, where large collections of digital images are created [1]. Many of these collections are the product of digitizing existing collections of analog photographs, diagrams, drawings, paintings, and prints [1]. Usually, the only way of searching through these digital data is by using a keyword, or just by browsing. Moreover, digital image data sets have opened the approach to content-based searching [8].

On the other hand, the authors in [9] used an image match criteria and a system that uses spatial information and visual features represented by dominant wavelet coefficients [1]. Although their system provides improved matching over image distance norms, it does not support any index structure. In fact, they mainly focused on efficient feature extraction by using wavelet transformation rather than an index structure to support speedy retrieval [1].

The QBIC system proposed in [10] presents one of the most notable techniques, developed by IBM, for querying by image content [1]. This allows the user to compose a query based on a variety of different visual properties, such as color, shape, and texture, which are semi-automatically extracted from the images. This method partially uses R*-tree as an image indexing method [1,11,12].

2.2. Dimensionality Reduction

Various methods of dimensionality reduction, such as PCA and LSH, can be found in the literature. For example, for the PCA approach, the authors in [13] proposed a method allowing the vector of locally aggregated descriptors (VLAD) to be improved. Their approach is based on a linear discriminant analysis (LDA) method to reduce the dimensionality of the VLAD descriptor. They used the nearest neighbor distance ratio to choose the nearest set so that the correspondence between a feature and the set was more stable [14].

The authors in [15] presented a new approach based on the random projection of the dimensionality reduction of high-dimensional datasets. The criteria used in this approach were the amount of distortion caused by the method and its computational complexity. Their results indicate that the projection still has a fast computing time [14].

In [16], the authors decided to reduce the dimensionality of their generated features from 170,000 to 20. The result obtained showed that a small output dimensionality is sufficient for a small number of object classes, in this case. These techniques showed excellent performance for text and facial features [14,17–19].

Recently with the apparition of CNN (convolutional neural network), the authors in [20] presented and showed three dimensionality reduction methods called regional maximum activation of convolutions (RMAC), RMAC+ and Multi-scale regional maximum activation of convolutions (MS-RMAC). They also explained and detailed the best position of each method layer that fit with CNN architecture. The results obtained show the efficiency of the use of dimensionality reduction methods, even with CNN architecture.

The authors in [21] proposed an approach based on CNN architecture Inception V4 as the backbone network to extract the deep features of the feature maps generated from the first reduction block of Inception V4 through using 5 × 5 convolutional kernels that are extracted and reorganized.

In the following section, we present the state of the art related to the LSH method, which is used to compress and reduce the dimensions of features. Several works have used the LSH algorithm, such as [22], in which the authors developed a new approach based on the Fisher vector. Their approach uses different methods of hashing, such as LSH and SH, in order to reduce the Fisher vector. First, they extracted the Fisher vector, as its size depends on the datasets of images. In this case, the LSH function was applied to reduce the size of this vector. The experiments were applied to two datasets: holidays (holidays data sets: http://lear.inrialpes.fr/jegou/data.php, accessed on 3 May 2022) and the benchmark of the University of Kentucky (https://www.uky.edu/iraa/benchmark-institutions, accessed on 3 May 2022). The obtained results show that the Fisher vector is good for image retrieval but has a high dimension. On the other hand, the combination between LSH and SH for high dimensions allowed the results to be improved.

The authors in [23] presented a new supervised approach of hashing for image retrieval. This approach adapted the image representation compared to the hashing functions. The proposed method first divided the matrix of semantic similarity into two values of approximate hash codes for the learning image. After that, they applied a convolutional neural network to learn this value and used the MNIST dataset. The obtained results showed the interest of using hash functions when the dimensions of the feature vectors are very large.

2.3. *Approximation Indexing Methods*

Approximation indexing methods are generally used to deal with high-dimension problems. We can use multidimensional indexing techniques by proposing an indexing method which makes an approximation of the space. In the literature, we can find many works that use these methods, such as in [15], where the authors provided a new approach, which is very efficient for high-dimensional data based on edges. The experiment's results showed that their approach was more efficient and dynamic compared to the pyramid technique.

In light of the above-mentioned problem, in this work, we propose an approach based on PCA and LSH for dimensionality reduction. The VA-File method is then used to accelerate the search phase, which is necessary for the main goal of this paper.

3. Proposed Method

The main objective of our method is to index images within a system that allows users to perform a fast and accurate search within a set of images. Our approach is based on three phases. First, a features extraction process is applied by using the SIFT and SURF methods. Secondly, we apply PCA and LSH as dimensionality reduction methods in order to reduce the dimensions of any features [1]. Finally, we use the VA-File method in order to speed up the search engine phase. We combined PCA, LSH and VA-File and compared this combination with other combinations. We kept the best combination that maintained the same accuracy. We can resume our approach within the schema shown in Figure 1.

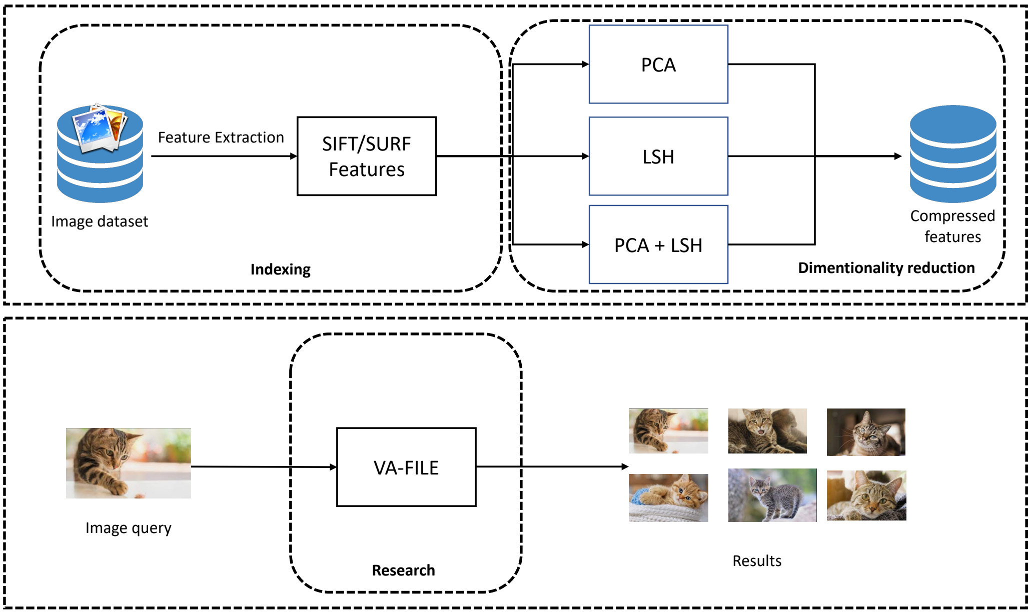


Figure 1. The general architecture of our proposed approach.

3.1. Pre-Processing

As mentioned in our previous work, the best filter algorithm that matches with both SIFT or SURF algorithms is the Gaussian blur. It is a kind of image blurring filter that uses a Gaussian function for calculating the transformation applied to each pixel in the image [1,24]. We can resume the process of applying the Gaussian filter (pretreatment) as in Algorithm 1.

Algorithm 1: Pretreatment

```
Ensure: $jpg\_files$:Image data sets;
  $jpg\_files \leftarrow \emptyset$;
  Read $Nbr - of - image$;
  for $j$ : 1 to $Nbr - of - image$ do
    $img\_j \leftarrow Get\_The\_Image(j)$;
    $img\_GB_j \leftarrow \textbf{GaussianBlur}(img\_j)$;
    $jpg\_files \leftarrow jpg\_files \cup img\_GB_j$;
  end for
  return $jpg\_files$;
```

The input of Algorithm 1 is a set of images. Once the algorithm is applied, we can calculate the Gaussian blur for all the images and store them in a new folder. This output is then used as the input for the next step.

3.2. Features Extraction

For features extraction, we use the SIFT and SURF algorithms for the image features. These algorithms allow points of interest, also called key points, to be extracted [25]. Each image descriptor can then be compared, or matched, to the descriptors extracted from other images [1].

1. SIFT descriptor: The SIFT descriptor presented in [26] provides a solution for indexing images [27]. In our case, we use OpenCV (OpenCV: https://opencv.org/, accessed on 3 May 2022) as the library to calculate the SIFT descriptor [27]. The result of this descriptor is presented by a matrix of n line and 128 column [1].
2. SURF descriptor:
   The SURF descriptor is based on the SIFT descriptor. In this case, we also use the

OpenCV library for calculating the SURF descriptor. The result of applying this descriptor is a matrix of n lines and 64 columns. Note that the SURF method is faster than the SIFT method [1].

The comparison between SIFT and SURF features is done by the use of a similarity measure, such as the following:

1. Brute-force matcher: one of the famous methods that can be used with the SIFT feature descriptor for image comparison. The algorithm takes the descriptor of one feature from the first set and matches it with all the other features in the second set by using distance calculations, and the closest one is returned [1,14].
2. FLANN-based matcher: FLANN [28] uses the approximation of the nearest neighbors methods. It is faster than the brute-force matcher for big data. The FLANN measure uses the K-means [29] for generic feature matching [1].

3.3. Dimensionality Reduction

In this paper, we use the dimensionality reduction methods PCA [1,30] and LSH to minimize the dimension of features.

3.3.1. Principal Component Analysis (PCA)

The main objective of our approach is to provide a solution that minimizes the dimension of the features. The first method used is PCA. In our previous work, we found that the best compression ratio that matched the SIFT and SURF features, without a loss in precision, was 70% [1]. The compression dimension of SIFT was calculated as shown in Equation (1):

$$Ncompression = 128 - (N * 128/100) \tag{1}$$

The compression dimension of SURF is calculated as shown in (2) [14]:

$$Ncompression = 64 - (N * 64/100) \tag{2}$$

where N is between 10 and 90 [14].

3.3.2. Locality Sensitive Hashing (LSH)

LSH presents a solution that allows the 'curse of dimensionality' to be overcome. The principle idea of LSH is the use of many functions of hashing with the same point. The steps of applying this method are as follows:

1. Pre-treatment (indexation): The algorithm uses l different hashing functions. A hashing function f is obtained by a concatenation of randomly chosen i hashing functions. After that, the algorithm creates l hashing tables, where each table corresponds to a hash function. Each table i contains the result of the hash function f_i. No empty tables are kept.
2. Search for query (q): The algorithm uses the same hashing functions, defined in Section 1, to calculate the hash of this query image. We compare the results within the l hashing tables, and find the hash points in the same position as in query q. The breakpoint of the algorithm is when a point is found in the table.

3.4. Vector Approximation File (VA-FILE)

The VA-File is an approximation method that can be used for indexing. It offers good performance for the 'curse of dimensionality', and it is based on the compression of data. The principle idea of this approach is as follows:

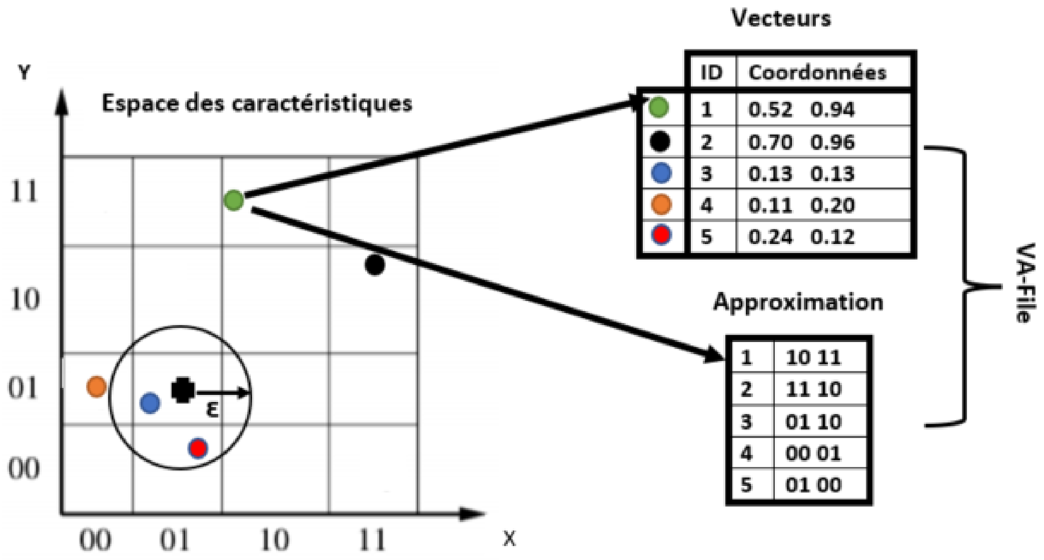
1. The algorithm uses two files. The first contains all the vectors of the database. In our case, the vectors are the SIFT or SURF features. The second file contains the approximation geometric of these vectors, as shown in Figure 2.

2. For the first query, the approximation file is sequentially reddened in order to select all the vectors that can be integrated into the final result. After that, the access to the data file is based on the result of the first step.
3. Then, the sequential search is performed on the approximation file (small size), which makes the search very fast, since it is based on a reduced subset of vectors, as shown in Figure 3.
4. To perform a KNN search using a VA-File, there are two methods: the simple scan algorithm (VA-SSA) or the near optimal algorithm (VA-NOA). In our case, we used VA-SSA, as shown in Algorithm 2.

Algorithm 2 needs to apply a first query to calculate the distance between this query and all the points in the vector features. After that, the algorithm can just compare this distance with the new query. If the distance is less than that calculated with the first query, then we can update the distance and add this point to the similar points.

Algorithm 2: Algorithm VA-NOA

```
Require: $\gamma$ distance ;
Ensure: vi the approximation vector, vd: file of distance, req1: the first query, req: a
  normal query, dist1: the distance function between req1 and vi, dist2: the distance
  function between req and req1;
  for all the points of the vector vi do
    $\gamma$ = max(dist1(req1,vi))
    vd += dist1(req1,vi)
  end for
  for all i $\in$ all points of vector vi do
    if dist2(req1,req) < $\gamma$ then
      if dist2(req,req1)<$\gamma$ then
        add the point $i$ with the similar points
        update vd
        $\gamma$ = max(vd)
      end if
    end if
  end for
  return the similar points;
```



	ID	Coordonnées
🟢	1	0.52 0.94
●	2	0.70 0.96
🔵	3	0.13 0.13
🟠	4	0.11 0.20
🔴	5	0.24 0.12

1	10 11
2	11 10
3	01 10
4	00 01
5	01 00

Figure 2. Geometric representation of VA-File.

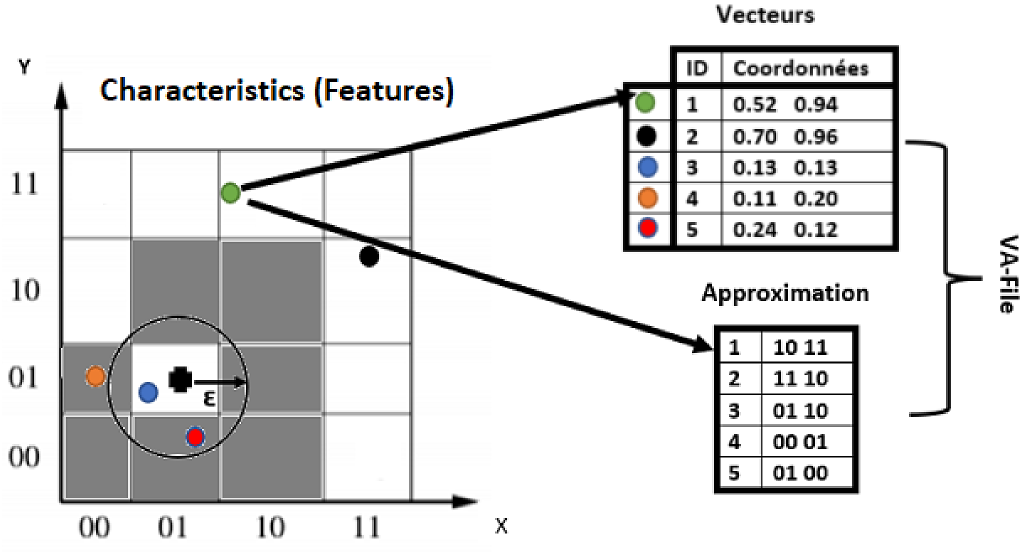


Figure 3. The research phase with VA-File.

4. Experimental Results and Analysis

Our approach is evaluated by using the recall and precision metrics.

4.1. The Evaluation Metric

In CBIR, the recall and precision graph are one of the most famous metrics used for the evaluation. Calculating recall precision needs at least three queries. Based on these queries, we can apply the following equations:

$$\text{Recall} = \frac{\text{correct of positive number}}{\text{all positive of number}} \tag{3}$$

$$\text{Precision} = \frac{\text{correct of positive number}}{\text{all matches of number}} \tag{4}$$

4.2. Experimental Results

In our approach, we use several image data sets to demonstrate the effect of our approach within the problem of large-scale image retrieval. Indeed, two different image databases are used, including *Wang* (Wang: http://wang.ist.psu.edu/docs/related/, accessed on 3 May 2022) which contains 10,000 images and 100 classes, and *ImageNet* (ImageNet: http://www.image-net.org/, accessed on 3 May 2022) which contains 10,000 classes and 16 millions of images. All the experiments are implemented by using the following hardware:

- CPU: Intel XEON with 64 threads.
- GPU: 4 × GTX 1080ti with 12 GB of RAM.
- RAM: 128 GB.

The proposed approach shown in this paper is based on four phases: features extraction, data compression within PCA and LSH, the combination between PCA and LSH, and the acceleration phase within VA-File. First, the 64 dimensional SURF and 128 dimensional SIFT features are extracted from all the images. After that, we apply PCA and LSH as dimensionality reduction methods. These steps are taken, as the goal of this work is to provide a CBIR system which allows us to minimize the dimensionality of image features and accelerate the search phase without any loss of precision of the system. The VA-File method is used to accelerate the search step. Table 1 and Figure 4 show the search time of an image query and the computation time of these algorithms within the Wang database.

Table 1. Computation (C.T) and the search time (S.T) within Wang database.

Methods	S.T (ms)	C.T (ms)
PCA (70%) [1]	168	460
PCA (70%) [1]/VA-FILE	146	460
LSH	164	570
PCA (70%) [1]/LSH	159	511
PCA (70%) [1]/LSH/VA-FILE	132	511

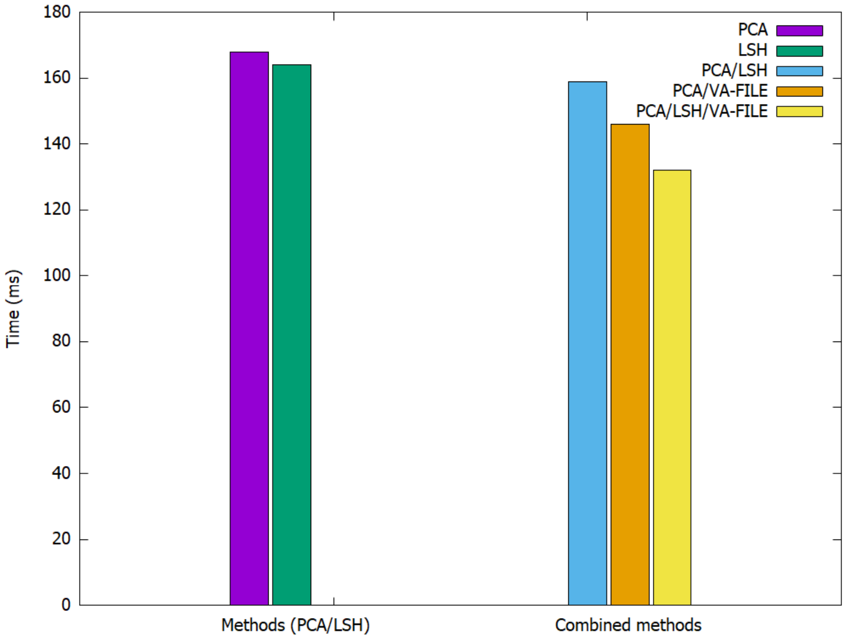


Figure 4. Computation and the research time within Wang database.

Within the ImageNet datasets, we obtained the results shown in Table 2 and the Figure 5.

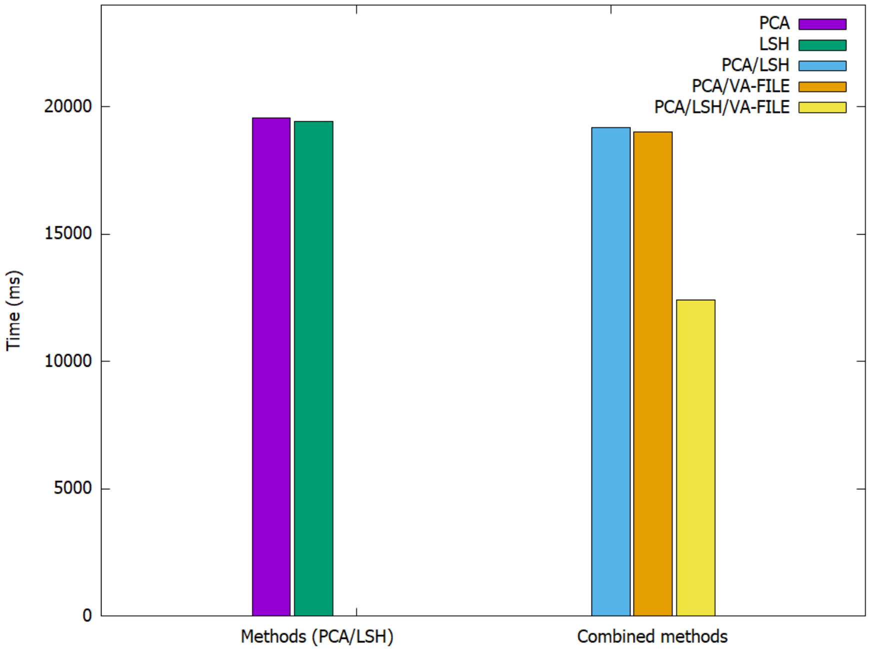


Figure 5. Computation and the search time within ImageNet database.

Table 2. Computation and the search time within ImageNet database.

Methods	S.T (ms)	C.T (ms)
PCA (70%) [1]	19,561	56,700
PCA (70%) [1]/VA-FILE	19,010	56,700
LSH	19,421	59,812
PCA (70%) [1]/LSH	19,171	59,004
PCA (70%) [1]/LSH/VA-FILE	12,423	59,004

As shown in Tables 1 and 2, the combination between PCA and VA-FILE is faster than PCA, which is used here with a reduction of 70% as in [1]. LSH is not faster because it depends on the number of hashing functions. However, when we use the result of applying PCA with a reduction of 70%, we achieve some acceleration of the search phase. The combination of the three methods, PCA, LSH and VA-File, presents the best combination that allows us to speed up the search phase with a factor ranging from 30% to 40%. Furthermore, we are able to lower the required storage space as a result of the compression, with a reduction ranging from 60% to 70% of SURF and SIFT features.

As shown Figure 6, the application of both PCA and LSH are better than LSH alone. This is due to the fact that the application of LSH after PCA within a range of 70% does not degrade information, because the result of PCA is a matrix with less information.

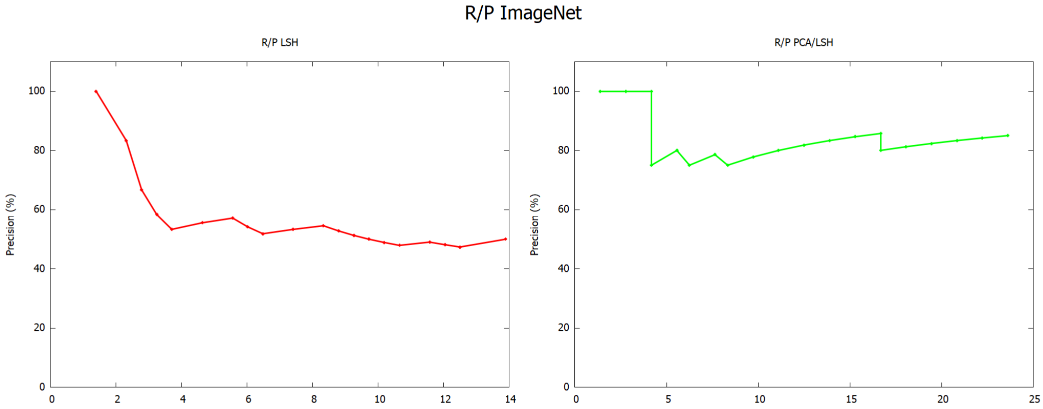


Figure 6. Recall/precision within ImageNet database.

The computation time, which is not high in the case of CBIR methods, is better still with the combination of PCA and VA-File because the use of either LSH or PCA/LSH/VA-File requires a lot of material resources because they require a lot of calculations. This is especially the case for LSH.

5. Conclusions

The main objective of this paper was to bring a CBIR system allowing us to minimize the dimensionality of image features and also to accelerate the search phase without any loss in precision. We showed and evaluated the experimental results acquire by using PCA and LSH for the dimensionality reduction of large scale images, and VA-File as an acceleration method for the search phase. Indeed, we analyzed the effect of the combination of different methods in order to find the best combination based on the results of the experiment. The experimental results show that we obtained a speed up of 30% to 40% of the search time, and a speed up of 60% to 70% of the memory storage space, because both PCA and LSH are dimensionality reduction methods. Moreover, the experimental results show that these kinds of algorithms need a high resource of materials. Finally, the results obtained within our approach guarantee positively the efficacy of our work.

As future works, we envisage to ameliorate our system by using the features obtained within the deep learning algorithms in order to validate our approach for both classical and deep learning features and also use the appropriate methods that fit with deep learning features.

Author Contributions: Conceptualization, M.A.B.; methodology, M.A.B.; formal analysis, M.A.B.; investigation, M.A.B.; writing—original draft preparation, M.A.B.; writing—review and editing, S.M. and S.A.M. and A.C.; visualization, M.A.B.; supervision, G.B. and S.M.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable

Informed Consent Statement: Not applicable.

Data Availability Statement: Wang: http://wang.ist.psu.edu/docs/related/ (accessed on 3 May 2022), ImageNet: http://www.image-net.org/ (accessed on 3 May 2022).

Acknowledgments: The authors would like to express their gratitude to academic editors and publishers the edit of this paper thanks to the waiver offer.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural network
CBIR	Content based image retrieval search engines
RMAC	Regional maximum activation of convolutions
MS-RMAC	Multi-scale regional maximum activation of convolutions
PCA	Principal component analysis
SIFT	Scale-invariant feature transform
SURF	Sped-up robust features
LSH	Locality sensitive hashing
VLAD	Vector of locally aggregated descriptors

References

1. Belarbi, M.A.; Mahmoudi, S.; Belalem, G. Pca as dimensionality reduction for large-scale image retrieval systems. *Int. J. Ambient. Comput. Intell.* **2017**, *8*, 45–58. [CrossRef]
2. Smith, J.R.; Chang, S.-F. Visualseek: A fully automated content-based image query system. In Proceedings of the Fourth ACM International Conference on Multimedia, New York, NY, USA, 1 February 1997; ACM: New York, NY, USA, 1997; pp. 87–98.
3. Belarbi, M.A.; Mahmoudi, S.; Belalem, G. Indexing video by the content. In *Information Systems Design and Intelligent Applications: Proceedings of Third International Conference INDIA 2016*; Springer: Berlin/Heidelberg, Germany, 2016; Volume 2, p. 21.
4. Sweeney, C.; Liu, L.; Arietta, S.; Lawrence, J. *Hipi: A Hadoop Image Processing Interface for Image-Based Mapreduce Tasks*; University of Virginia: Charlottesville, VA, USA, 2011.
5. Comer, D. Ubiquitous b-tree. *ACM Comput. Surv. (CSUR)* **1979**, *11*, 121–137.
6. Bentley, J.L. Multidimensional binary search trees in database applications. *IEEE Trans. Softw. Eng.* **1979**, *4*, 333–340. [CrossRef]
7. Hadjieleftheriou, M.; Manolopoulos, Y.; Theodoridis, Y.; Tsotras, V.J. R-trees: A dynamic index structure for spatial searching. In *Encyclopedia of GIS*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 993–1002.
8. Dubey, R.S.; Choubey, R.; Bhattacharjee, J. Multi feature content based image retrieval. *Int. J. Comput. Sci. Eng.* **2010**, *2*, 2145–2149.
9. Hirata, K.; Kato, T. Query by visual example. In Proceedings of the International Conference on Extending Database Technology, Vienna, Austria, 23–27 March 1992; Springer: Berlin/Heidelberg, Germany, 1992; pp. 56–71.
10. Faloutsos, C.; Barber, R.; Flickner, M.; Hafner, J.; Niblack, W.; Petkovic, D.; Equitz, W. Efficient and effective querying by image content. *J. Intell. Inf. Syst.* **1994**, *3*, 231–262. [CrossRef]
11. Lee, D.-H.; Kim, H.-J. A fast content-based indexing and retrieval technique by the shape information in large image database. *J. Syst. Softw.* **2001**, *56*, 165–182. [CrossRef]
12. Flickner, M.; Sawhney, H.; Niblack, W.; Ashley, J.; Huang, Q.; Dom, B.; Gorkani, M.; Hafher, J.; Lee, D.; Petkovie, D.; et al. Query by image and video content: The qbic system. *Computer* **1995**, *28*, 23–32. [CrossRef]

13. Cai, H.; Wang, X.; Wang, Y. Compact and robust fisher descriptors for large-scale image retrieval. In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP'2011), Santander, Spain, 18–21 September 2011; pp. 1–6.
14. Belarbi, M.A.; Mahmoudi, S.; Belalem, G.; Mahmoudi, S.A. Web-based multimedia research and indexation for big data databases. In Proceedings of the 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), Rabat, Morocco, 24–26 October 2017; pp. 1–7.
15. Bingham, E.; Mannila, H. Random projection in dimensionality reduction: Applications to image and text data. In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 26–29 August 2001; ACM: New York, NY, USA, 2001; pp. 245–250.
16. Schwartz, W.R.; Kembhavi, A.; Harwood, D.; Davis, L.S. Human detection using partial least squares analysis. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009; pp. 24–31.
17. Shi, Q.; Petterson, J.; Dror, G.; Langford, J.; Strehl, A.L.; Smola, A.J.; Vishwanathan, S.V.N. Hash kernels. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Shanghai, China, 7–8 November 2009; pp. 496–503.
18. Weinberger, K.; Dasgupta, A.; Langford, J.; Smola, A.; Attenberg, J. Feature hashing for large scale multitask learning. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009; ACM: New York, NY, USA, 2009; pp. 1113–1120.
19. Shi, Q.; Li, H.; Shen, C. Rapid face recognition using hashing. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2753–2760.
20. Cools, A.; Belarbi, M.A.; Belarbi, M.A.; Mahmoudi, S.A. A Comparative Study of Reduction Methods Applied on a Convolutional Neural Network. *Electronics* **2022**, *11*, 1422. [CrossRef]
21. Hou, F.; Liu, B.; Zhuo, L.; Zhuo, Z.; Zhang, J. Remote Sensing Image Retrieval with Deep Features Encoding of Inception V4 and Largevis Dimensionality Reduction. *Sens. Imaging* **2021**, *22*, 20. [CrossRef]
22. Perronnin, F.; Liu, Y.; Sánchez, J.; Poirier, H. Large-scale image retrieval with compressed fisher vectors. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3384–3391.
23. White, D.A.; Jain, R. Similarity indexing with the ss-tree. In Proceedings of the Twelfth International Conference on Data Engineering, New Orleans, LA, USA, 26 February–1 March 1996; pp. 516–523.
24. Bouchech, H.J.; Foufou, S.; Abidi, M. Strengthening surf descriptor with discriminant image filter learning: Application to face recognition. In Proceedings of the 26th International Conference on Microelectronics (ICM), Doha, Qatar, 14–17 December 2014; pp. 136–139.
25. Mikolajczyk, K.; Schmid, C. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [CrossRef] [PubMed]
26. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* 2004, *60*, 91–110. [CrossRef]
27. Valgren, C.; Lilienthal, A.J. Sift, surf and seasons: Long-term outdoor localization using local features. In Proceedings of the 3rd European Conference on Mobile Robots (EMCR), Freiburg, Germany, 19–21 September 2007.
28. Muja, M.; Lowe, D.G. Fast matching of binary features. In Proceedings of the 2012 Ninth Conference on Computer and Robot Vision, Toronto, ON, Canada, 28–30 May 2012; pp. 404–410.
29. Hartigan, J.A.; Wong, M.A. Algorithm AS 136: A k-means clustering algorithm. *J. R. Stat. Soc.* **1979**, *28*, 100–108. [CrossRef]
30. Ke, Y.; Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004 (CVPR 2004), Washington, DC, USA, 27 June–2 July 2004; Volume 2, p. II.