A BFS-Tree of Ranking References for Unsupervised Manifold Learning

Daniel Carlos Guimarães Pedronette, Lucas Pascotti Valem

Department of Statistics, Applied Mathematics and Computing (DEMAC), São Paulo State University (UNESP), Rio Claro, Brazil

Ricardo da S. Torres

Department of ICT and Natural Sciences, Faculty of Information Technology and Electrical Engineering, NTNU Norwegian University of Science and Technology, Ålesund, Norway.

Abstract

Contextual information, defined in terms of the proximity of feature vectors in a feature space, has been successfully used in the the construction of search services. These search systems aim to exploit such information to effectively improve ranking results, by taking into account the manifold distribution of features usually encoded. In this paper, a novel unsupervised manifold learning is proposed through a similarity representation based on ranking references. A breadth-first tree is used to represent similarity information given by ranking references and is exploited to discovery underlying similarity relationships. As a result, a more effective similarity measure is computed, which leads to more relevant objects in the returned ranked lists of search sessions. Several experiments conducted on eight public datasets, commonly used for image retrieval benchmarking, demonstrated that the proposed method achieves very high effectiveness results, which are comparable or superior to the ones produced by state-of-the-art approaches.

Keywords: content-based image retrieval, unsupervised manifold learning, tree representation, ranking references

1. Introduction

How to effectively retrieve image objects from large collections? This basic research question has been extensively addressed by researchers in the past 25 years. Typical solutions range from traditional textual description indexing schemes based on database technologies to approaches that rely on sophisticated semantic-aware image content description strategies.

The use of content-based image retrieval (CBIR) systems relies on two basic steps: the image content description itself and the ranking of collection images according to their similarity to a query object. The image content description refers to a set of approaches, which is concerned with the representation of an image as a point in an n-dimensional space [1]. The ranking step, in turn, relies on assessing how *close* representations (*features*) of collection objects are from the query point in the feature space. The closer the features are, the more similar two images are assumed to be.

Which image representation is more appropriate for supporting image retrieval by content? Traditional approaches often took advantage of content visual properties based on global shape, color, and texture; or local and mid-level representations. State-of-the-art image content description has focused on data-driven learning approaches, usually based on neural networks [2, 3]. Regardless the content description used [4], however, effective ranking also depends on the utilization of effective distance/similarity functions. Direct pairwise comparisons of points in the feature space, often based on Euclidean-like distance functions, are common practices in several ranking approaches in the literature. Despite being popular, those strategies fail to produce effective results, when data points are spread over the feature space in very complex arrangements, usually referred as manifold.

In this scenario, some methods aim at ranking collection objects with

respect to the intrinsic manifold structure [5]. Such class of learning/ranking algorithms are broadly based on the assumption that data is sampled from a low dimensional manifold embedded in a higher dimensional Euclidean space [6]. Usually, the motivation is to avoid handling the large dimensionality of a feature space in contrast with the intrinsic dimensionality of the data manifold, which is small [7]. In addition, many datasets have underlying cluster or manifold structure, in which nearby data points, or points belonging to the same cluster or manifold, are very likely to share the same semantic label [8]. Therefore, computing more globally measures in order to taking into account such information consists in an effective way to improve the effectiveness of image retrieval/ranking tasks.

In fact, a family of post-processing methods has been proposed to encode the underlying structure of the dataset manifold [1, 7, 9–11]. A typical advantage of such approaches consists in their capacity of redefining the distance/similarity in a completely unsupervised manner. Unsupervised methods are commonly based on diffusion process [1, 7, 9, 10] and graph learning [12] approaches. Mainly due to the high computational costs associated with diffusion-based approaches, rank-based methods [11, 13–17] have attracted a lot of attention recently.

In this paper, a novel unsupervised manifold learning is proposed centered on a rank-based similarity representation. A Breadth-First Search (BFS) Tree is used to represent the Ranking References among images. The tree representation provides a hierarchical representation of top-k ranking in such a way that edge weights are computed based on rank correlations. Two levels of ranking references are encoded on the tree and each image is represented according to its path to the root. Subsequently, the structure of the tree is exploited to discovery underlying similarity relationships. In this way,

similarity connections are established among nodes and leaves, which are not initially connected by ranking references. The discovered relationships allow to recompute the similarity measure among images. As a result, based on a more effective similarity measure, the effectiveness of retrieval tasks are improved.

Although various unsupervised post-processing methods have been proposed recently for retrieval scenarios [7, 10, 16–20], our approach presents remarkable novelties. The primary contribution in face of related methods consists in the representation of similarity information in terms of a BFS-Tree of Ranking References, as well as the use of this data structure for discovering similarity relationships. While the analysis of ranked lists [16, 17] and graph-based approaches [18–20] has already been exploited, to the best of our knowledge this is the first initiative relying on trees of ranking references in order to derive an unsupervised manifold learning algorithm. Besides that, the proposed method uses recently proposed rank-based procedures as pre and post-processing steps in order to achieve more effective retrieval results.

An extensive experimental evaluation was conducted to assess the effectiveness of the proposed method on image retrieval tasks. Experiments were conducted on eight public datasets of different sizes and considering different retrieval scenarios. The experiments also considered diverse visual features, including global, data-driven ones, and hash-based approaches. Achieved results demonstrate consistently that the proposed approach can achieve effectiveness gains in diverse situations, reaching gains up to +62%. The method is also evaluated in comparison with state-of-the-art post-processing and retrieval approaches, achieving better or comparable results.

The paper is organized as follows: Section 2 formally defines the problem and the rank model while Section 3 presents the proposed manifold learn-

ing algorithm. Section 4 presents the experimental evaluation conducted. Finally, Section 5 outlines our conclusions and directions for future work.

2. Rank and Retrieval Model

This section introduces the image retrieval and ranking model used in this work. A ranking-centered formulation presented in other works [18, 20, 21] is used. Let $\mathcal{C}$=$\{img_1, img_2, \ldots, img_n\}$ be an image collection, where n denotes the size of the collection. Let $\mathcal{D}$ be an image descriptor defined as a tuple $\mathcal{D} = (\epsilon, \rho)$, where: ϵ: $\hat{I} \rightarrow \mathbb{R}^n$ is a function, which extracts a feature vector $v_{\hat{I}}$ from an image $\hat{I}$; and ρ: $\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^+$ is a distance function that computes the distance between two images according to the distance between their corresponding feature vectors.

Therefore, a distance between two images img_i, img_j can be computed by $\rho(\epsilon(img_i), \epsilon(img_j))$. For readability purposes, the notation $\rho(i, j)$ is used along the paper for denoting the distance between images img_i and img_j. Notice that, although the model is defined in terms of image objects, it can be easily extended to handle different data types (e.g., videos, time series).

Grounded on the distance function ρ, a ranking and retrieval model can be derived. A general image retrieval tasks can be modeled as the computation of a ranked list τ_q in response to a query image img_q, according to the distance function ρ. The top positions of ranked lists are expected to contain the most relevant images with regard to the query image, such that only the top-L ranked images are considered, with $L \ll n$.

The ranked list τ_q can be formally defined as a permutation (img_1, img_2, $\ldots$, img_L) of the subset $\mathcal{C}_L \subset \mathcal{C}$, which contains the L most similar images to query image img_q, such that and $|\mathcal{C}_L| = L$. A permutation τ_q is a bijection from the set $\mathcal{C}_L$ onto the set $[n_L] = \{1, 2, \ldots, L\}$. For a permutation τ_q, we

interpret $\tau_q(i)$ as the position (or rank) of image img_i in the ranked list τ_q. If img_i is ranked before img_j in the ranked list of img_q ($\tau_q(i) < \tau_q(j)$), then $\rho(q, i) \leq \rho(q, j)$.

Taking each collection image as a query image, a set of ranked lists $\mathcal{T} = \{\tau_1, \tau_2, \ldots, \tau_n\}$ can be obtained. The set $\mathcal{T}$ encodes a rich source of similarity/dissimilarity information about the collection $\mathcal{C}$, once each ranked list establishes a similarity relationship among the query image and all collection images. The purpose of the proposed manifold learning algorithm is to exploit such information in order to improve the effectiveness of image retrieval tasks.

3. BFS-Tree of Ranking References for Manifold Learning

While traditional content-based retrieval approaches are based on pairwise distance or similarity measures, manifold learning strategies discovery and encode the underlying dataset geometry in order to compute more global measures. For this purpose, a central challenge consists in obtaining effective and efficient representations capable of encoding relevant inter-object similarity information present in a dataset.

In this paper, we propose to represent the similarity information by exploiting a Breadth-First Search (BFS) Tree of Ranking References. Considering a graph defined by ranking references, the proposed tree is constructed as a result to a breadth-first search. Given a query image taken as the root of the tree, the first level is defined by ranking references to the top-k similar images to the query. Subsequently, the next levels are defined by the ranked references of each of top-k images.

The Breadth-First Search is used due to its capacity of representing the similarity information between the query image (q-root), its k-neighborhood and its respective neigbhors in a graph structure. Actually, the co-occurrence

of images at different levels of the tree structure represents a strong indication of similarity, while a single (or low) occurrence indicates noise or wrong retrieval results. There are different graph-based strategies which have been exploited for post-processing methods, including shortest-path propagation [12] and connected components [18]. However, the BFS-trees represent a very effective and efficient structure for the representation task.

Figures 1 and 2 illustrate the construction of the proposed BFS-Tree of Ranking References for a query image img_q. Figure 1 presents the ranked lists retrieved for the query image and for each image from its neighborhood set. Figure 2, in turn, illustrates the relationships given by ranking references (black lines) and the implicit similarity relationships encoded by the tree (dashed red lines). As we can observe, the key role of the BFS-Tree representation consists in allowing the discovery of new similarity relationships.

A novel algorithm is proposed by exploiting the BFS-Tree representation, in order to compute a more general and effective similarity measure. The algorithm is organized through four main steps:

1. **Pre-Processing (Rank Normalization):** the ranked lists are pre-processed through a two-step rank normalization procedure;
2. **BFS-Tree of Ranking References:** a representation of the dataset is constructed based on BFS-trees of ranking references;
3. **BFS-Tree Similarity:** a more global similarity measure is computed by exploiting the information encoded in the BFS-trees;
4. **Post-Processing (Rank Diffusion):** a post-processing step based on a rank diffusion approach is conducted to spread similarity information and improve the final retrieval accuracy.

Each step of the algorithm is detailed and defined in next sections.

Figure 1: The ranked lists for a given query image (within green border) and for each image from its neighborhood set.

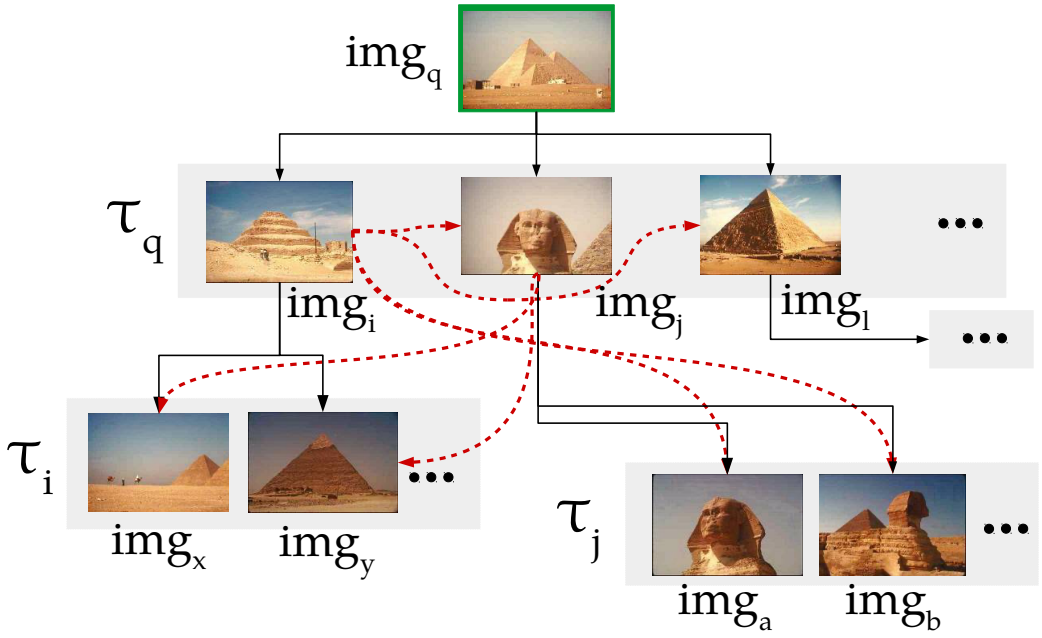

Figure 2: BFS-Tree of Ranking References in an image retrieval session: the continuous black lines represent the similarity information defined by the ranked lists. The dashed red lines represent the new similarity relationships discovered by the proposed approach.

3.1. Rank Normalization

Since each ranked list establishes an independent similarity relationship, the rank references are not symmetric. Therefore, an image img_i well ranked for a query image img_q does not imply that img_q is well ranked in the ranked list of img_i. The objective of the rank normalization is to increase the symmetry of the k-neighborhood, improving the results of retrieval tasks. In fact, various approaches showed the benefits of improving the symmetry of the k-neighborhood for retrieval tasks [18, 22].

The proposed method employs a two-step normalization approach exploiting both mutual [16] and reciprocal [11] neighborhoods. Although both methods analyze the agreement between different ranked lists, the strategy used is different, which can impact the retrieval results. While the mutual neighborhood considers rank positions from both rank references, the reciprocal neighborhood uses only the most pessimistic estimation, given by the maximum position.

The first normalization is combined with the subsequent step by updating the ranked lists through a stable sorting algorithm. In general, the second step can benefit from symmetry improvements already achieved. More specifically, the reciprocal normalization provides a stronger indication of similarity, especially when both ranked lists agree at top positions. On the other hand, it is more susceptible to outliers, since only one lower position defines the score. In this way, the objective is to improve symmetry of ranking similarities firstly by the mutual normalization, which is less susceptible to outliers. Next, we apply the reciprocal normalization in order to obtain a stronger indication of similarity.

Firstly, a mutual rank distance ρ_m based on the mutual neighborhood is defined as:

$$\rho_m(i,j) = \tau_i(j) + \tau_j(i), \tag{1}$$

where $\tau_i(j) \leq L$. Given the new distance ρ_m, the ranked lists are updated by a stable sorting algorithm until the top-L positions. Subsequently, another normalization is performed using a reciprocal neighborhood and deriving a reciprocal rank distance ρ_r, as follows:

$$\rho_r(i,j) = max(\tau_i(j), \tau_j(i)). \tag{2}$$

A re-sorting procedure is conducted through a stable algorithm until the L-th position, giving rise to the final set of ranked list used by the BFS-Tree algorithm.

3.2. BFS-Tree of Ranking References

The proposed tree is defined in terms of the k-neighborhood of collection images. Therefore, first we present a formal definition of the k-neighborhood set of a given image img_q:

$$\mathcal{N}_k(q) = \{\mathcal{R} \subseteq \mathcal{C}, |\mathcal{R}| = k \wedge \forall x \in \mathcal{R}, y \in \mathcal{C} - \mathcal{R} : \rho_r(q,x) \leqslant \rho_r(q,y)\}, \tag{3}$$

where ρ_r denotes the rank distance defined by the normalization step.

A BFS-Tree of Ranking References is constructed based on a query image img_q, which represents the root of the tree (referred to as q-root). The tree is formally defined as a weighted undirected graph $G_q = (V_q, E_q, w)$, where V_q denotes the vertex set, E_q denotes the edge set, and a positive weighting function $w(e)$, that associates weights with edges $e \in E_q$.

The set of vertices V_q is given by union of the q-root, the neighborhood set of img_q and its respective k-neighbors. Due to the possibility of co-occurrence of the same image in distinct neighborhood sets of the tree, a

discriminative notation is used to differentiate the same image in different nodes of the graph. Let img_i and img_j be neighbors of img_q. Let img_z be an image contained in the neighborhood sets of img_i and img_j ($img_z \in \mathcal{N}_k(i) \cap \mathcal{N}_k(j)$). Two nodes are created in the graph to represent such relationships: z_i and z_j, where the subscript notation denotes the antecessor in the tree of ranking references. Only the q-root is represented by q, without the subscript notation. Formally the set of vertices V_q can be defined as:

$$V_q = \{q\} \cup \{i_q \mid img_i \in \mathcal{N}_k(q)\} \cup \{x_i \mid img_x \in \mathcal{N}_k(i) \wedge img_i \in \mathcal{N}_k(q)\} \quad (4)$$

The edge set E_q is defined considering two levels of neighborhood sets. The two levels of neighborhood sets refer to the neighborhood of q-root, and the neighborhood of each q-root's neighbor. Once the k-neighborhood is considered for each neighbor of the q-root's neighbor, a total of k^2 edges are considered. Formally, the edge set E_q is defined as follows:

$$E_q = \{(q, i_q) \mid img_i \in \mathcal{N}_k(q)\} \cup \{(i_q, x_i) \mid img_x \in \mathcal{N}_k(i) \wedge \ img_i \in \mathcal{N}_k(q)\}. \quad (5)$$

Figure 3 illustrates the BFS-Tree of Ranking References. The function $w(\cdot, \cdot)$ defines a weight for the similarity relationship defined by the tree. The continuous black lines represent the edges defined by the BFS-Tree. The dashed lines represent new similarity relationships which can be inferred by analyzing the tree. Section 3.3 discusses how these relationships are exploited for defining a new similarity function among images.

The function $w(\cdot, \cdot)$ is computed by a recently proposed rank correlation measure [23] based on a probabilistic model. The Rank-Biased Overlap (RBO) [23] compares the overlap of two rankings at incrementally increasing depths. This measure takes a parameter that specifies the probability of considering the overlap at the next level. The weight of the overlap measured at

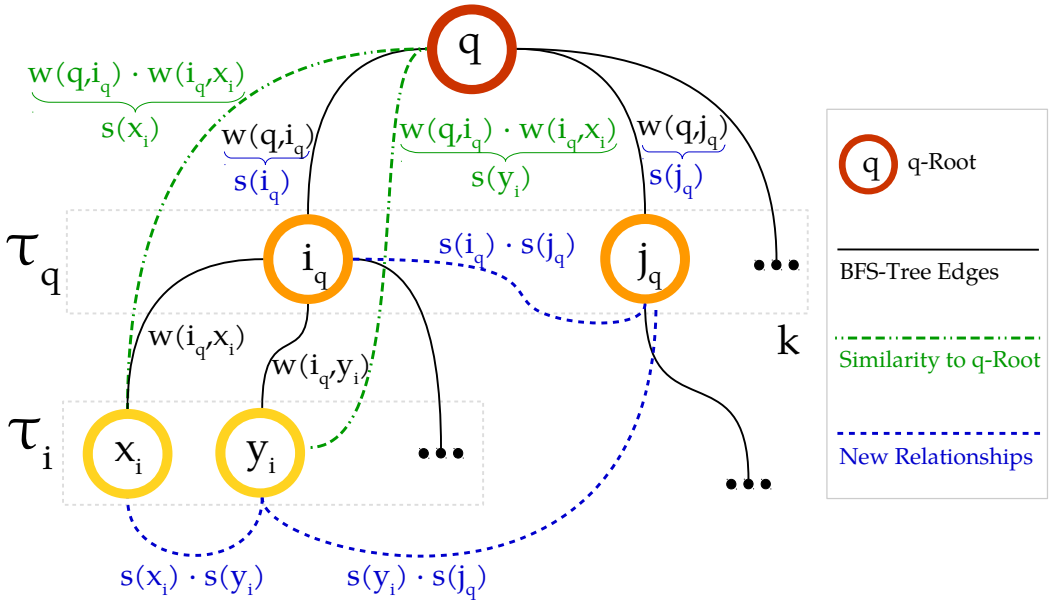

Figure 3: BFS-Tree of Ranking References: tree edges defined by ranked lists and new similarity relationships discovered.

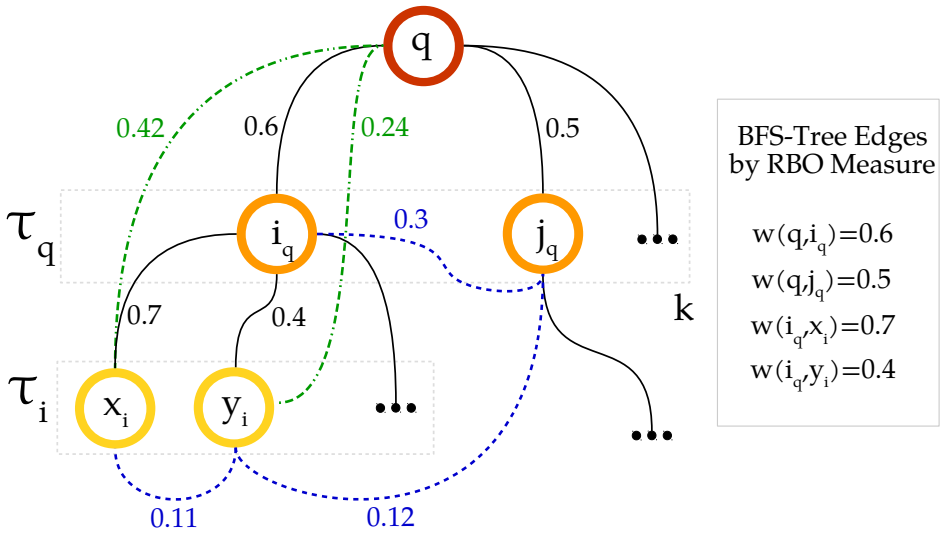

Figure 4: Example of similarity scores defined by the BFS-Tree representation.

each depth is computed based on these probabilities. The rank correlation score is defined in the interval [0,1] as follows:

$$w(q,i) = (1-p)\sum_{d=1}^{k} p^{d-1} \times \frac{|\mathcal{N}_k(q) \cap \mathcal{N}_k(i)|}{d}, \tag{6}$$

where p is a constant, which determines the strength of the weighting to top ranks.

3.3. BFS-Tree Similarity

The objective of the proposed manifold learning algorithm is to exploit the BFS-Tree to define a new and more effective similarity measure. The BFS-Tree is defined based on relationships given by the neighborhood sets. However, many other relationships can be inferred based on the structure of the tree, allowing the extraction of relevant information about the dataset manifold.

Firstly, a similarity function given by $s(\cdot)$ defines the similarity between any node in the BFS-Tree and the q-root. The main idea consists in taking into account the similarity weight of the path which leads to the q-root. For the neighbors of q-root, such similarity is directly defined by the function $w(\cdot,\cdot)$. Figure 3 illustrates the neighbors of q-root on the first level of the tree, given by the ranked list τ_q, which contains images img_i and img_j (in orange).

For other elements, the similarity is computed by the multiplication of weights contained in the path. The reason consists in defining a smaller similarity score to images far from the q-root. The edges named as "*similarity to q-root*" refer to edges between the q-root and its neighbors of neighbors. The "*new relationships*" refer to edges between nodes at same level or different levels of the tree. Since the function $w(\cdot,\cdot)$ is defined in the interval [0,1], the value of $s(\cdot)$ tends to be smaller in such cases. Figure 3 illustrates this

situation by dashed green lines, on the second level of the tree which is given by the ranked list τ_i (images img_x and img_y). An example of instanced tree with computed values is illustrated in Figure 4.

Formally, the function $s(\cdot)$ can be defined as:

$$s(x_y) = \begin{cases} 1, & x = q \\ w(q, x_q), & y = q \wedge img_x \in \mathcal{N}_k(q) \\ w(q, y_q) \times w(y_q, x_y), & x, y \neq q \wedge img_y \in \mathcal{N}_k(q) \wedge img_x \in \mathcal{N}_k(y) \\ 0, & x_y \notin V_q \end{cases} \tag{7}$$

Once the similarity between any tree node and the q-root is defined by $s(\cdot)$, this information is exploited to define a similarity between any pairs of images in the tree. Let img_i and img_j be images in the tree, such that $img_i, img_j \in Vq$. The similarity between img_i and img_j according to the tree defined by q-root is given by $\sigma_q(i, j)$. The function σ_q is computed by the sum of products of similarities of img_i and img_j to q-root considering different predecessors, formally defined as follows:

$$\sigma_q(i, j) = \sum_{x,y \in V_q} s(i_x) \times s(j_y) \tag{8}$$

A more global similarity score between img_i and img_j is computed by taking into account the similarity score defined by all dataset images taken as q-roots. Considering all trees in which images img_i and img_j occur, a function $\sigma_a(i, j)$ is computed as follows:

$$\sigma_a(i, j) = \sum_{img_q \in \mathcal{C}} \sigma_q(i, j) \tag{9}$$

Formally, all the images in the collection are taken as queries ($img_q \in \mathcal{C}$) for computing Equation 9. However, the value of $\sigma_q(i, j)$ is greater than zero only for trees in which img_i and img_j appear. In this way, as new queries

come, there are two options: (i) compute the algorithm considering the whole dataset or; (ii) compute incrementally by exploiting data structures to identify only the trees that contain img_i and img_j.

3.4. Rank Diffusion

Diffusion processes have been established as a traditional and relevant post-processing tool for improving the image retrieval effectiveness [10]. Such approaches operate on affinity graphs for capturing the intrinsic manifold structure of datasets. In general, diffusion processes consider as input a pairwise affinity matrix W, which encodes the similarity relationships among images. Based on the edge weights defined by the matrix W, the diffusion processes spread the affinities through the graph. A more global similarity measure is defined between pairs of vertices in terms of their connectivity. As a result, more effective retrieval results can be obtained.

More recently, several rank-based approaches have been proposed for post-processing tasks, with objectives similar to diffusion process. While such methods achieve comparable effectiveness results, they present very positive efficient aspects, since the processing costs can be constrained to top-ranking positions. In [21], a rank-diffusion approach was proposed, where an iteration of the diffusion process is approximated through the product of similarities of common top-rank positions.

In this work, our motivation is to use the BFS-tree similarity model jointly with a single rank diffusion iteration in order to further improve the effectiveness. The a single iteration is performed by post-processing the computed similarity function σ_a. The new similarity function $\sigma_r(i, j)$ takes into account the similarity to the common top-L positions of img_i and img_j. The function σ_r is formally defined as:

$$\sigma_r(i,j) = \sum_{x \in \mathcal{N}_L(i) \cap \mathcal{N}_L(j)} \sigma_a(i,x) \times \sigma_a(j,x) \tag{10}$$

Finally, the function σ_r is used to compute the final post-processed ranked lists through a stable sorting step.

3.5. Rank Fusion

Despite the continuous advances in image features, the visual content is diversified and often requires different features to encode all available information. In fact, diverse visual retrieval has been proposed in order to represent diverse and complementary aspects about images [4]. Therefore, a natural and relevant research venue consists in combining the different features to reach more effective retrieval results.

In this scenario, a rank fusion approach is proposed by exploiting the capacity of the BFS-Tree representation in capturing the dataset manifold. Based on the captured similarity information, a multiplicative approach inspired by [16] is used. Let the notation $\sigma_{r,c}$ denotes the function σ_r computed for the current feature c and let d denotes the number of features being aggregated, the fused similarity function σ_f is defined as:

$$\sigma_f(i,j) = \prod_{c=1}^{d} (1 + \sigma_{r,c}(i,j)). \tag{11}$$

Based on the fused similarity score σ_f, a single set of ranked lists can be computed based on a sorting procedure.

4. Experimental Evaluation

A comprehensive experimental evaluation was conducted with the aim of assessing the effectiveness of the proposed method. Diverse retrieval scenarios

were considered, including various public datasets and several image descriptors, based on global, local, and deep-learning based features. A rigorous experimental protocol was adopted, involving analysis of parameters, statistical and visual analysis and comparison with state-of-the-art approaches.

This section is organized as follows: Section 4.1 depicts the experimental protocol, describing the datasets, features, and measures adopted. The impact of parameters is analyzed in Section 4.2. Section 4.3 describes the results obtained on shape, color, and texture retrieval tasks. Section 4.4 discusses the experimental results on natural image retrieval tasks, considering different datasets. Section 4.5 evaluates the proposed method on larger datasets and in conjunction with hash-based approaches. A visual and efficiency analysis is presented in Section 4.6. Finally, Section 4.7 presents a comparison with related methods conducted on three datatasets.

4.1. Experimental Protocol

The adopted experimental protocol was defined with the purpose of evaluating the proposed method on different conditions. Thus, six public datasets with diverse content, size, and characteristics were considered. Table 1 presents a brief description of selected datasets. All datasets used all images as queries, except for Holidays [24], which defines a specific protocol with 500 queries.

An analogous criterion was employed regarding the selection of image descriptors. Diverse image descriptors of different types were considered, including local, global, and convolutional neural network-based features. Hashing techiniques and indexing structures were also considered for larger datasets, as described in Section 4.5. The diversity of conditions aims at evaluating the capacity of the proposed manifold learning algorithm in obtaining effectiveness enhancements on different scenarios. The descriptors

Table 1: Description of datasets considered in the experimental evaluation.

Dataset	**Size**	**Type**	**General Description**	**Effectiv. Measure**
MPEG-7 [25]	1,400	Shape	A popular dataset containing 1400 shapes divided into 70 classes. Commonly used for evaluation of post-processing methods.	MAP, Recall@40
Soccer [26]	280	Color Scenes	Images from 7 soccer teams, containing 40 images each team.	MAP
Brodatz [27]	1,776	Texture	A well-known dataset for texture retrieval evaluation composed of 111 textures divided into 16 blocks.	MAP
Holidays [24]	1,491	Scenes	Commonly used as image retrieval benchmark, the dataset is composed of 1,491 personal holiday pictures with 500 queries.	MAP
Corel5K [28]	5,000	Objects/ Scenes	Composed of 50 categories with 100 images each class, including diverse scene content such as fireworks, bark, microscopy images, tiles, trees, etc.	MAP
UKBench [29]	10,200	Objects/ Scenes	Popular benchmark, composed of 2,550 objects or scenes. Each object/scene is captured 4 times from different viewpoints, distances, and illumination conditions.	N-S Score
CIFAR-10 [30]	60,000	Animals/ Objects	Dataset composed of images of animals and transportation means organized into 10 classes, with 6,000 images per class.	MAP
ALOI [31]	72,000	Objects	Images from 1,000 classes of objects, with different viewpoint and illumination.	MAP

used for each dataset are presented in Table 2.

Regarding effectiveness measures, the Mean Average Precision (MAP) is considered for most of experiments. In addition, the N-S score [29] is used for the UKBench [29] dataset, while the Recall at 40 (bull's eye score) for the MPEG-7 [25] dataset.

In order to evaluate the impact of algorithm, the relative effectiveness gains are also reported. The relative gain is defined as $G = (M_a - M_b)/M_b$,

Table 2: Image descriptors considered for each dataset.

Dataset	Image Features	Type
Soccer [26]	Auto Color Correlograms (ACC) [32], Border/Interior Pixel Classification (BIC) [33], Global Color Histogram (GCH) [34]	Color
MPEG-7 [25]	Articulation-Invariant Representation (AIR) [35], Aspect Shape Context (ASC) [36], Beam Angle Statistics (BAS) [37], Contour Features Descriptor (CFD) [38], Inner Distance Shape Context (IDSC) [39], Segment Saliences (SS) [40]	Shape
Brodatz [27]	Color Co-Occurrence Matrix (CCOM) [41], Local Activity Spectrum (LAS) [42], Local Binary Patterns (LBP) [43],	Texture
Corel5K [28]	ACC [32], ACC Spatial Pyramid (ACC-Spy) [32, 44], Color and Edge Directivity Descriptor Spatial Pyramid (CEED-Spy) [44, 45], Convolutional Neural Network by Caffe [2] (CNN-Caffe), FCTH Spatial Pyramid (FCTH-Spy) [44, 46], Joint Composite Descriptor Spatial Pyramid (JCD-Spy) [44, 47], and Local Binary Patterns Spatial Pyramid (LBP-Spy) [43, 44]	Color, Texture, CNN
Holidays [24]	Joint Composite Descriptor (JCD) [47], ACC [32], Color and Edge Directivity Descriptor Spatial Pyramid (CEED-Spy) [44, 45], Convolutional Neural Network by Caffe [2] (CNN-Caffe), Convolutional Neural Network by OverFeat [3] (CNN-OverFeat), Scalable Color Descriptor (SCD) [48]	Color, Texture, BoVW, CNN
UKBench [29]	CEED-Spy [44, 45], ACC [32], ACC Spatial Pyramid (ACC-SPy) [32, 44], CNN-Caffe [2], Fuzzy Color and Texture Histogram Spatial Pyramid (FCTH-SPy) [44, 46], SCD [48], Vocabulary Tree (VOC) [49]	Color, Texture, BoVW, CNN
CIFAR-10 [30]	CNN-NASNET [50], CNN-DPN92 [51], CNN-RESNET [52], Deep Cauchy Hashing (DCH) [53] , Deep Triplet Quantization (DTQ) [54]	CNN, Hashing
ALOI [31]	CNN-NASNET [50], CNN-VGG16 [55], CNN-DPN92 [51], CNN-RESNET [52], Deep Cauchy Hashing (DCH) [53] , Deep Triplet Quantization (DTQ) [54]	CNN, Hashing

where M_b and M_a denote respectively the effectiveness before and after the execution of the manifold learning algorithm.

4.2. Impact of Parameters

This section aims at assessing the robustness of the proposed method with regard to different parameters settings. Three parameters are evaluated: k, which defines the neighborhood size; p which is a parameter related to the RBO measure and; and L, which defines the size of ranked lists.

In fact, the BFS-Tree algorithm requires only one parameter k, since the size of ranked lists L defines a trade-off between effectiveness and efficiency. Therefore, the neighborhood size is very relevant, once the BFS tree coverage is defined by the parameter k. Figure 5 analyses the impact of k on effectiveness for the MPEG-7 dataset considering four shape descriptors (AIR, ASC, CFD, and IDSC). The effectiveness was evaluated through the MAP measure. As we can observe, the produced curve is very stable. Most of the susceptibility to variations occur on the beginning of the curve, for small values of k. For greater values, only very small fluctuations can be noticed, demonstrating a low sensibility of the method to various parameters settings.

Figure 6 presents an analogous analysis for parameter L. The obtained results are very similar, which most of effectiveness variations occurring for small values of L. Another experiment also analyzed the impact of parameter p, evaluated along with the neighborhood size k in Figure 7. The parameters were varied in the intervals [0.55,1] and [5,40], respectively, for p and k. For each pair, the MAP obtained is reported. A large red surface can be observed, demonstrating the robustness of the proposed method. We used p=0.7 and k=20 for most of experiments, except for UKBench [29] and Holidays [24], which used k=5 and p=0.9.

The UKBench and Holidays datasets used different parameters settings in comparison with other datasets due to the very small number of images per class. The UKBench dataset, for instance, is composed of categories

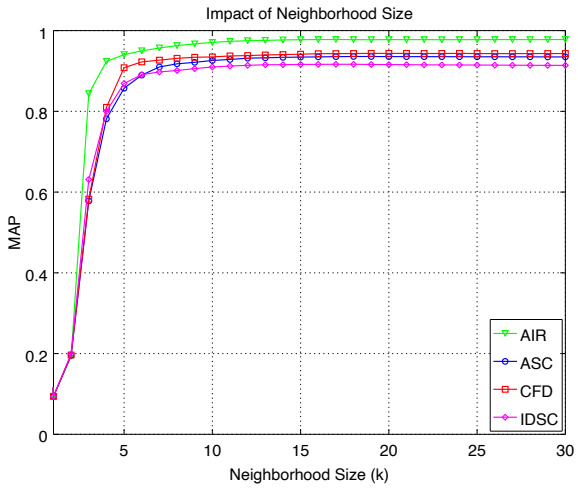

Figure 5: Impact of neighborhood size.

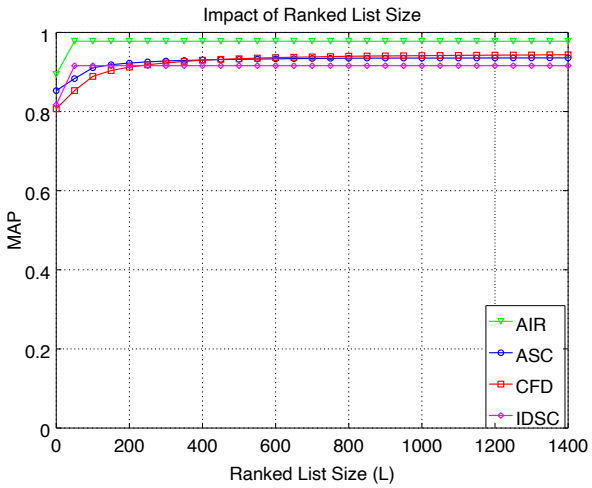

Figure 6: Impact of size of ranked lists.

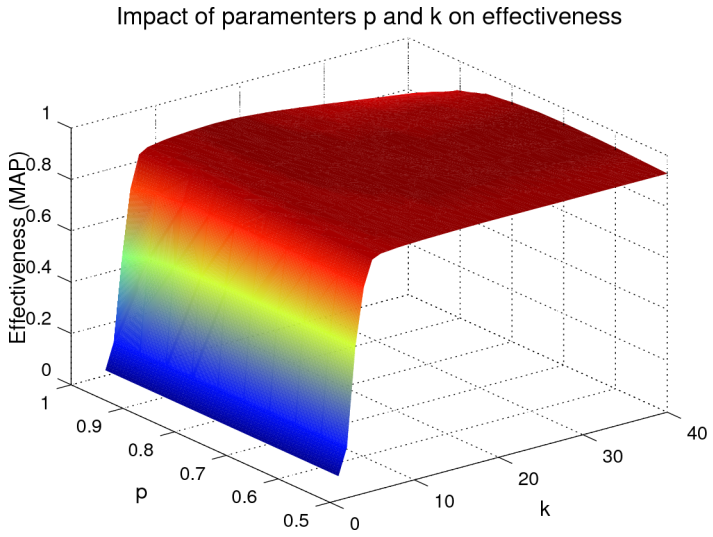

Figure 7: Analysis of conjoint impact of parameters k and p.

with only 4 images per category. Actually, datasets with small number of images per class impose very challenging scenarios for unsupervised manifold learning algorithms. In addition, the experimental protocol is very similar

to other works [18, 56], which use different neighborhood sizes for UKBench and Holidays datasets. The parameter p regulates the weight of top-positions on the RBO correlation measure and is directly associated with the size of neighborhood k. For small number of images per class, the weight varies more smoothly, with higher values (p=0.9).

4.3. General Image Retrieval: Shape, Color, and Texture

The first experiment considers general image retrieval tasks, involving global descriptors based on shape, color, and texture. The proposed manifold algorithm was evaluated on three datasets and twelve descriptors. Statistical paired t-tests were conducted for assessing the impact of the BFS-Tree algorithm, verifying the statistical significance of the difference between effectiveness results before and after the algorithm execution.

Table 3 presents the results, considering a fixed neighborhood size (k=20) and the best parameter setting for each descriptor. As we can observe, the results obtained by fixed k and the best parameter setting are very similar. While the effectiveness gains varied from +9.05% to +44.65%, the difference between the best and fixed k reached at most 1.49% and 0.47% on average. Such small difference demonstrates the robustness of the method to parameter settings. In general, the results indicate that using a fixed value of k (as 20) for different features and datasets is sufficient for obtaining relevant effectiveness gains.

The results are briefly analyzed in the following:

• *Shape Retrieval:* All shape descriptors yielded very significant gains on MPEG-7 dataset. A remarkable results was obtained for the SS [40] descriptor, which reached a gain of +44.65%, being improved from 37.67% to 54.49%.

Table 3: BFS-Tree RR for general image retrieval tasks (shape, color, and texture features), considering the Mean Average Precision (MAP).

Descriptor	**Dataset**	**Original Score (MAP)**	**BFS-Tree RR** k=20	**BFS-Tree RR** Best k	**Best Param Value**	**Relative Gain**	**Stat. Sig. 99%**
Shape Descriptors							
SS [40]	MPEG-7	37.67%	53.87%	54.49%	35	+44.65%	•
BAS [37]	MPEG-7	71.52%	84.86%	84.96%	18	+18.79%	•
IDSC [39]	MPEG-7	81.70%	91.60%	91.68%	18	+12.22%	•
CFD [38]	MPEG-7	80.71%	94.35%	94.37%	21	+16.92%	•
ASC [36]	MPEG-7	85.28%	93.56%	93.60%	18	+9.76%	•
AIR [35]	MPEG-7	89.39%	97.77%	97.84%	40	+14.73%	•
Color Descriptors							
GCH [34]	Soccer	32.24%	35.91%	36.90%	40	+14.45%	•
ACC [32]	Soccer	37.23%	46.71%	48.20%	40	+29.47%	•
BIC [33]	Soccer	39.26%	48.62%	49.81%	40	+36.87%	•
Texture Descriptors							
LBP [43]	Brodatz	48.40%	52.36%	52.78%	10	+9.05%	•
CCOM [41]	Brodatz	57.57%	67.96%	68.09%	16	+18.27%	•
LAS [42]	Brodatz	75.15%	81.75%	82.19%	13	+9.37%	•

• *Color Retrieval:* Similarly to shape retrieval results, all color descriptors achieved substantive gains, ranging from +14.45% to +36.87%.

• *Texture Retrieval:* The texture descriptors yielded effectiveness gains up to +18.27%. All texture descriptors also reported positive gains.

Notice that the effectiveness gains obtained for all descriptors (shape, color, and texture) are statistical significant. An experiment was also conducted to evaluate the BFS-Tree algorithm for rank fusion tasks. The results are presented in Table 4.

The descriptors with the best MAP scores for each dataset was selected, considering a fixed neighborhood size and the best k value. Observe that very high effective results are achieved and all combinations outperformed the respect best descriptor in isolation.

Table 4: BFS-Tree RR for rank aggregation tasks considering shape, color, and texture.

Descriptor	**Type**	**Dataset**	**Neighb. Size (k)**	**Score (MAP)**
CFD [38]	Shape	MPEG-7	-	80.71%
ASC [36]	Shape	MPEG-7	-	85.28%
AIR [35]	Shape	MPEG-7	-	89.39%
CFD+ASC	Shape	MPEG-7	20	98.22%
CFD+ASC	Shape	MPEG-7	40	98.43%
CFD+AIR	Shape	MPEG-7	20	99.55%
CFD+AIR	Shape	MPEG-7	35	99.70%
AIR+ASC	Shape	MPEG-7	20	99.17%
AIR+ASC	Shape	MPEG-7	40	99.34%
CFD+ASC+AIR	Shape	MPEG-7	20	99.87%
CFD+ASC+AIR	Shape	MPEG-7	40	99.94%
ACC [32]	Color	Soccer	-	37.23%
BIC [33]	Color	Soccer	-	39.26%
BIC+ACC	Color	Soccer	20	48.89%
BIC+ACC	Color	Soccer	40	50.13%
CCOM [41]	Texture	Brodatz	-	57.57%
LAS [42]	Texture	Brodatz	-	75.15%
LAS+CCOM	Texture	Brodatz	20	83.94%
LAS+CCOM	Texture	Brodatz	40	84.20%

4.4. Natural Image Retrieval

The BFS-Tree algorithm was evaluated on natural image retrieval tasks considering three well-known datasets: the University of Kentucky Recognition Benchmark - UKBench [29], the Holidays [24] and Corel5K [28] datasets. The MAP score is used as effectiveness measure for Holidays and Corel5K. For the UKBench, the N-S score is used, which is computed between 1 and 4, according to the number of relevant images among the first four images retrieved.

Tables 5, 6, and 7 present the results obtained for UKBench, Holidays, and Corel5K datasets, respectively. In spite of the use of a very diversified set of descriptors, effectiveness gains are obtained in all situations. Even for Holidays and UKBench datasets, which are very challenging for unsupervised manifold learning algorithms due to the small number of images per class, the results are always positive. For Corel5K dataset, high effective results are obtained, reaching effectiveness gains up to +62.20% for CNN-Caffe.

4.5. Larger Datasets and Hashing Techniques

This section discusses the evaluation of the BFS-Tree algorithm on larger datasets, considering CIFAR-10 [30] and ALOI [31] datasets, composed, respectively, of 60K and 72K images. For larger datasets, how to obtain the ranked lists used as input for the BFS-Tree algorithm is also of paramount importance, once computing the Euclidean distances through brute force can be unfeasible. Therefore, our experiments considered different hashing techniques and indexing structures on unsupervised scenarios to obtain the ranked lists.

The feature extraction considered different CNN features followed by a dimensionality reduction procedure using PCA[1]. The redimensioned feature vectors are used as input by indexing and hashing approaches: Ball-Tree [59] and Local Sensitive Hashing (LSH) [60]. Additionally, recent deep-based methods for obtaining the hashing codes were also used: Deep Cauchy Hashing (DCH) [53] and Deep Triplet Quantization (DTQ) [54]. Both hashing approaches and CNN features are trained only on ImageNet [61] in order to keep the unsupervised setting. All approaches were used to obtain top-1000 ranking results and the non-deterministic methods considered 3 executions,

[1] All experiments considered vectors of 100 dimensions.

Table 5: BFS Tree RR on UKBench [29] dataset.

Descriptor	**N-S Score**	**BFS-Tree RR**	**Relative Gain**
JCD [47]	2.79	2.97	+6.45%
CEED-SPy [44, 45]	2.81	3.05	+8.54%
FCTH-SPy [44, 46]	2.91	3.15	+8.24%
SCD [48]	3.15	3.34	+6.03%
CNN-Caffe [2]	3.31	3.57	+7.85%
ACC [32]	3.36	3.57	+6.25%
VOC [49]	3.54	3.70	+4.52%
CNN-VGG [57]	3.65	3.81	+4.38%
CNN-OLDFP [58]	3.84	3.93	+2.34%
VOC + ACC + CNN-Caffe	-	3.91	
CNN-OLDFP + CNN-VGG + VOC	-	**3.95**	

Table 6: BFS-Tree RR on the Holidays [24] dataset.

Descriptor	**Original MAP**	**BFS-Tree RR**	**Relative Gain**
JCD [47]	52.83%	53.37%	+1.02%
FCTH-SPy [44, 46]	55.42%	59.61%	+7.56%
CEED-SPy [44, 45]	56.09%	59.03%	+5.24%
CNN-Caffe [2]	64.09%	69.96%	+9,15%
ACC [32]	64.29%	68.80%	+7.02%
CNN-OverFeat [3]	82.59%	82.95%	+0.44%
CNN-OLDFP [58]	88.46%	88.77%	+0.35%
ACC + CNN-OverFeat	-	84.37%	-
CNN-OLDFP + CNN-OverFeat	-	89.97%	-
ACC + CNN-OLDFP + CNN-OverFeat	-	**90.02%**	-

reporting the results of confidence intervals at 85% of confidence.[2]

[2]For the BFS-Tree algorotihm, we used L=100 and k=20 and k=50, respectively, for the ALOI and CIFAR-10 datasets.

Table 7: Effectiveness results on Corel5K [28] dataset.

Descriptor	Original MAP	BFS-Tree RR	Relative Gain
LBP-Spy	16.28%	19.40%	+19.16%
FCTH-Spy	27.89%	34.72%	+24.49%
JCD-Spy	29.18%	36.92%	+26.53%
ACC	27.75%	39.33%	+41.73%
ACC-Spy	29.76%	38.86%	+30.58%
CEDD-Spy	30.01%	39.07%	+30.19%
CNN-Caffe	28.07%	45.53%	+62.20%
ACC-Spy+CEED-Spy	-	42.25%	-
CNN-Caffe+ACC-Spy+CEED-Spy	-	53.00%	-

Tables 8 and 9 present the results of the BFS-Tree algorithm on the ALOI [31] and CIFAR-10 [30] datasets. We can observe that positive effectiveness gains were obtained for all approaches on both datasets. The BFS-Tree algorithm reached a MAP of 91.15% for RESNET on ALOI [31] dataset. The higher-effectiveness gains were obtained by the hash-based approach DCH [53] on the ALOI [31] dataset, reaching +29.99%.

4.6. Visual and Efficiency Analysis

The proposed manifold learning algorithm redefines the similarity relationships encoded in the whole dataset, with a direct impact on the similarity space. With the purpose of allowing a clear visualization of such impact, we constructed a 3-D representation of the dataset before and after the execution of the BFS-Tree algorithm. Three arbitrary images, called as reference images, are selected from the MPEG-7 dataset. Other collection images are positioned in the space according to their dissimilarity to each reference images, considering each of the axis

Table 8: Effectiveness results on ALOI [31] dataset.

Feature	Indexing/Hashing	Original MAP	BFS-Tree RR	Relative Gain
NASNET	Ball-Tree [59]	64.71%	80.99%	+25.16%
VGG16	Ball-Tree [59]	75.94%	88.87%	+17.03%
DPN92	Ball-Tree [59]	77.38%	89.75%	+15.98%
RESNET	Ball-Tree [59]	79.66%	91.15%	+14.41%
NASNET	LSH [60]	33.07% ± 1.93	36.15% ± 0.0248	+9.29% ± 1.0948
VGG16	LSH [60]	41.31% ± 1.76	43.69% ± 0.0214	+5.73% ± 0.7579
DPN92	LSH [60]	38.05% ± 0.99	40.01% ± 0.0123	+5.14% ± 0.8139
RESNET	LSH [60]	41.19% ± 1.03	43.12% ± 0.0137	+4.68% ± 0.6805
	DCH [53]	52.85% ± 1.11	68.70% ± 0.0134	+29.99% ± 0.5217
	DTQ [54]	39.94% ± 1.80	50.83% ± 0.0169	+27.30% ± 1.6970

Table 9: Effectiveness results on CIFAR-10 [30] dataset.

Feature	Indexing/Hashing	Original MAP	BFS-Tree RR	Relative Gain
NASNET	Ball-Tree [59]	50.93%	50.96%	+0.06%
DPN92	Ball-Tree [59]	55.56%	55.85%	+0.52%
RESNET	Ball-Tree [59]	54.76%	56.10%	+2.45%
NASNET	LSH [60]	24.98% ± 0.0080	25.07% ± 0.0077	+0.35% ± 0.3170
DPN92	LSH [60]	26.87% ± 0.0107	27.46% ± 0.0096	+2.20% ± 0.4943
RESNET	LSH [60]	23.91% ± 0.0255	25.13% ± 0.0234	+5.15% ± 1.5019
	DCH [53]	7.83% ± 1.03%	8.2% ± 0.0106	+4.65% ± 0.4140
	DTQ [54]	8.64% ± 0.78%	9.4% ± 0.0083	+8.72% ± 0.7103

A representation illustrating the similarity space before and after the algorithm are showed in Figures 8 and 9, respectively. Red circles represent similar images to the reference images and the remaining images in blue are illustrated in blue. The selected reference images are illustrated in Figure 10.

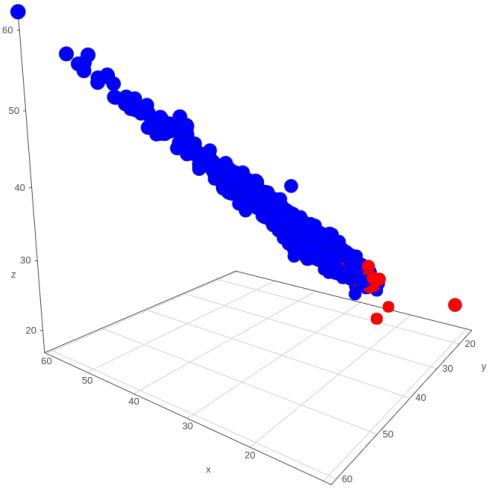

Figure 8: Dissimilarity space before the BFS-Tree RR algorithm.

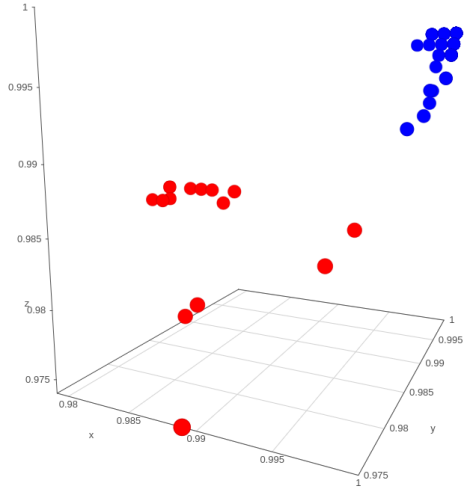

Figure 9: Dissimilarity space after the BFS-Tree algorithm.

Figure 10: Similar reference images considered, respectively for axis x, y, and z. The similar images are illustrated in red and the other remaining images in blue.

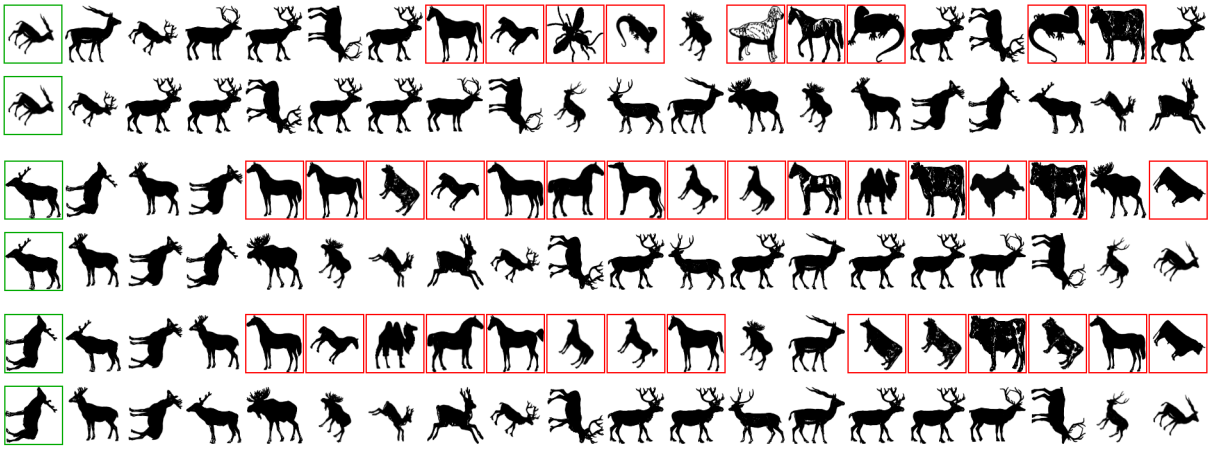

Figure 11: Retrieval results for each of reference images before and after the execution of the BFS-Tree algorithm. The query images are illustrated within green borders and non-similar within red borders.

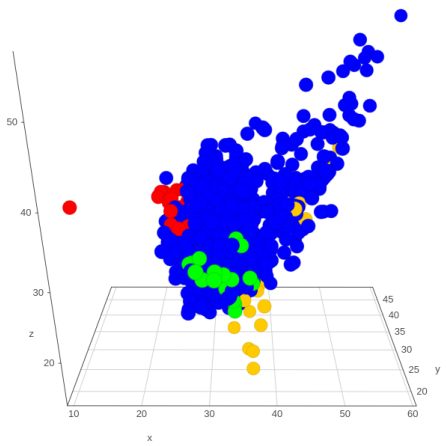

Figure 12: Dissimilarity space before the BFS-Tree RR algorithm.

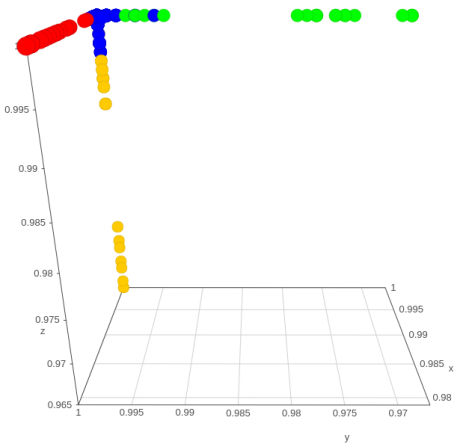

Figure 13: Dissimilarity space after the BFS-Tree algorithm.

Figure 14: Dissimilar reference images considered for axis x, y, and z. The similar images to each reference image are represented respectively in colors red, green, and yellow. The remaining images are illustrated in blue.

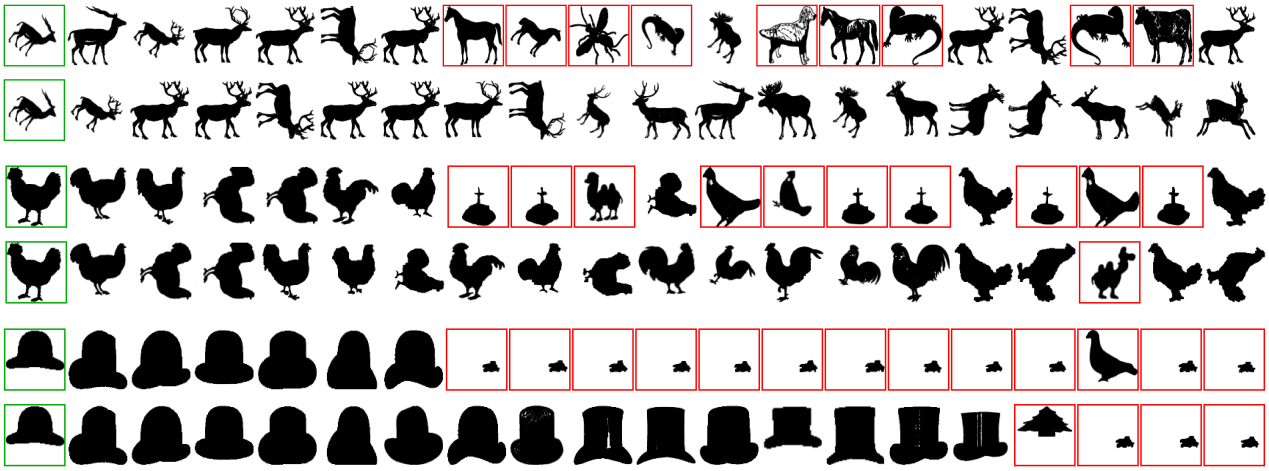

Figure 15: Retrieval results for each of reference images before and after the execution of the BFS-Tree algorithm. The query images are illustrated within green borders and non-similar within red borders.

Observe that similar and non-similar images are mixed in the similarity space in Figure 8, which leads to low-effective retrieval results. After the execution of the algorithm, the increase of the separability between similar and non-similar images is remarkable in Figure 9.

The impact observed in the similarity space can also be noticed in the ranking results. Figure 11 illustrates the retrieval results obtained before and after the algorithm execution for each of reference images. The effectiveness gains are remarkable for all reference images.

An analogous representation is also constructed using dissimilar reference images. Figures 12 and 13 illustrated the 3-D representations before and after the execution of the algorithm. The reference images are showed in Figure 14. The respective retrieval results are illustrated in Figure 15.

Figures 11 and 15 illustrate the top-20 results from the MPEG-7 dataset, which is composed of 70 classes with 20 similar shapes each class. The gains illustrated in the mentioned images are associated with the Precision measure, which considers the number of similar images at top retrieval results. The number of missing similar cases presents a direct impact on the Recall measure. Results for the MPEG-7 dataset using the Recall metric are discussed in Section 4.7. Besides that, most of the evaluation conducted in the paper consider the Mean Average Precision (MAP), which takes into account both Precision and Recall measures jointly.

The impact of the BFS-Tree algorithm can also be visualized in Figure 16. The figure illustrates the retrieval results on Corel5K dataset for different queries (within green borders), before and after the algorithm execution. Non-similar images are illustrated with red borders. As we can observe, the retrieval results after the algorithm are significantly more precise.

Regarding efficiency aspects, the runtime for each dataset and a com-

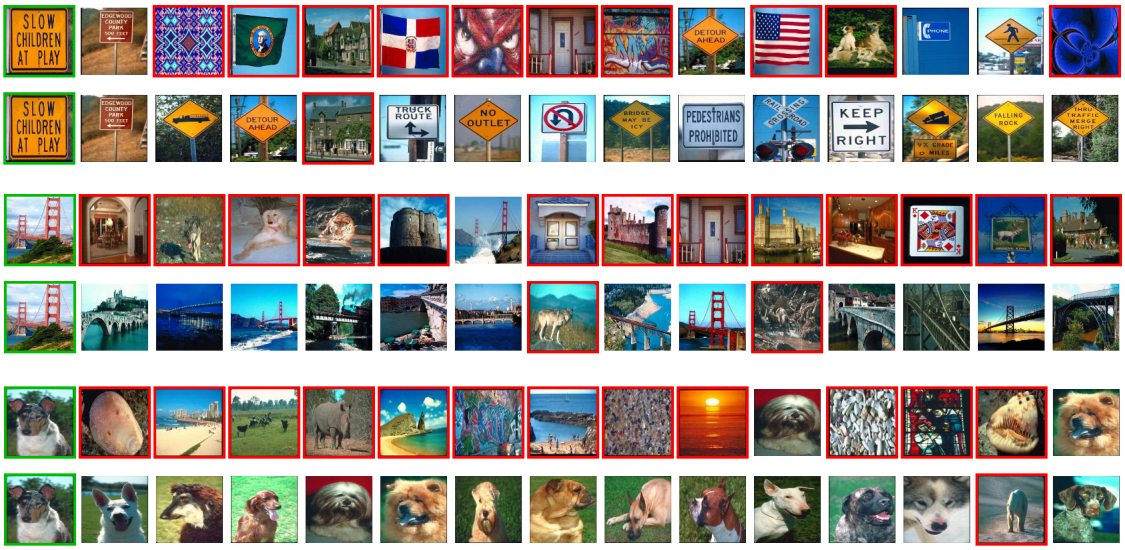

Figure 16: Retrieval results for query images from Corel5K dataset. The first and second line for each query illustrate results before and after the execution of the BFS-Tree algorithm. The query images are illustrated within green borders and non-similar within red borders.

plexity analysis is presented. The experiments were executed on an Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz with Linux 4.4.0 - Ubuntu 14.04. Table 10 presents the runtime for different datasets, considering the various steps of the algorithm. The reported times considered an average of 10 executions. The algorithm is currently implemented in a publicly available framework of unsupervised learning algorithms (UDLF) [62].

An important characteristic consists in the asymptotic complexity of the proposed algorithm, which is very low. All the analysis can be computed considering only the top-L positions of ranked lists for trank normalization and the top-k positions for the BFS-tree analysis. Since the parameters k and L are constant, the asymptotic complexity is $O(1)$ for each query image and $O(n)$ for the whole dataset.

Table 10: Runtime of BFS-Tree on different datasets.

Dataset	Rank Normalization	BFS-Tree Analysis	Rank Diffusion	Sorting	Full Runtime	Time per query
Soccer	0.0071 s	0.1471 s	0.0828 s	0.0051 s	0.2423 s	0.8654 ms
MPEG-7	0.2451 s	0.5697 s	8.2473 s	0.5265 s	9.5887 s	6.8490 ms
Holidays	0.4850 s	0.0244 s	9.5770 s	0.2954 s	10.3820 s	6.9631 ms
Brodatz	0.4331 s	0.6819 s	15.9960 s	0.7704 s	17.8817 s	10.0685 ms
Corel5K	12.3492 s	2.1720 s	452.8500 s	24.7158 s	492.0870 s	98.4174 ms
UKBench	0.2203 s	0.7998 s	3.0103 s	0.0611 s	4.0917 s	0.4011 ms
ALOI	11.8774 s	128.1810 s	1251.9800 s	8.0586 s	1400.1000 s	19.4458 ms

4.7. Comparison with Other Approaches

The BFS-Tree algorithm is also compared with diverse state-of-the-art related methods. The comparison considered recently proposed unsupervised learning methods and retrieval approaches on datasets commonly used as benchmark for image retrieval: MPEG-7 [25], Holidays [24], and UKBench [29].

Table 11 reports the effectiveness results on the MPEG-7 [25]. According to the dataset protocol, the bull's eye score (Recall@40), which counts similar images within the top-40 rank positions, is used as effectiveness measure. Several state-of-the-art post-processing methods are evaluated on four different shape descriptors (IDSC, ASC, CFD, and AIR). As we can observe, the BFS-Tree algorithm achieved high-effective scores for all features, reaching the best result for three of them.

Table 12 presents the the MAP scores obtained by the BFS-Tree algorithm on the Holidays [24] dataset, in comparison with state-of-the-art retrieval methods. Table 13 reports the N-S score obtained on the UKBench [29] dataset in comparison with recent retrieval and unsupervised post-processing approaches. Notice that very high effective results are re-

Table 11: Comparison of bull's eye score on the MPEG-7 [25] dataset.

Shape Descriptors		
CFD [38]	-	84.43%
IDSC [39]	-	85.40%
ASC [36]	-	88.39%
AIR [35]	-	93.67%
Post-Processing Methods		
Algorithm	**Descriptor(s)**	**Bull's eye score**
Graph Transduction [63]	IDSC [39]	91.00%
Self-Smoothing Operator [1]	IDSC [39]	92.77%
Local Constr. Diff. Process [9]	IDSC [39]	93.32%
Shortest Path Propagation [12]	IDSC [39]	93.35%
SCA [56]	IDSC [39]	93.44%
BFS-Tree of Ranking Refs	**IDSC [39]**	**93.68%**
RDP [64]	IDSC [39]	93.78%
Correlation Graph [65]	ASC [36]	95.22%
Local Constr. Diff. Process [9]	ASC [36]	95.96%
Smooth Neighborhood [66]	ASC [36]	95.98%
Reciprocal kNN Graph + CCs [18]	ASC [36]	96.04%
BFS-Tree of Ranking Refs	**ASC [36]**	**96.18%**
Tensor Product Graph [7]	ASC [36]	96.47%
Correlation Graph [65]	CFD [38]	94.27%
RL-Sim [16]	CFD [38]	94.27%
Rank Diffusion [21]	CFD [38]	96.19%
Reciprocal kNN Graph + CCs [18]	CFD [38]	96.61%
BFS-Tree of Ranking Refs	**CFD [38]**	**96.90%**
RL-Sim [16]	AIR [35]	99.94%
Tensor Product Graph [7]	AIR [35]	99.99%
Generic Diffusion Process [10]	AIR [35]	100%
Neighbor Set Similarity [14]	AIR [35]	100%
BFS-Tree of Ranking Refs	**AIR [35]**	**100%**

ported on both datasets, outperforming most of considered approaches.

Table 12: Comparison with state-of-the-art on the Holidays [24] dataset.

MAP scores for recent retrieval methods.			
Jégou *et al.* [24]	Tolias *et al.* [67]	Qin *et al.* [68]	Zheng *et al.* [69]
75.07%	82.20%	84.40%	85.20%

Zheng *et al.* [70]	Pedronette et al. [18]	Iscen et al. [71]	Li *et al.* [72]	**BFS-Tree RR** ACC+CNN-OLDFP +CNN-OverFeat
85.80%	86.19%	87.50%	89.20%	**90.02%**

Table 13: Comparison with state-of-the-art on the UKBench [29] dataset.

N-S scores for recent retrieval methods					
Zheng *et al.* [73]	Qin *et al.* [11]	Zhang *et al.* [19]	Zheng *et al.* [74]	Bai *et al.* [56]	Xie *et al.* [75]
3.57	3.67	3.83	3.84	3.86	3.89

Pedronette et al. [18]	Bai et al. [76]	Bai et al. [64]	**BFS-Tree RR** VOC+ACC+ CNN-Caffe	**BFS-Tree RR** CNN-OLDFP+ +CNN-VGG+VOC
3.93	3.93	3.93	3.94	**3.95**

5. Conclusions

Defining an accurate similarity measure between data is a task of crucial importance in many areas, specially image retrieval. Beside that, structures capable of modelling the dataset manifold have been established as relevant tools for more effective similarity measures. In this paper, a novel unsupervised manifold learning is proposed based on a BFS-Tree structure. The most relevant conclusion is the potential of exploiting representation strategies to discovery underlying similarity relationships. The proposed approach uses a BFS-Tree of Ranking References considering the query image as the root

of the tree. By representing the top-k neighbors and their respective neighbors in the same data structure, the manifold learning algorithm allows for reestablishing the similarity relationships and, therefore, improving retrieval results.

An extensive experimental evaluation was conducted and the proposed approach achieved very significant effectiveness gains on diverse situations. A rigorous experimental protocol and comparisons with various state-of-the-art approaches also demonstrated the effectiveness of the proposed method. Future work focuses on the investigation of other measures for defining the BFS-Tree representation. In addition, another promising research direction consists in developing adaptive strategies for automatically searching an optimal value of k. We intend to investigate the use of scores derived from the BFS-tree structure to identify an adaptive optimal neighborhood size.

Acknowledgments

The authors are grateful to CAPES (grant #88881.145912/2017-01), FAPESP - São Paulo Research Foundation (grants #2018/15597-6, #2017/25908-6, #2014/12236-1, #2015/24494-8, #2016/50250-1, and #2017/20945-0), FAPESP-Microsoft Virtual Institute (grants #2013/50155-0, #2013/50169-1, and #2014/50715-9) and CNPq - National Council for Scientific and Technological Development (grants #308194/2017-9 and #307560/2016-3). This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] J. Jiang, B. Wang, Z. Tu, Unsupervised metric learning by self-smoothing operator, in: International Conference on Computer Vision (ICCV'2011), 2011,

pp. 794–801.

[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, CoRR abs/1408.5093.

[3] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN features off-the-shelf: an astounding baseline for recognition, in: Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW'14), pp. 512–519.

[4] Y. Uchida, Local feature detectors, descriptors, and image representations: A survey, CoRR abs/1607.08368.

[5] D. Zhou, J. Weston, A. Gretton, O. Bousquet, B. Schölkopf, Ranking on data manifolds, in: 16th International Conference on Neural Information Processing Systems, NIPS'03, 2003.

[6] X. Zhou, M. Belkin, N. Srebro, An iterated graph laplacian approach for ranking on manifolds, in: 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, 2011, pp. 877–885.

[7] X. Yang, L. Prasad, L. Latecki, Affinity learning with diffusion on tensor product graph, IEEE Transactions on Pattern Analysis and Machine Intelligence, 35 (1) (2013) 28–38.

[8] B. Xu, J. Bu, C. Chen, D. Cai, X. He, W. Liu, J. Luo, Efficient manifold ranking for image retrieval, in: 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '11, 2011, pp. 525–534.

[9] X. Yang, S. Koknar-Tezel, L. J. Latecki, Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval., in: CVPR'2009, 2009, pp. 357–364.

[10] M. Donoser, H. Bischof, Diffusion processes for retrieval revisited, in: IEEE Conferece on Computer Vision and Pattern Recognition (CVPR'2013), 2013, pp. 1320–1327.

[11] D. Qin, S. Gammeter, L. Bossard, T. Quack, L. van Gool, Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors, in: CVPR'2011, 2011, pp. 777 –784.

[12] J. Wang, Y. Li, X. Bai, Y. Zhang, C. Wang, N. Tang, Learning context-sensitive similarity by shortest path propagation, Pattern Recognition 44 (10-11) (2011) 2367–2374.

[13] D. C. G. Pedronette, J. Almeida, R. da S. Torres, A scalable re-ranking method for content-based image retrieval, Information Sciences 265 (1) (2014) 91–104.

[14] X. Bai, S. Bai, X. Wang, Beyond diffusion process: Neighbor set similarity for fast re-ranking, Information Sciences 325 (2015) 342 – 354.

[15] D. C. G. Pedronette, O. A. Penatti, R. da S. Torres, Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks, Image and Vision Computing 32 (2) (2014) 120 – 130.

[16] D. C. G. Pedronette, R. da S. Torres, Image re-ranking and rank aggregation based on similarity of ranked lists, Pattern Recognition 46 (8) (2013) 2350–2360.

[17] Y. Chen, X. Li, A. Dick, R. Hill, Ranking consistency for image matching and object retrieval, Pattern Recognition 47 (3) (2014) 1349 – 1360.

[18] D. C. G. Pedronette, F. M. F. Gonalves, I. R. Guilherme, Unsupervised manifold learning through reciprocal knn graph and connected components for image retrieval tasks, Pattern Recognition 75 (2018) 161 – 174.

[19] S. Zhang, M. Yang, T. Cour, K. Yu, D. Metaxas, Query specific rank fusion for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (4) (2015) 803–815.

[20] D. C. G. Pedronette, J. Almeida, R. da S. Torres, A graph-based ranked-list model for unsupervised distance learning on shape retrieval, Pattern Recognition Letters 83, Part 3 (2016) 357 – 367.

[21] D. C. G. Pedronette, R. da S. Torres, Unsupervised rank diffusion for content-based image retrieval, Neurocomputing 260 (2017) 478 – 489.

[22] H. Jegou, C. Schmid, H. Harzallah, J. Verbeek, Accurate image search using the contextual dissimilarity measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (1) (2010) 2–11.

[23] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Trans. on Information Systems 28 (4) (2010) 20:1–20:38.

[24] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision, ECCV '08, 2008, pp. 304–317.

[25] L. J. Latecki, R. Lakmper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour.

[26] J. van de Weijer, C. Schmid, Coloring local feature extraction, in: European Conf. on Computer Vision (ECCV'2006), Vol. Part II, 2006, pp. 334–348.

[27] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover, 1966.

[28] Content-based image retrieval using color difference histogram, Pattern Recognition 46 (1) (2013) 188 – 198.

[29] D. Nistér, H. Stewénius, Scalable recognition with a vocabulary tree, in: CVPR'2006, Vol. 2, 2006, pp. 2161–2168.

[30] A. Krizhevsky, Learning multiple layers of features from tiny images, Tech. rep. (2009).
URL https://www.cs.toronto.edu/~kriz/cifar.html

[31] J.-M. Geusebroek, G. J. Burghouts, A. W. M. Smeulders, The amsterdam library of object images, International Journal of Computer Vision 61 (1) (2005) 103–112.

[32] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih, Image indexing using color correlograms, in: CVPR'97, 1997, pp. 762–768.

[33] R. O. Stehling, M. A. Nascimento, A. X. Falcão, A compact and efficient image retrieval approach based on border/interior pixel classification, in: CIKM'2002, 2002, pp. 102–109.

[34] M. J. Swain, D. H. Ballard, Color indexing, International Journal on Computer Vision 7 (1) (1991) 11–32.

[35] R. Gopalan, P. Turaga, R. Chellappa, Articulation-invariant representation of non-planar shapes, in: ECCV'2010, Vol. 3, 2010, pp. 286–299.

[36] H. Ling, X. Yang, L. J. Latecki, Balancing deformability and discriminability for shape matching, in: ECCV'2010, Vol. 3, 2010, pp. 411–424.

[37] N. Arica, F. T. Y. Vural, BAS: a perceptual shape descriptor based on the beam angle statistics, Pattern Recognition Letters 24 (9-10) (2003) 1627–1639.

[38] D. C. G. Pedronette, R. da S. Torres, Shape retrieval using contour features and distance optmization, in: VISAPP'2010, Vol. 1, 2010, pp. 197 – 202.

[39] H. Ling, D. W. Jacobs, Shape classification using the inner-distance, IEEE Trans. on Pattern Analysis and Machine Intelligence 29 (2) (2007) 286–299.

[40] R. da S. Torres, A. X. Falcão, Contour Salience Descriptors for Effective Image Retrieval and Analysis, Image and Vision Computing 25 (1) (2007) 3–13.

[41] V. Kovalev, S. Volmer, Color co-occurence descriptors for querying-by-example, in: Int. Conference on Multimedia Modeling, 1998, p. 32.

[42] B. Tao, B. W. Dickinson, Texture recognition and image retrieval using gradient indexing, Journal of Visual Comunication and Image Representation 11 (3) (2000) 327–342.

[43] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. on Pattern Analysis and Machine Intellig. 24 (7) (2002) 971–987.

[44] M. Lux, Content based image retrieval with LIRe, in: Proceedings of the 19th ACM International Conference on Multimedia, MM '11, 2011.

[45] S. A. Chatzichristofis, Y. S. Boutalis, Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval, in: Inte. Conference on Computer Vision Systems, ICVS'08, 2008, pp. 312–322.

[46] S. A. Chatzichristofis, Y. S. Boutalis, Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval, in: Int. Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), 2008, pp. 191–196.

[47] K. Zagoris, S. Chatzichristofis, N. Papamarkos, Y. Boutalis, Automatic image annotation and retrieval using the joint composite descriptor, in: 14th Panhellenic Conference on Informatics (PCI), 2010, pp. 143–147.

[48] B. Manjunath, J.-R. Ohm, V. Vasudevan, A. Yamada, Color and texture descriptors, IEEE Transactions on Circuits and Systems for Video Technology 11 (6) (2001) 703–715.

[49] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, T. Han, Contextual weighting for vocabulary tree based image retrieval, in: IEEE International Conference on Computer Vision (ICCV'2011), 2011, pp. 209–216.

[50] B. Zoph, V. Vasudevan, J. Shlens, Q. V. Le, Learning transferable architectures for scalable image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), 2018.

[51] Y. Chen, J. Li, H. Xiao, X. Jin, S. Yan, J. Feng, Dual path networks, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems 30, Curran Associates, Inc., 2017, pp. 4467–4475.

[52] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[53] Y. Cao, M. Long, B. Liu, J. Wang, Deep cauchy hashing for hamming space retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 1229–1237.

[54] B. Liu, Y. Cao, M. Long, J. Wang, J. Wang, Deep triplet quantization, in: ACM International Conference on Multimedia, MM 18, 2018, p. 755763.

[55] S. Liu, W. Deng, Very deep convolutional neural network based image classification using small training sample size, in: 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 2015, pp. 730–734.

[56] S. Bai, X. Bai, Sparse contextual activation for efficient visual re-ranking, IEEE Transactions on Image Processing (TIP) 25 (3) (2016) 1056–1069.

[57] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556.

[58] K. R. Mopuri, R. V. Babu, Object level deep feature pooling for compact image representation, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015, pp. 62–70.

[59] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[60] A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in: Proceedings of the 25th International Conference on Very Large Data Bases, VLDB 99, 1999, p. 518529.

[61] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A Large-Scale Hierarchical Image Database, in: CVPR09, 2009.

[62] L. P. Valem, D. C. G. Pedronette, An unsupervised distance learning framework for multimedia retrieval, in: ACM on International Conference on Multimedia Retrieval, ICMR'17, 2017, pp. 107–111.

[63] X. Yang, X. Bai, L. J. Latecki, Z. Tu, Improving shape retrieval by learning graph transduction, in: European Conference on Computer Vision (ECCV'2008), Vol. 4, 2008, pp. 788–801.

[64] S. Bai, X. Bai, Q. Tian, L. J. Latecki, Regularized diffusion process on bidirectional context for object retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (5) (2019) 1213–1226.

[65] D. C. G. Pedronette, R. da S. Torres, Unsupervised manifold learning by correlation graph and strongly connected components for image retrieval, in: Int. Conference on Image Processing (ICIP'2014), 2014.

[66] S. Bai, S. Sun, X. Bai, Z. Zhang, Q. Tian, Smooth neighborhood structure mining on multiple affinity graphs with applications to context-sensitive similarity, in: European Conf. on Computer Vision (ECCV), 2016, pp. 592–608.

[67] G. Tolias, Y. Avrithis, H. Jgou, To aggregate or not to aggregate: Selective match kernels for image search, in: IEEE International Conference on Computer Vision (ICCV'2013), 2013, pp. 1401–1408.

[68] D. Qin, C. Wengert, L. V. Gool, Query adaptive similarity for large scale object retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2013), 2013, pp. 1610–1617.

[69] L. Zheng, S. Wang, Q. Tian, Coupled binary embedding for large-scale image retrieval, IEEE Trans. on Image Processing (TIP) 23 (8) (2014) 3368–3380.

[70] L. Zheng, S. Wang, Z. Liu, Q. Tian, Packing and padding: Coupled multi-index for accurate image retrieval, in: Conf. on Computer Vision and Pattern Recognition (CVPR'2014), 2014, pp. 1947–1954.

[71] A. Iscen, G. Tolias, Y. S. Avrithis, O. Chum, Mining on Manifolds: Metric learning without labels, in: IEEE Int. Conf. Computer Vision and Pattern Recognition (CVPR2018), 2018.

[72] X. Li, M. Larson, A. Hanjalic, Pairwise geometric matching for large-scale object retrieval, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2015), 2015, pp. 5153–5161.

[73] L. Zheng, S. Wang, Q. Tian, Lp-norm idf for scalable image retrieval, IEEE Transactions on Image Processing 23 (8) (2014) 3604–3617.

[74] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, Q. Tian, Query-adaptive late fusion for image search and person re-identification, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[75] L. Xie, R. Hong, B. Zhang, Q. Tian, Image classification and retrieval are one, in: ACM Int. Conf. on Multimedia Retrieval (ICMR), 2015, pp. 3–10.

[76] S. Bai, X. Bai, Q. Tian, L. J. Latecki, Regularized diffusion process for visual retrieval, in: Conf. on Artificial Intelligence (AAAI), 2017, pp. 3967–3973.