Accepted version of paper published in IEEE Transactions on Image Processing (TIP), 2023.
DOI: https://doi.org/10.1109/TIP.2023.3268868

Rank Flow Embedding for Unsupervised and Semi-Supervised Manifold Learning

Lucas Pascotti Valem, Daniel Carlos Guimarães Pedronette
and Longin Jan Latecki.

***Abstract*—Impressive advances in acquisition and sharing technologies have made the growth of multimedia collections and their applications almost unlimited. However, the opposite is true for the availability of labeled data, which is needed for supervised training, since such data is often expensive and time-consuming to obtain. While there is a pressing need for the development of effective retrieval and classification methods, the difficulties faced by supervised approaches highlight the relevance of methods capable of operating with few or no labeled data. In this work, we propose a novel manifold learning algorithm named Rank Flow Embedding (RFE) for unsupervised and semi-supervised scenarios. The proposed method is based on ideas recently exploited by manifold learning approaches, which include hypergraphs, Cartesian products, and connected components. The algorithm computes context-sensitive embeddings, which are refined following a rank-based processing flow, while complementary contextual information is incorporated. The generated embeddings can be exploited for more effective unsupervised retrieval or semi-supervised classification based on Graph Convolutional Networks. Experimental results were conducted on 10 different collections. Various features were considered, including the ones obtained with recent Convolutional Neural Networks (CNN) and Vision Transformer (ViT) models. High effective results demonstrate the effectiveness of the proposed method on different tasks: unsupervised image retrieval, semi-supervised classification, and person Re-ID. The results demonstrate that RFE is competitive or superior to the state-of-the-art in diverse evaluated scenarios.**

***Index Terms*—ranking, embedding, unsupervised, semi-supervised, manifold learning, person Re-ID**

I. Introduction

CONTENT-based Image Retrieval (CBIR) is a central tool behind a diversified range of applications. In fact, it can be seen as technology that helps to organize digital picture archives by their visual content [1], including a broad spectrum of approaches, from general object retrieval to medical diagnostics support and person re-identification [1]–[3]. A traditional task is given by a query-by-example arrangement, which consists in retrieving the most similar images to a query image defined by the user from an image collection [4]. While involving various challenges and the fundamental open problem of robust image understanding [1], it can also be seen as a rank-centered task, once the retrieved images are expected to be ranked according to the user needs.

The ranking tasks performed by CBIR approaches typically rely on two basic steps: the image content representation itself and the similarity measurement of collection images to the query. The image representation is concerned with mapping an image to a point in a high-dimensional feature space. The similarity measurement, in turn, relies on assessing how close representations of collection images are from the query point in the feature space [5]. Conventionally, it is accomplished by computing the pairwise dissimilarity between feature representations in the Euclidean space [6].

Extensive advances have been made in image representation techniques over the last decades. Originally, the extraction of global features defined the dominant approach, where a myriad of features were proposed, mainly based on visual properties such as shape, texture, and color. The global features gave rise to local feature strategies, based on Bag-of-Words (BoW) model, largely studied over a decade [7]. More recently, the success of deep neural networks on feature representation has made them a fundamental tool in image retrieval. Models pretrained on huge datasets are broadly used through transfer learning to extract features of images [2], [8].

Despite the huge advances in representation strategies, especially supported by recent deep features given by Convolutional Neural Networks (CNN) and Vision Transformers (ViT) models, a major limitation is associated with the pairwise formulation of similarity measurements. In fact, both traditional and deep-based representations lie on manifolds in a high-dimensional space [9] such that pairwise similarity measures are insufficient to reveal the intrinsic relationship between images. Instead, similarities can be estimated more accurately along the geodesic paths of the underlying data manifold [6]. The goal of such strategies is to somehow mimic human behavior in judging the similarity among objects; i.e., by considering the context of other objects.

In this research direction, different approaches have been proposed to post-process pairwise measures in order to compute more global and effective similarity measures [5], [10]–[14]. Different techniques and a comprehensive terminology have been employed, all following the common objective of capturing the structural similarity information encoded in the datasets through unsupervised contextual analysis. Such contextual-sensitive similarity measures have been successfully applied to capture the geometry of the underlying manifold in order to improve retrieval tasks.

Diffusion processes demonstrated high potential in capturing the underlying manifold structure [6], [15]. Diffusion processes use a weighted graph, where each image is represented by a node, and edge weights are defined by pairwise affinity values. The pairwise affinities are re-evaluated in the context of other images, by spreading the similarity values across the graph. Affinities are spread on the manifold, which in turn improves the retrieval scores [10]. Several variants have been proposed [10], including methods capable of analyzing high-order similarity relationships [6]. In addition, such approaches are supported by a strong mathematical background but are often associated to high computational costs [16].

The authors are grateful to Fulbright Commission, São Paulo Research Foundation - FAPESP (grants #2018/15597-6 and #2017/25908-6), Brazilian National Council for Scientific and Tech. Development - CNPq (grants #309439/2020-5 and #422667/2021-8), and Petrobras. This work was also partly supported by the National Science Foundation, USA Grant No. IIS-1814745.

L. P. Valem and D. C. G. Pedronette are with the Department of Statistics, Applied Math. and Computing, State University of São Paulo, Rio Claro, Brazil (e-mail: {daniel.pedronette,lucas.valem}@unesp.br).

L. J. Latecki is with Department of Computer and Information Sciences, Temple University, Philadelphia, USA (e-mail: latecki@temple.edu).

Re-ranking and rank-based manifold learning methods constitute another representative category of unsupervised post-processing methods [5], [12], [17]–[19]. In fact, ranked lists provide a rich source of contextual information once they establish a similarity relationship among a set of images, in contrast to pairwise relations. Additionally, the most relevant information in the ranked lists is located at top positions, which enables the development of efficient algorithms [20]. Reciprocal similarity relationships [19], [21], [22] and rank correlation measures [18], [23], [24] have been successfully applied by various approaches.

Graphs and embeddings are modeling tools that also have been demonstrating a high potential for contextual similarity analysis. The shortest path in the graph is used to define the similarity between images in [25]. Connected Components are exploited in [19], [26] for spreading confident similarity relationships. Lately, hypergraphs have been exploited, mainly due to their capacity of representing high-order similarity information [6], [14]. More recently, approaches that learn a mapping function to an embedded space have been proposed that exhibit the capacity of generalizing to new data [27], but such approaches are still rarely considered in the literature.

On the other hand, unsupervised image retrieval and semi-supervised classification are well-known and largely studied tasks. However, they remain challenging interconnected tasks, with many applications in diverse scenarios (person re-identification [28], remote sensing [29], medical imaging [30], and many others). In spite of many advances, most of the approaches address one specific problem. Our contribution is an unsupervised rank-based approach capable of refining similarity information and computing a context-sensitive representation, which can be exploited for improving the effectiveness of both unsupervised retrieval and semi-supervised classification. We propose a novel manifold learning algorithm named Rank Flow Embedding (RFE). The proposed method is based on different and complementary ideas recently exploited by manifold learning approaches in order to provide a better contextual representation of dataset objects. The algorithm computes rank-based embeddings which are refined along the processing flow for each step. This approach constitutes a key innovation in the sense that constitutes an unsupervised contextual-sensitive method capable of computing a novel representation and not only a similarity measure.

Firstly, a rank-based formulation is used to define a hypergraph model capable of representing high-order similarity information encoded in ranked lists. The hypergraph is used for iterative re-ranking, based on the similarity among embeddings defined by hyperedges (*h-embeddings*). Next, Cartesian product operations are performed on hyperedges for maximizing their similarity relationships. While hyperedges effectively represent regional relationships, broader similarity relationships are also relevant. In this direction, hypergraph structures are also used to model a graph and define high-confident Connected Components (CCs), aiming at estimating class information of datasets. The information encoded in the CCs is exploited for a new re-ranking step and used as class representatives to compute low-dimensional embeddings. Such embeddings, in turn, can be exploited for more effective semi-supervised classification tasks.

The proposed method presents various contributions and innovations regarding related work. Among them:

- Most unsupervised context-sensitive approaches establish a novel similarity measure [5], [6], [31], but not a novel representation. Beyond that, this work proposes a novel rank-based approach for learning context-sensitive representations. More effective representations are fundamental for many applications, including unsupervised retrieval and semi-supervised classification, scenarios in which the method was evaluated;
- The proposed approach presents substantial innovations in the way of computing such representations. The embeddings and their encoded similarity information are refined through a flow of rank-based structures and operations. Although some strategies already have been individually exploited (graphs [25], hypergraphs [6], and connected components [26]), our work allows the sequential refinement of similarity information along these structures. In addition, the proposed approach includes relevant distinctions in how such structures are defined and used. More specifically: *(i)* The hypergraph model used is defined based on a novel rank normalization function, proposed in this work and named as reciprocal sigmoid; *(ii)* The computation of connected components is based on a ranking of candidate edges, which estimates the confidence of edges using the hypergraph embeddings. The strategy consists of a novel approach proposed in this work; *(iii)* The use of similarity to the connected components for defining the dimensions of novel representations is also an innovation proposed in this paper.
- The method can be used in scenarios where the queries are not part of the dataset (*unseen queries*), which is fundamental for many real-world applications and has been little exploited by related work in post-processing methods.

The effectiveness of the proposed method was confirmed with a wide and diversified experimental evaluation. The experimental results were obtained on 10 public datasets, including traditional image retrieval benchmarks and person Re-ID datasets. For each dataset, different features were considered including CNN and recent Vision Transformers features. On semi-supervised classification, the evaluation considered the proposed RFE embedding classified by different Graph Convolutional Network (GCN) models. An ablation study was also conducted in order to assess the impact of each step of the proposed method. The experimental evaluation also considers comparisons with other state-of-the-art approaches on various datasets. The results demonstrate the effectiveness of the proposed method on different tasks: unsupervised image retrieval, semi-supervised classification, and person Re-ID.

This paper is organized as follows. Section II presents the formal definition of addressed problems. Section III presents the proposed RFE method. Section IV describes the conducted experimental evaluation. Finally, Section V states conclusions and discusses the possible future works.

II. Problem Formulation

This section discusses the notation used and formal definitions of main tasks involved, mostly following related work [5], [14], [16]. Each task is discussed in the following subsections.

A. Feature Extraction and Similarity Computing

Although images are the focus of this paper, a more global definition using multimedia objects is used. The content of multimedia objects is represented by a feature extraction procedure. A d-dimensional representation is obtained and allows the pairwise comparison between objects. The comparison can be computed by two functions defined as:

- ϵ: $o_i \rightarrow \mathbb{R}^d$ is a function, which extracts a feature vector $v_{\hat{i}}$ from a multimedia object o_i;
- δ: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ is a function that computes the distance between two multimedia objects according to the distance between their corresponding feature vectors.

The distance between two objects o_i, o_j is computed as $\delta(\epsilon(o_i),\ \epsilon(o_j))$. The Euclidean distance is often employed to compute δ, although the proposed ranking method is independent of distance measures. A similarity measure $\rho(o_i, o_j)$ can be computed based on distance function δ and used for ranking tasks. The notation $\rho(i, j)$ is used along the paper.

B. Retrieval and Rank Model

Let $\mathcal{C}$=$\{o_1,\ o_2,\ \ldots, o_n\}$ be a multimedia collection, where $n = |\mathcal{C}|$ denotes the size of the collection $\mathcal{C}$. The target task refers to retrieving multimedia objects (images, videos) from $|\mathcal{C}|$ based on their content. Let o_q denotes a query object. A ranked list τ_q can be computed in response to the query o_q based on the similarity function ρ. The top positions of ranked lists are expected to contain the most similar objects to the query object.

Since τ_q can be expensive to compute when n is high, the ranked list considers only a sub-set of the collection. Formally, let τ_q be a ranked list that contains only the L most similar objects to o_q, where $L \ll n$. Let $\mathcal{C}_L$ be a sub-set of the collection $\mathcal{C}$, such that $\mathcal{C}_L \subset \mathcal{C}$ and $|\mathcal{C}_L| = L$. The ranked list τ_q can be defined as a bijection from the set $\mathcal{C}_L$ onto the set $[L] = \{1, 2, \ldots, L\}$. For a permutation τ_q, we interpret $\tau_q(i)$ as the position (or rank) of the object o_i in the ranked list τ_q. If o_i is ranked before o_j in the ranked list of o_q, i.e., $\tau_q(i) < \tau_q(j)$, then $\rho(q, i) \geq \rho(q, j)$.

Every object $o_i \in \mathcal{C}$ can be taken as a query o_q. A set of ranked lists $\mathcal{T} = \{\tau_1, \tau_2,\ \ldots,\ \tau_n\}$ can also be obtained, with a ranked list for each object in the collection $\mathcal{C}$. Based on the rank model, the neighborhood set can also be defined. Let o_q be a multimedia object taken as query, a neighborhood set $\mathcal{N}(q, k)$ that contains the k most similar multimedia objects to o_q can be defined as follows:

$$\mathcal{N}(q, k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in \mathcal{C} - \mathcal{S} : \tau_q(i) < \tau_q(j)\}. \quad (1)$$

C. Rank-based Manifold and Representation Learning

The proposed RFE method aims to capture the structure of the dataset manifold by exploiting the similarity information encoded in the set of ranked lists $\mathcal{T}$. As a result, the RFE are evaluated on two objectives: (*i*) computing a more effective similarity measure and ranking result for unsupervised retrieval and; (*ii*) computing a more effective embedding to represent each image, which can be used by other tasks, as semi-supervised classification.

Regarding unsupervised manifold learning, a new and more effective set of ranked $\mathcal{T}_r$ is computed with the aim of improving the effectiveness of ranking results. More formally, we can describe the method as function f_m:

$$\mathcal{T}_r = f_m(\mathcal{T}) \quad (2)$$

The aggregation problem is also considered, in which different sets of ranked lists $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_d\}$ are taken as input aiming at computing a more effective set $\mathcal{T}_r$.

Regarding representation learning, the objective is to compute an embedding that provides a more effective representation for a given object o_i based on the contextual similarity information encoded in $\mathcal{T}$. Formally, it can be defined as function f_e:

$$\mathbf{e}_i = f_e(\mathcal{T}, o_i), \quad (3)$$

where $\mathbf{e}_i$ is a vector on a d_e-dimensional embeding space.

III. Proposed Method

How to effectively design context-aware measures is a challenging question, which is closely associated with how to represent each image in terms of the collection in which is contained. Analogous to convolution and pooling operations used on CNNs, the proposed *Rank Flow Embedding* (RFE) employ subsequent rank-based operations in order to define more effective contextual representations. In fact, representations are derived from similarity to other images modeled by rank information. Such representations, in turn, are used to derive more effective similarity measures. Such mechanism is repeated through a flow of distinct and complementary operations in order to extract the maximum of available contextual information.

Figure 1 presents the main steps of the proposed approach and the respective workflow. The proposed manifold learning algorithm can be used for unsupervised re-ranking, producing ranked lists as output retrieval results, or for representation learning, producing contextual vector representations. The method can be summarized by the following steps:

1) **Ranked Lists Normalization**: ranked lists are recomputed considering a sigmoid score computed based on the reciprocal ranked lists positions;
2) **Re-ranking by Hypergraph Embeddings**: an iterative step that employs a hypergraph structure to analyze the underlying similarity information contained in the ranked lists. This step defines the h-embeddings and hyperedge weights, which are used by next steps;
3) **Re-Ranking by Cartesian Product**: a Cartesian product step is used to spread the similarity information among elements in the same hyperedge;
4) **Re-ranking by Connected Components**: high-confident connected components (CCs) are defined based on hypergraph structures (Step 2). The CCs are computed based on the most confidential edges identified through the hyperedge weights. The CCs encode class information and cause objects in the same CC to have their similarities increased;
5) **Embeddings by Connected Components**: more effective embeddings are computed for each dataset element considering their similarity to the identified CCs. This step is directed for semi-supervised classification, since a low-dimensional embedding is obtained.

Each stage is detailed and formally defined in the next sections. In general, each step incrementally improves the effectiveness of rank-based similarity information and computes structures which are exploited in next steps. While Steps 1-3 are suitable for general retrieval tasks, the Step 4 is focused on datasets with larger similarity groups, in which information from CCs can be better exploited. Hence Step 4 is not suitable for datasets with large numbers of very small classes. Step 5 uses the constructed structures for computing embeddings used for classification. Besides the standard retrieval pipeline and the embeddings for classification, rank aggregation tasks and the use of unseen queries are also discussed.

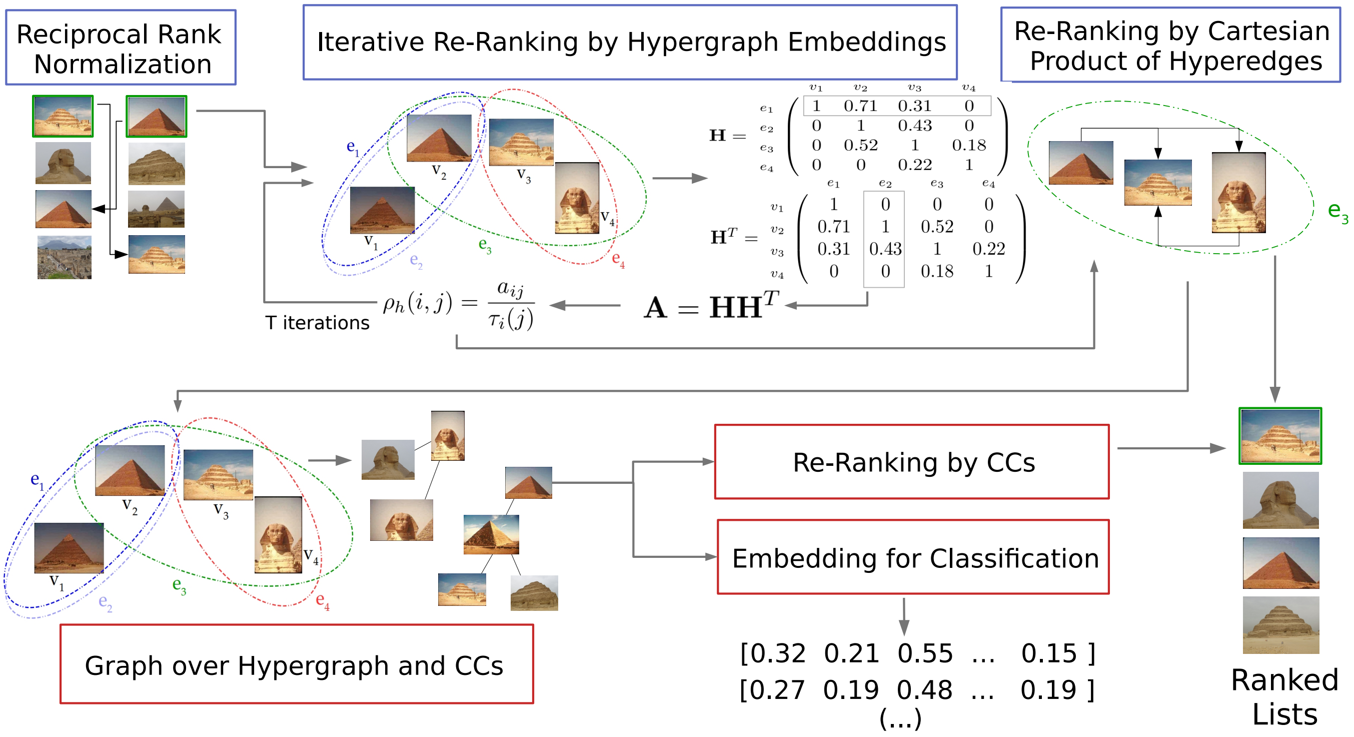

Reciprocal Rank Normalization
Iterative Re-Ranking by Hypergraph Embeddings
Re-Ranking by Cartesian Product of Hyperedges
e_1
e_2
e_3
e_4
v_1
v_2
v_3
v_4
$\mathbf{H} = \begin{array}{c|cccc} & v_1 & v_2 & v_3 & v_4 \\ \hline e_1 & 1 & 0.71 & 0.31 & 0 \\ e_2 & 0 & 1 & 0.43 & 0 \\ e_3 & 0 & 0.52 & 1 & 0.18 \\ e_4 & 0 & 0 & 0.22 & 1 \end{array}$
$\mathbf{H}^T = \begin{array}{c|cccc} & e_1 & e_2 & e_3 & e_4 \\ \hline v_1 & 1 & 0 & 0 & 0 \\ v_2 & 0.71 & 1 & 0.52 & 0 \\ v_3 & 0.31 & 0.43 & 1 & 0.22 \\ v_4 & 0 & 0 & 0.18 & 1 \end{array}$
$\mathbf{A} = \mathbf{H}\mathbf{H}^T$
$\rho_h(i,j) = \frac{a_{ij}}{\tau_i(j)}$
T iterations
Re-Ranking by CCs
Embedding for Classification
Graph over Hypergraph and CCs
[0.32 0.21 0.55 ... 0.15]
[0.27 0.19 0.48 ... 0.19]
(...)
Ranked Lists

Fig. 1: Overall organization of Rank Flow Embedding: in blue boxes the initial steps and in red boxes optional steps for refining retrieval and for computing embedding for semi-supervised classification.

A. Rank Normalization by Reciprocal Sigmoid

In opposite to the majority of distance measures, the ranking information is not symmetric. The increase of symmetry generally produces a positive impact on the effectiveness of similarity information, widely exploited by reciprocal rank analysis [17], [19]. However, most of the reciprocal approaches apply linear analysis to rank positions. In this paper, we use a non-linear scoring function that assigns high weights to top-rank positions, with a fast decay around the neighborhood size, given by k. With this objective, a sigmoid function is applied. Additionally, a higher relevance is assigned to the original rank position (squared) in comparison with the reciprocal rank position (linear). The new similarity between objects o_i and o_j is defined by ρ_n:

$$\rho_n(i,j) = \sigma(i,j)^2 \times \sigma(j,i). \tag{4}$$

The function σ which assigns weights according to rank positions is defined as:

$$\sigma(x,y) = 1 - \frac{1}{1 + e^{-\alpha(\tau_x(y)-k/2))}}, \tag{5}$$

where α is a constant empirically evaluated in the experimental analysis.

Based on the measure ρ_n, which is computed between the objects in the top-L positions, the ranked lists are updated with a stable sorting algorithm. The stable sorting is used in order to keep the position in the case of a tie. An updated set of ranked list $\mathcal{T}_n$ is obtained as output.

B. Re-Ranking by Hypergraph Embeddings

The contextual representation model used for data elements and how to exploit it to compute more effective similarity measures is a fundamental task in rank-based manifold learning. In this work, we use a hypergraph model based on ranking information inspired by [14], [32]. The hypergraph establishes relations among set of objects, allowing to represent high-order similarity relationships. The proposed RFE method compute contextual embeddings based on hypergraph information and define an iterative re-ranking procedure based on comparison of such embeddings.

*1) **Hypergraph Embeddings**:* Formally, a hypergraph model is defined by a tuple $H = (V, E_h, w)$, where V represents a finite set of vertices and E_h denotes the set of hyperedges. The hyperedges set E_h can be defined as the family of subsets of V such that $\bigcup_{e_i \in E_h} = V$. A hyperedge e_i is said to be incident to a vertex v_j if $v_j \in e_i$. For each hyperedge e_i, a positive weight $w(e_i)$ is assigned, which denotes the confidence of the relationships established by the hyperedge e_i.

Each vertex $v_i \in V$ represents an object in the collection: $o_i \in \mathcal{C}$. For each object, a hyperedge is created by exploiting first and second-order neighborhood information. A hyperedge e_i is defined based on the neighborhood set of o_i and its respective neighbors. Formally, let $o_x \in \mathcal{N}(i,k)$ be a neighbor of o_i and let $o_j \in \mathcal{N}(x,k)$ be a neighbor of o_x, the hyperedge e_i is defined as:

$$e_i = \mathcal{N}(i,k) \bigcup_{o_x \in \mathcal{N}(i,k)} \mathcal{N}(x,k). \tag{6}$$

Consequently, each image o_i is now also represented by a hyperedge e_i. Since the number of hyperedges is equal to the number of vertices, the obtained hypergraph can be represented by a square incidence matrix $\mathbf{H}_m$ of size $|E_h| \times |V|$, where elements $\mathbf{H}_m$ are define as:

$$h_m(e_i, v_j) = \begin{cases} r(e_i, v_j), & \text{if } v_j \in e_i, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Row i of h_m tells which vertices belong to hyperedge e_i and the score $r(e_i, v_j)$ indicates the degree of belonging of the vertex v_j to hyperedge e_i. The score r is computed according to the number and relevance of mentions to v_j in the hyperedge e_i and is defined as:

$$r(e_i, v_j) = \sum_{o_x \in \mathcal{N}(i,k) \wedge o_j \in \mathcal{N}(x,k)} w_p(i,x) \times w_p(x,j), \tag{8}$$

where $w_p(i,x)$ is a function that assigns a weight of relevance to o_x according to the position in the ranked list τ_i. Notice that the score r incorporates information from first and second-order ranking references, i.e., from neighbors and neighbors of neighbors. The weight assigned to o_x according to the position of the ranked list τ_i is defined by a log-based function as:

$$w_p(i,x) = 1 - \log_k \tau_i(x). \tag{9}$$

The function $w_p(i,x)$ reaches the maximum value of 1, which is assigned to the first position of the ranked lists and corresponds to the query image. For the subsequent positions in the ranked lists, the function decays fast.

While the hyperedge e_i provides a more comprehensive contextual representation for the object o_i, it can also be susceptible to noise in certain circumstances. As it considers second-order similarity relationships, non-relevant objects in rankings of neighbors can generate undesired references in the hyperedge e_i. With the aim of filter out such cases, we include a consistency check among hyperedges in order to obtain a more precise representation.

The main idea consists in verifying for each element in the hyperedge e_i how it is referenced by other hyperedges. Most of objects in e_i are expected to be relevant and compose a consistent set of high-similarity among each other. Thus, a given relevant object $o_j \in e_i$ is expected to be referenced with high scores in the other hyperedges which represents most of elements in e_i. On the other hand, a noisy and non-relevant object $o_n \in e_i$ is not expected to be referenced in the same hyperedges.

In this way, the filtered score for a given object $o_j \in e_i$ is computed by multiplying scores in e_i by the score of o_j in hyperedges of elements referenced in e_i, which can be obtained by a matrix $\mathbf{H}$ computed as

$$\mathbf{H} = {\mathbf{H}_m}^2. \tag{10}$$

The computation of matrix $\mathbf{H}$ defines the embeddings provided by the hypergraph model to represent each object, which we denote as *h-embeddings*. For an object o_i, its respective h-embedding can be defined by the correspondent row of matrix $\mathbf{H}$, such that:

$$\mathbf{h}_i = [h_{i1}, h_{i2}, \ldots, h_{in}], \tag{11}$$

where h_{ij} defines the similarity of object o_j in the hyperedge e_i, also denoted as $h(i,j)$.

The definition of the hypergraph also includes a confidence of each hyperedge, given by the function $w(e_i)$. A highly-effective hyperedge is expected to contain a consistent set of vertices. Therefore, it is expected to contain only a few vertices with high score values given by $h(e_i, \cdot)$. Hence, the weight $w(e_i)$ is defined as:

$$w(e_i) = \sum_{j \in \mathcal{N}_h(i,k)} h(i,j), \tag{12}$$

where $\mathcal{N}_h(i,k)$ is a neighborhood set defined among the elements with top $h(e_i, \cdot)$ score values in the hyperedge. The $\mathcal{N}_h$ set containing the vertices with the highest values of $h(e_i, \cdot)$ is formally defined as:

$$\mathcal{N}_h(q,k) = \{\mathcal{S} \subseteq e_q, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in e_q - \mathcal{S} : h(q,i) > h(q,j)\}. \tag{13}$$

Based on the previous equations, we can define a function $f_h(\cdot)$ that, given a set of ranked lists $\mathcal{T}_n$ as input, computes a hypergraph H and its respective *h-embeddings* given by the matrix $\mathbf{H}$. The function is defined as follows:

$$(H, \mathbf{H}) = f_h(\mathcal{T}_n). \tag{14}$$

In fact, the matrix $\mathbf{H}$ and the weight of edges $w(.)$ contain the main similarity information encoded in the hypergraph model. Both structures are exploited by the proposed RFE method and refereed along the paper. Firstly, the information encoded in matrix $\mathbf{H}$ is exploited to define a contextual similarity measure used for re-ranking.

*2) **Hypergraph-based Re-Ranking**:* While similar objects present similar ranked lists, it is expected that the respective h-embeddings are also similar. Once the similarity information is encoded in the matrix $\mathbf{H}$, a similarity measure between two embeddings $\mathbf{h}_i$ and $\mathbf{h}_j$ can be computed by its product $\mathbf{h}_i\mathbf{h}_j$. This operation can be modeled for all the objects by multiplying the matrix $\mathbf{H}$ by its transpose, with the objective of obtain the affinity matrix $\mathbf{A}$, defined as follows:

$$\mathbf{A} = \mathbf{H}\mathbf{H}^T. \tag{15}$$

The elements of matrix $\mathbf{A}$ given by a_{ij} denote the similarity between objects o_i, o_j. The matrix $\mathbf{A}$ contains most of the similarity information extracted based on the hypergraph, such that it can be used to define a more effective similarity measure ρ_h. In addition, the proposed measure also considers a residual similarity information, given by the original ranking position. The measure is defined as:

$$\rho_h(i,j) = \frac{a_{ij}}{\tau_i(j)}. \tag{16}$$

Based on the similarity computed by the function ρ_h, an updated set of ranked lists ${\mathcal{T}_h}^{(t)}$ is obtained by applying a stable sorting algorithm. The ranked lists, in turn can be used to compute a novel hypergraph and the procedure can be iteratively repeated, such that the superscript $^{(t)}$ denotes the iteration.

After a certain number of T iterations, the set of ranked lists ${\mathcal{T}_h}^{(T)}$ is provided to the function f_h, which returns a matrix $\mathbf{H}_a$ and a updated hypergraph H_a, used in next steps of the rank flow. The index a is used to indicate that they were obtained based on the affinity matrix:

$$(H_a, \mathbf{H}_a) = f_h({\mathcal{T}_h}^{(T)}). \tag{17}$$

C. Re-Ranking by Cartesian Product

A Cartesian product step is used to expand the similarity information contained in the updated set of hyperedges E_h^a. Inspired by [14], [33], the procedure exploits high-order similarity relationships represented on hyperedges to compute more effective pairwise measures. Formally, given two hyperedges $e_q, e_i \in E_h^a$, the Cartesian product between them can be defined as:

$$e_q \times e_i = \{(v_x, v_y) : \ v_x \in e_q \wedge v_y \in e_i\}. \tag{18}$$

The notation ${e_q}^2$ is used aiming to indicate the Cartesian product between elements of the same hyperedge e_q, such that $e_q \times e_q = {e_q}^2$. For each pair of vertices $(v_i, v_j) \in {e_q}^2$ a pairwise relationship $p : E_h^a \times V \times V \rightarrow \mathbb{R}^+$ is established.

A value p is computed based on the weight $w(e_q)$, which indicates the level of confidence of the hyperedge that originated the association. As previously mentioned, the weight $w(e_i)$ can be interpreted as the confidence estimations of associations encoded on hyperedge e_i The degrees of association of v_i and v_j are defined by:

$$p(e_q, v_i, v_j) = w(e_q) \times h(e_q, v_i) \times h(e_q, v_j). \quad (19)$$

A pairwise similarity measure based on the Cartesian product is defined considering relationships contained in all the hyperedges. This formulation presents the idea of exploiting the co-occurrence of v_i and v_j in different hyperedges, performing a sum of all the values of $p(\cdot, v_i, v_j)$:

$$\rho_c(i,j) = \sum_{e_q \in E \wedge (v_i, v_j) \in {e_q}^2} p(e_q, v_i, v_j). \quad (20)$$

Based on the similarity function ρ_c, a more effective set of ranked lists $\mathcal{T}_c$ is computed by a stable sorting algorithm. The ranked lists set $\mathcal{T}_c$ is provided to the function f_h that computes an updated hypergraph and h-embeddings. The index c is used to indicate that they were obtained after the Cartesian product step:

$$(H_c, \mathbf{H}_c) = f_h(\mathcal{T}_c). \quad (21)$$

D. Graph over Hypergraph and Connected Components

Although the hypergraph model provides an effective tool to represent regional similarity information, it does not represent the similarity among objects in the same class/cluster but more distant in the dataset manifold. In order to represent such information, a high-confident graph is defined based on h-embeddings computed after Cartesian product operations. The Connect Components are extracted from this graph and are used to represent class information and the global structure of similarity relationships encoded in the dataset.

*1) **Graph Definition**:* Formally, the graph is defined as G = (V, E), such that the set of vertices $V = \mathcal{C}$, where each node represents a collection object. The set of edges E is computed based on information provided by the hypergraph representation. Firstly, a set of candidate edges $\mathcal{E}_c$ is defined based on the neighborhood set of each object as:

$$\mathcal{E}_c = \bigcup_{q \in V} \bigcup_{i \in \mathcal{N}(q,k)} \{(q,i)\}. \quad (22)$$

In order to select the most confident edges, the set of candidates are ranked. The ranked list τ_c is defined as a permutation of the set of candidate edges $\mathcal{E}_c$. The permutation τ_c is the bijection of the set $\mathcal{E}_c$ onto the set $[n_k] = \{1, 2, \ldots, n_k\}$, The position of the pair (q,i) in the ranked list is denoted by $\tau_c((q,i))$. The permutation is defined such that if (q,i) is ranked before (j,l), e.g, $\tau_c((q,i)) < \tau_c((j,l))$, then $s_c(q,i)$ $\geq s_c(j,l)$. The function s_c is a similarity measure attributed to pairs based on the similarity between h-embeddings and confidence of the hyperedge, defined as:

$$s_c(i,j) = \mathbf{h}_{c_i} \mathbf{h}_{c_j}^T \times w(e_i) \times w(e_j), \quad (23)$$

where the pair (i,j) identifies a pairs of hyperedges $e_i, e_j \in E_h^c$, and E_h^c denotes a set of hyperedges of the hypergraph H_c. Once ranked, a threshold should be established in order to defined the number of edges that are created. The threshold t_c is defined as:

$$t_c = \frac{\sum_{e_q \in E_h^c} w(e_q)}{2 \times n}. \quad (24)$$

The edge set E is be defined using the threshold t_c as

$$E = \{(o_q, o_i) \mid (q,i) \in \mathcal{E}_c \wedge \tau_c((q,i) < t_c\}. \quad (25)$$

The process of building the graph can be understood as a function f_g that receives as input a hypergraph H_c and a matrix $\mathbf{H}_c$ (output of the Cartesian product) and computes a graph G:

$$G = f_g(H_c, \mathbf{H}_c). \quad (26)$$

*2) **Connected Components**:* Based on the defined graph, its respective Connected Components (CC) are extracted. Formally, each CC is defined as a set of objects $\mathcal{C}_i$. Given two objects o_i, $o_j \in \mathcal{C}_l$, there is a path (edge) between o_i, o_j. Search algorithms in graphs (e.g. Depth and Breadth-First) and Tarjan algorithm can be used to compute the CCs. The output for the dataset is provided by the set of connected components $\mathcal{S} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$, such that $\bigcup_{\mathcal{C}_i \in \mathcal{S}} = \mathcal{S}$ and $\bigcap_{\mathcal{C}_i \in \mathcal{S}} = \emptyset$.

The connected components are sets of similar objects and it is expected that such structures encode the information of sets or classes of the dataset. Following this reasoning, an embedding is created based on the *h-embeddings* of the elements that are part of it. Given a connected component q, the cc-embedding $\mathbf{c}_q$ is defined as:

$$\mathbf{c}_q = \sum_{o_i \in \mathcal{C}_q} \mathbf{h}_{c_i}. \quad (27)$$

Once the Connected Components (CCs) encode information associated to representation of classes, the similarity to such CCs embeddings can be exploited for computing a more globally contextual similarity measure. In this way, a novel embedding is computed for each object according to its similarity to the CCs embeddings. Formally, let $\mathbf{e}_q$ be an embedding of an object of index q. The computation of the value of position i of this vector (embedding) is done as follows:

$$\mathbf{e}_q[i] = \mathbf{h}_{c_q} \mathbf{c}_i^T, \quad (28)$$

where i identifies the connected component $\mathcal{C}_i \in \mathcal{S}$ and $\mathbf{c}_i$ denotes the embedding that corresponds to this CC. In this way, the embeddings can be computed for each element of the dataset.

*3) **Re-Ranking by Connected Components**:* The re-ranking by CCs exploits information about elements in the same CC. In this way, the elements that present high similarity values in the same CC, have their similarities increased. The first step of this process consists into define the k elements with the highest values in each connected component. A neighborhood set $\mathcal{N}_c(q,k)$ is defined for each element of index q considering a constant k:

$$\mathcal{N}_c(q,k) = \{\mathcal{S} \subseteq \mathcal{C}, |\mathcal{S}| = k \wedge \forall o_i \in \mathcal{S}, o_j \in \mathcal{C} - \mathcal{S} : \mathbf{c}_q[i] > \mathbf{c}_q[j]\}. \quad (29)$$

The ranked list τ_{c_q} can be defined as the permutation of objects that have the k highest values in the embedding $\mathbf{c}_q$. The permutation is defined as the bijection of the set $\mathcal{N}_c(q,k)$ to the set $[k] = \{1, 2, \ldots, k\}$. The position of an object o_i in the ranked list computed by the embedding of the connect component $\mathbf{c}_q$ is defined as $\tau_{c_q}(i)$. If o_i is ranked before o_j in a ranked list, this means, $\tau_{c_q}(q,i) < \tau_{c_q}(q,j)$, therefore $\mathbf{c}_q[i] \geq \mathbf{c}_q[j]$.

The re-ranking by CCs exploits three complementary information: (*i*) the similarity between embeddings; (*ii*) the object belonging to the same connected component and; (*iii*) the residual information of rank position. The similarity $\rho_e(i,j)$ is defined in order to combine such information, formally defined as:

$$\rho_e(i,j) = \sum_{o_i,o_j \in \mathcal{N}_c(q,k)} \frac{1+\sqrt{\tau_{c_q}(q,i)^2+\tau_{c_q}(q,j)^2} \times \mathbf{e}_i\mathbf{e}_j^T}{\tau_i(j)}. \tag{30}$$

Based on the similarity function ρ_e, a set of ranked lists $\mathcal{T}_e$ is obtained by a stable sorting algorithm. The set of ranked lists $\mathcal{T}_e$ is provided to the function f_h that computes a new hypergraph H and matrix $\mathbf{H}$. The index e is used to indicate that they were obtained after the step of the connected components:

$$(H_e, \mathbf{H}_e) = f_h(\mathcal{T}_e). \tag{31}$$

E. Embeddings for Classification

The class information encoded in the re-ranking by CCs can be useful for other machine learning tasks. In this way, novel representations are computed for dataset objects and used as embeddings for semi-supervised classifiers. Given the ranked lists $\mathcal{T}_e$ and the hypergraph H_e obtained in the previous step, we obtain a graph with the updated connected components following the same equations defined in Section III-D. Thus, the updated graph is defined as follows:

$$G_e = f_g(H_e, \mathbf{H}_e). \tag{32}$$

The new connected components, considering the component $\mathbf{c}$ after the step of CC (index e) for the element q, are obtained as follows:

$$\mathbf{c}_{e_q} = \sum_{o_i \in \mathcal{C}_{e_q}} \mathbf{h}_{e_i}. \tag{33}$$

Finally, each of the positions of the embedding vector, which is going to be used for classification, computed as follows:

$$\mathbf{e}_{e_q}[i] = \mathbf{h}_{e_q}\mathbf{c}_{e_i}^T, \tag{34}$$

where the index e indicates that the variables were obtained after the re-ranking by the connect components. The contextual embedding $\mathbf{e}_{e_q}$ is used as features by semi-supervised classifiers.

F. Unseen Queries

The formulation used by RFE considered an already known dataset, where all the elements of the dataset can be taken as queries. However, RFE also allows to perform queries with elements that does not belong to the dataset, in a formulation known in the literature as unseen queries. To make this possible, RFE follows a strategy proposed in [8] by decoupling off-line procedures (for the whole dataset) of on-line procedures (for the unseen query).

On off-line setting, the conventional steps of the method (normalization, re-ranking by embeddings, Cartesian product, re-ranking by connected components) are normally executed for all the known elements in the dataset. So, when a new external query (unseen query) need to be evaluated, the k most similar elements are computed for each of them and a h-embedding is generated for the new query. The cosine distance between the query embedding and pre-computed embeddings in the whole dataset is used to rank the unseen query, producing the ranked lists for such elements.

G. Rank Aggregation

The RFE can also be exploited to fuse different features, in rank aggregation tasks. Different ranked lists sets $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_d\}$ are used as input with the objective of computing a more effective output set $\mathcal{T}_r$. The normalization step is performed individually for each of the rankers and the values are accumulated in a single sparse matrix M_f, once only top-L positions are considered. New ranked lists $\mathcal{T}_f$ are obtained by the sorting objects based on scores given by the matrix M_f. After that, the RFE (which can be understood as a function f_r) is executed for the ranked lists $\mathcal{T}_f$ and the list $\mathcal{T}_r$ is obtained as result:

$$\mathcal{T}_r = f_m(\mathcal{T}_f). \tag{35}$$

IV. Experimental Evaluation

This section discusses the experimental evaluation conducted to assess the effectiveness of the proposed method. Section IV-A describes the datasets and experimental settings. Section IV-B discusses the impact of parameters while Section IV-C presents an ablation study that includes an analysis of the impact of each step in our proposed method. Section IV-D and IV-E present the results on unsupervised retrieval and semi-supervised classification tasks, respectively. The results for unseen queries are described in Section IV-F. Sections IV-G and IV-H compare RFE with other state-of-the-art approaches for retrieval and classification, respectively. Finally, Section IV-I presents a visual analysis for both tasks.

A. Experimental Settings

A broad experimental evaluation was conducted on 10 different image datasets, which are presented in Table I. The datasets vary in size from 400 to 72,000 images. In this work, there are two different experimental scenarios: *(i)* unsupervised image retrieval, which was assessed on all datasets; and *(ii)* semi-supervised image classification conducted on the Flowers and Corel5k datasets. The retrieval category encompasses not only general-purpose image datasets, but also person Re-ID datasets (i.e., CUHK03, Market, Duke).

Due to the highly diverse aspects of each dataset, we employed different evaluation measures in each case to enable comparisons with other approaches. In the classification task, we used accuracy as the evaluation measure. In contrast, for the retrieval task, other measures were used, with Mean Average Precision (MAP) being the most common. For Re-ID datasets, the R1 (which, in this case, is equivalent to Precision@1) was included, since it is commonly reported in the literature. For the UKbench dataset, which has the smallest number of images per class (only 4), the NS-Score was used. The NS-Score is the average of correct images at the top-4 positions of the ranked lists.

We adopted the evaluation protocol for each dataset based on common practices in the literature. For most of them, all the images were considered as queries, except for Holidays [34] and Re-ID ones, where a different protocol was adopted [35]–[37]. For Holidays, there is a specific set of queries [34]. Regarding Re-ID, each dataset has a set of queries and a corresponding gallery set [35]–[37], which is the set of images that are ranked in relation to the query.

A comprehensive set of descriptors (features) were used considering both traditional and deep learning extractors, including Convolutional Neural Networks (CNNs) and Visual Transformers (VIT). For most of the datasets, a similar set

TABLE I: Datasets considered in the experimental evaluation.

Dataset Name	Num. of Classes	Dataset Size	Evaluation Measures
ORL Faces [38]	40	400	Recall@15
Flowers [39]	17	1,360	Accuracy, MAP
MPEG-7 [40]	70	1,400	Recall@40
Holidays [34]	500	1,491	MAP
Corel5k [41]	50	5,000	Accuracy, MAP
UKBench [42]	2,550	10,200	NS-Score, MAP
CUHK03 [35], [43]	1,467	14,097	R1, MAP [35]
Market1501 [36]	1,501	32,217	R1, MAP [36]
DukeMTMC [37]	1,812	36,411	R1, MAP [37]
ALOI [44]	1,000	72,000	MAP

of descriptors were used to keep the evaluation consistent. All the CNNs were trained on ImageNet dataset[1]. For Re-ID datasets (i.e., CUHK03, Market, Duke), we used CNNs which are more specific for Re-ID and trained on the MSMT17 dataset, extracted using *torchreid*[2].

The semi-supervised classification relies on Graph Convolutional Networks (GCNs), which are stochastic. Since the results of the executions vary, we report an average of 5 executions on 10 different folds. This was adopted for our method and all the baselines. For unsupervised retrieval, the executions are deterministic.

B. Parametric Space Analysis

Initially, an experiment was conducted to visualize the impact of parameter α in the reciprocal sigmoid function, which is used in order to compute the rank normalization. This is the first step of our proposed approach, described in Section III-A. The normalization mainly relies on Equation 5, which defines a reciprocal sigmoid function (σ). Figure 2 presents the values for Equation 5 (σ in y-axis) as the Rank Position ($\tau_x(y)$ in x-axis) varies. Different values of alpha were considered. The figure reveals that α is responsible for changing the steepness of the sigmoid curve, which refers to how quickly the output of the function changes as the input (i.e., the rank position) increases. However, it is challenging to determine an appropriate value of α based solely on this plot.

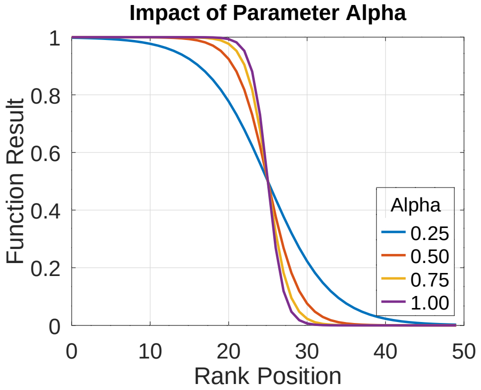


Fig. 2: Impact of parameter α in function σ (Equation 5) as the rank position varies.

Based on this issue, an analysis was conducted with the objective of identifying default parameters. Figure 3 presents the impact of parameters α and T (number of iterations) on the MAP results for two datasets (i.e., Flowers and Corel5k). The CNN-ResNet [45] was considered for this experiment. Since we are not evaluating the parameter k in this case, we set it to the number of elements per class ($k = 100$). This is done to keep the focus of the analysis on α and T. The surface shows that the lowest values of α and T are more appropriate. Notice that the set of parameters (T, α) = $(2, 0.1)$ is close to the best results in all cases (a, b, and c). Therefore, we used these values for all subsequent experiments.

C. Ablation Study

An ablation study was conducted to analyze the effectiveness of each step of the proposed method on 6 different datasets. We evaluated the retrieval results incrementally from Steps 1 to 4, as discussed in Section III. Step (0) corresponds to the original features, Step (1) involves ranked lists normalization, Step (2) performs re-ranking by hypergraph embeddings, Step (3) computes re-ranking by Cartesian product, and Step (4) re-ranks by connected components. In this case, we excluded Step (5), which generates embeddings, as it is only necessary for semi-supervised classification.

Figure 4 presents the effectiveness results for every step of the proposed approach. For each dataset, two descriptors were evaluated. The descriptors considered were SWIN-TF [46], VIT-B16 [47], Inner Distance Shape Context (IDSC) [48], Contour Features Descriptor (CFD) [49], OSNET-AIN [50], and OSNET-IBN [50]; which are among the top-performing ones. The experiment was conducted using the best value of k in each case. Notice that the values consistently increase along the performed steps, indicating the relevance of each step. However, the datasets Holidays and Ukbench (c and e) revealed a different behavior, where Step 4 slightly decreases the MAP. This is probably caused by the fact that different from others, these datasets have a small number of images per class. Therefore, all the subsequent retrieval results presented in the next sections include Steps 1-4, except for UKBench and Holidays datasets, which use Steps 1-3.

D. Retrieval Results

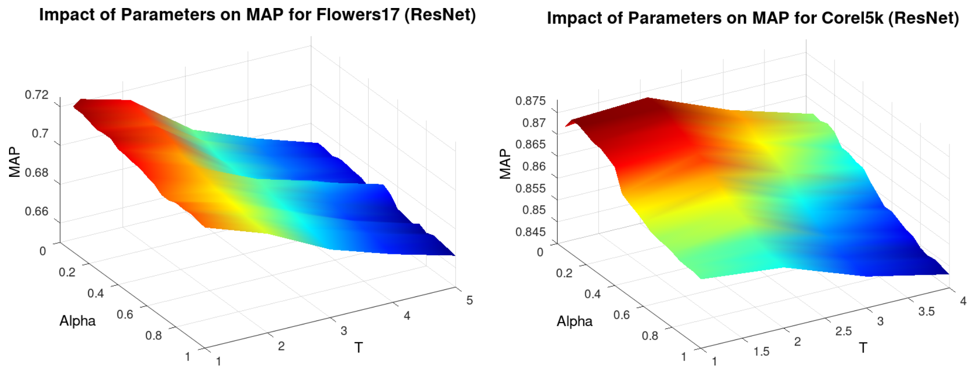
In image retrieval tasks, there are two different scenarios, which are both included in our evaluation: *(i)* standard re-ranking, where only one descriptor (feature) is considered; and *(ii)* rank-aggregation, which combines one or more features. For all experiments, we considered two variations for the parameter k (size of the neighborhood set): a default value [3] and the best value. The best k is reported considering the executions with k in range $[5, 120]$ with increments of 5. In general, the results revealed that our method is very robust to the change of k.

Firstly, we evaluate RFE on Flowers, Corel5k, and ALOI datasets; which are general-purpose image datasets that use the same protocol and evaluation measure. Table II presents the results. For standard re-raking, a relative gain was reported considering the improvement in relation to the original input descriptor. Since many descriptors are combined in rank aggregation, a gain is not reported in these scenarios. Notice that for all the cases, significant gains were obtained (up to +50.84%), and the fusion was able to improve the results even further. The best result for each dataset is highlighted in bold and marked with a gray background. For the three datasets, the best MAP is above 95%.

[1] https://github.com/Cadene/pretrained-models.pytorch

[2] https://github.com/KaiyangZhou/deep-person-reid

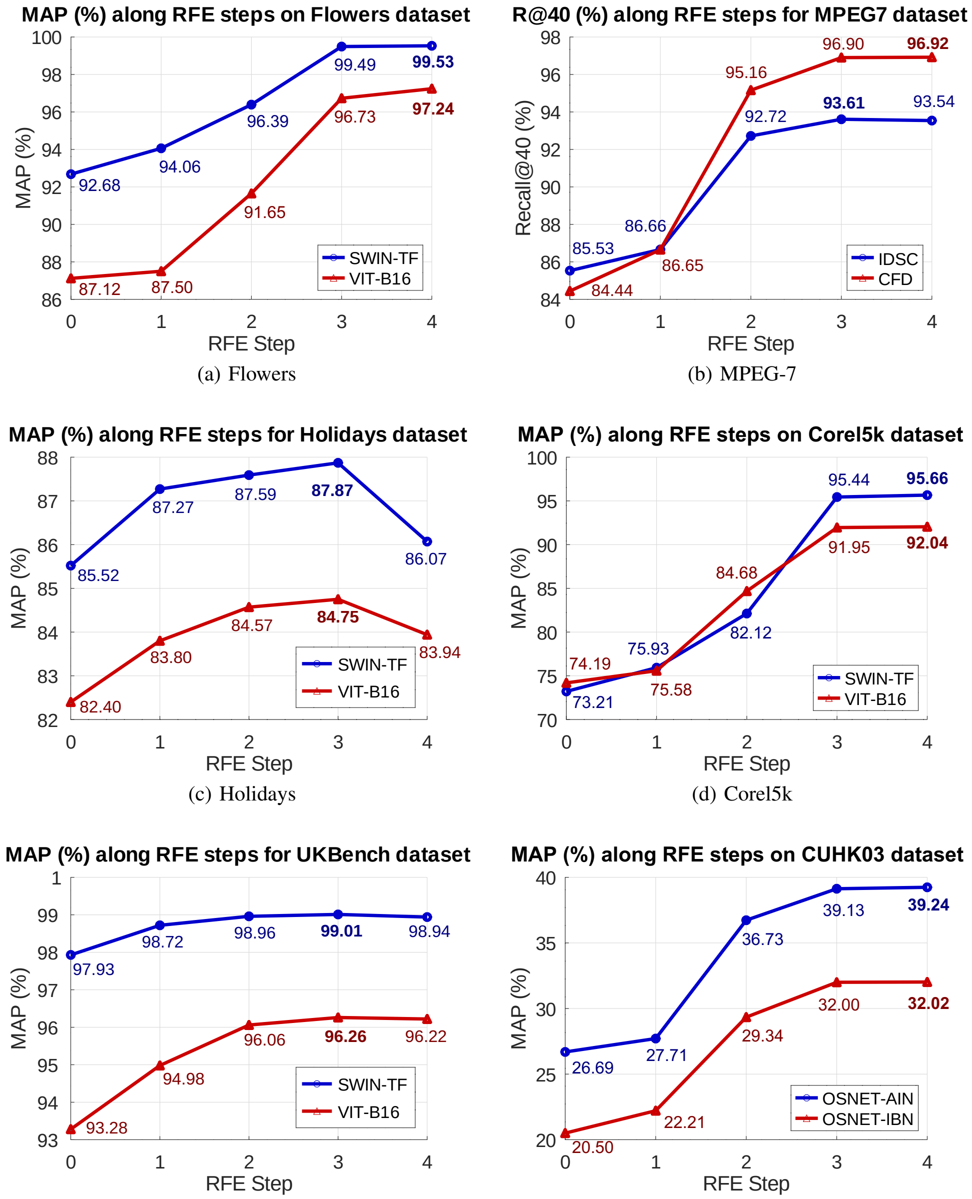
[3] The default values are: $k = 60$ for Flowers and Corel5k; $k = 5$ for Holidays and UKBench; and $k = 20$ for all the others.



(a) Flowers

(b) Corel5k

Fig. 3: Impact of parameters α and T (number of iterations) on MAP for two datasets.



(a) Flowers

(b) MPEG-7

(c) Holidays

(d) Corel5k

(e) UKBench

(f) CUHK03

Fig. 4: Ablation study on six datasets considering two descriptors each. The graphs present the effectiveness values (MAP or R@40 depending on the dataset) for each step of the proposed approach. The best value for each plot is highlighted in bold.

TABLE II: Retrieval results of the proposed method (RFE) on general purpose **image datasets (Flowers, Corel5k, and ALOI)**. The results are reported for **MAP (%)** evaluation measure considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values for each dataset are highlighted in bold with a gray background.

Descriptors	Original MAP	Method w/ default k	Method w/ best k	Relative Gain
		Flowers		
		Re-Ranking		
CNN-DPNet [51]	49.06	69.47	69.95 (k=70)	+42.58%
CNN-ResNet [45]	50.00	72.32	72.62 (k=75)	+45.23%
CNN-SENet [52]	40.85	61.26	61.26 (k=60)	+49.96%
CNN-Xception [53]	45.27	66.65	66.81 (k=65)	+47.57%
T2T-VIT24T [54]	38.03	54.99	55.03 (k=70)	+44.73%
VIT-B16 (VIT) [47]	87.12	92.28	97.24 (k=80)	+11.61%
SWIN-TF (STF) [46]	92.68	97.96	99.53 (k=85)	+7.39%
		Rank-Aggregation		
ResNet+DPNet	—	80.07	80.13 (k=75)	—
VIT+ResNet	—	94.63	97.67 (k=80)	—
VIT+STF	—	**98.07**	**99.65 (k=85)**	—
VIT+ResNet+STF	—	97.64	99.28 (k=90)	—
		Corel5k		
		Re-Ranking		
CNN-DPNet [51]	63.69	81.58	85.48 (k=100)	+34.22%
CNN-ResNet [45]	63.46	84.11	87.97 (k=100)	+38.61%
CNN-SENet [52]	55.57	78.77	83.38 (k=100)	+50.06%
CNN-Xception [53]	52.92	76.33	79.82 (k=90)	+50.84%
T2T-VIT24T [54]	58.97	80.46	84.10 (k=100)	+42.62%
VIT-B16 (VIT) [47]	74.19	90.02	92.04 (k=100)	+24.06%
SWIN-TF (STF) [46]	73.21	93.55	95.66 (k=105)	+30.70%
		Rank-Aggregation		
ResNet+DPNet	—	87.66	91.22 (k=100)	—
VIT+ResNet	—	93.28	95.01 (k=100)	—
VIT+STF	—	**95.39**	**96.79 (k=100)**	—
VIT+ResNet+STF	—	95.20	96.79 (k=100)	—
		ALOI		
		Re-Ranking		
CNN-DPNet [51]	79.09	94.45	96.32 (k=30)	+21.79%
CNN-ResNet [45]	81.97	94.79	96.37 (k=30)	+17.57%
CNN-SENet [52]	78.41	93.91	95.87 (k=30)	+22.27%
CNN-Xception [53]	76.07	93.40	95.36 (k=30)	+25.36%
T2T-VT24T [54]	76.90	93.46	95.36 (k=30)	+24.00%
VIT-B16 (VIT) [47]	79.40	93.55	95.40 (k=30)	+20.16%
SWIN-TF (STF) [46]	89.97	96.68	97.81 (k=30)	+8.71%
		Rank-Aggregation		
ResNet+DPNet	—	95.71	97.06 (k=30)	—
VIT+ResNet	—	95.70	97.13 (k=30)	—
VIT+STF	—	96.07	97.53 (k=30)	—
VIT+ResNet+STF	—	**96.59**	**97.73 (k=30)**	—

The same set of experiments was conducted for two datasets commonly used as image retrieval benchmarks: Holidays and UKbench. Since they have a small number of images per class, the best k is reported considering all the executions with k in the range $[1, 20]$ with increments of 1. Tables III and IV present the results for Holidays and Ukbench, respectively. As can be seen, expressive gains were obtained for both datasets and measures. For single descriptor executions, positive gains were obtained in all the cases, achieving gains up to +7.42%. For NS-Score, the results are very close to the maximum value, which is 4. It is also possible to notice a correlation between MAP and NS-Score values.

We also assessed RFE for person Re-ID (i.e., CUHK03, Market, and Duke datasets). These datasets are usually more challenging. They involve identifying and matching individuals across different camera views or even across different locations and times. People's appearances can vary significantly due to changes in lighting, pose, clothing, and accessories. These factors can make it difficult to match the same person in different images. Table V reports the results on these datasets. Since R1 is also commonly used for Re-ID evaluation, it was also included. The R1 corresponds to the first value of the CMC (Cumulative Matching Characteristics) curve, which indicates the number of ranked lists that have an image that corresponds to the same individual in the first position after the query image (which, in this case, is equivalent to Precision@1). The best k is reported considering all the executions with k in the range $[5, 50]$ with increments of 5. Notice that significant gains were obtained in all the cases (up to +65.88%), which were also improved by the rank-aggregation in most cases. These results reveal the potential of our approach in dealing not only with general-purpose scenarios but also with other challenging and more specific ones such as Re-ID.

TABLE III: Retrieval results of the proposed method (RFE) on the **Holidays dataset**. The results are reported for **MAP (%)** evaluation measure considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values are highlighted in bold with a gray background.

Descriptors	Original MAP	Method w/ default k	Method w/ best k	Relative Gain
		Re-Rank		
CNN-DPNet [51]	70.58	74.64	75.00 (k=6)	+6.25%
CNN-OLDFP [55]	88.46	89.58	90.11 (k=6)	+1.87%
CNN-ResNet [45]	74.87	77.15	77.37 (k=4)	+3.33%
CNN-SENet [52]	71.59	74.36	74.36 (k=5)	+3.88%
CNN-Xception [53]	64.93	68.24	68.48 (k=6)	+5.46%
T2T-VIT24T [54]	69.04	73.98	74.03 (k=6)	+7.23%
VIT-B16 (VIT) [47]	82.40	84.75	84.75 (k=5)	+2.85%
SWIN-TF (STF) [46]	85.52	87.87	87.87 (k=5)	+2.75%
		Rank-Aggregation		
VIT+ResNet	—	86.11	86.22 (k=6)	—
VIT+OLDFP	—	**91.64**	**91.97 (k=4)**	—
ResNet+OLDFP	—	88.08	88.33 (k=4)	—
OLDFP+STF	—	90.84	90.88 (k=4)	—
VIT+ResNet+OLDFP	—	89.98	90.35 (k=4)	—
VIT+OLDFP+STF	—	90.90	91.52 (k=4)	—

E. Classification Results

The proposed approach is capable of generating embeddings that can be utilized in various applications beyond retrieval. In this section, we employ RFE for semi-supervised classification on two general-purpose image datasets (i.e., Flowers and Corel5k). The process of embedding generation is unsupervised and encompasses all the steps of the proposed approach (from 1 to 5). Our hypothesis is that the RFE embeddings can be used to train semi-supervised classifiers, resulting in improved accuracy. We employed very recent Graph Convolutional Neural Networks (GCNs) models along with the traditional Support Vector Machine (SVM) with a polynomial kernel. The GCNs can operate on graphs, and they have become increasingly popular due to their ability to handle complex relationships between data points, which cannot be easily modeled using traditional machine learning methods.

Tables VI and VII present the results on Flowers and Corel5k datasets, respectively. In all the classifiers, the default parameters were used, proposed by the original authors. The GCNs were trained considering 50 epochs and $k = 40$ for the input kNN graphs. Our study compares the accuracy of classifiers that used the original features with those that used embeddings generated by the proposed RFE. We highlight in bold the best result for each classifier and in red the best for each dataset. The results demonstrate that the embeddings generated by our proposed approach are effective and have the potential to improve results across various classifiers. Notably, positive gains were obtained for all methods and features.

F. Unseen queries

Encountering scenarios where query images are not included in the dataset being evaluated is not uncommon. These

TABLE IV: Retrieval results of the proposed method (RFE) on the **UKBench dataset**. The results are reported for both **NS-Score and MAP** evaluation measures considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values are highlighted in bold with a gray background.

Evaluation Measure	**NS-Score**				**MAP (%)**			
Descriptors	**Original NS-Score**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**	**Original MAP**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**
		Re-Ranking				**Re-Ranking**		
CNN-DPNet [51]	3.46	3.71	3.72 (k=6)	+7.42%	90.47	94.58	94.67 (k=6)	+4.65%
CNN-OLDFP [55]	3.85	3.93	3.93 (k=5)	+2.24%	97.74	98.92	98.92 (k=5)	+1.21%
CNN-ResNet [45]	3.67	3.85	3.85 (k=6)	+4.94%	94.54	97.31	97.31 (k=5)	+2.93%
CNN-SENet [52]	3.56	3.76	3.76 (k=5)	+5.52%	92.15	95.55	95.55 (k=5)	+3.69%
CNN-Xception [53]	3.49	3.75	3.75 (k=6)	+7.60%	90.83	95.35	95.35 (k=6)	+4.99%
T2T-VIT24T [54]	3.48	3.75	3.75 (k=5)	+7.78%	90.26	95.40	95.40 (k=5)	+5.69%
VIT-B16 [47]	3.62	3.80	3.80 (k=6)	+5.00%	93.28	96.26	96.26 (k=5)	+3.19%
SWIN-TF [46]	3.86	3.94	3.94 (k=6)	+2.01%	97.93	98.98	99.01 (k=6)	+1.10%
		Rank-Aggregation				**Rank-Aggregation**		
VOC+OLDFP	—	3.90	3.90 (k=6)	—	—	98.22	98.22 (k=5)	—
VOC+ResNet	—	3.92	3.93 (k=6)	—	—	98.76	98.79 (k=6)	—
VOC+VIT-B16	—	3.92	3.92 (k=6)	—	—	98.69	98.77 (k=7)	—
OLDFP+ResNet	—	3.94	3.95 (k=6)	—	—	99.13	99.13 (k=5)	—
OLDFP+VIT-B16	—	3.93	3.94 (k=6)	—	—	98.94	98.99 (k=6)	—
ResNet+VIT-B16	—	3.91	3.91 (k=5)	—	—	98.45	98.45 (k=5)	—
OLDFP+SWIN-TF	—	**3.97**	**3.97 (k=6)**	—	—	**99.53**	**99.57 (k=6)**	—
VOC+OLDFP+ResNet	—	3.94	3.94 (k=6)	—	—	99.07	99.07 (k=5)	—
VOC+OLDFP+VIT-B16	—	3.94	3.95 (k=6)	—	—	99.09	99.13 (k=6)	—
VOC+ResNet+VIT-B16	—	3.94	3.95 (k=6)	—	—	99.13	99.15 (k=6)	—
OLDFP+ResNet+VIT-B16	—	3.94	3.94 (k=6)	—	—	99.07	99.08 (k=6)	—
OLDFP+ResNet+SWIN-TF	—	3.96	3.96 (k=6)	—	—	99.40	99.41 (k=6)	—
VOC+OLDFP+ResNet+VIT-B16	—	3.95	3.95 (k=6)	—	—	99.20	99.28 (k=7)	—
VOC+OLDFP+VIT-B16+SWIN-TF	—	3.96	3.96 (k=6)	—	—	99.36	99.43 (k=6)	—

TABLE V: Retrieval results of the proposed method (RFE) on three person **Re-ID datasets (CUHK03, Market, and Duke)**. The results are reported for both **R1 and MAP** evaluation measures considering re-ranking (single descriptor) and rank-aggregation (fusion of descriptors). The best values are highlighted in bold with a gray background (MAP as the criteria).

Evaluation Measure	**R1 (%)**				**MAP (%)**			
Descriptors	**Original R1**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**	**Original MAP**	**Method w/ default k**	**Method w/ best k**	**Relative Gain**
					CUHK03			
		Re-Ranking				**Re-Ranking**		
HACNN [56]	8.36	12.80	12.80 (k=20)	+53.03%	9.33	14.27	14.41 (k=15)	+54.42%
MLFN [57]	9.47	13.69	13.79 (k=15)	+45.63%	9.85	15.14	15.18 (k=15)	+54.11%
OSNet-AIN [50]	26.39	**36.67**	**36.89 (k=15)**	+39.76%	26.69	**39.12**	**39.24 (k=15)**	+47.00%
OSNet-IBN [50]	20.31	29.65	29.82 (k=15)	+46.85%	20.50	31.94	32.02 (k=15)	+56.18%
ResNet50 [45]	12.24	17.84	18.37 (k=15)	+50.15%	12.74	19.77	19.77 (k=20)	+55.18%
		Rank-Aggregation				**Rank-Aggregation**		
OSNet-AIN+OSNet-IBN	—	36.19	37.16 (k=15)	—	—	38.51	39.13 (k=15)	—
OSNet-AIN+ResNet50	—	33.54	33.54 (k=20)	—	—	35.40	35.40 (k=20)	—
OSNet-IBN+ResNet50	—	29.56	29.56 (k=20)	—	—	31.40	31.40 (k=20)	—
OSNet-AIN+OSNet-IBN+ResNet50	—	33.91	33.91 (k=20)	—	—	35.94	35.94 (k=20)	—
					Market			
		Re-Ranking				**Re-Ranking**		
HACNN [56]	49.23	52.20	52.82 (k=15)	+7.30%	22.29	31.93	32.10 (k=25)	+44.02%
MLFN [57]	46.59	49.58	49.76 (k=15)	+6.82%	21.11	30.65	30.89 (k=25)	+46.30%
OSNet-AIN [50]	69.95	70.99	70.99 (k=20)	+1.49%	42.33	57.38	58.21 (k=25)	+37.52%
OSNet-IBN [50]	66.45	67.25	67.90 (k=15)	+2.19%	36.31	52.71	53.23 (k=25)	+46.60%
ResNet50 [45]	46.59	51.72	51.90 (k=15)	+11.41%	21.92	34.09	34.81 (k=25)	+58.82%
		Rank-Aggregation				**Rank-Aggregation**		
OSNet-AIN+OSNet-IBN	—	**72.42**	**72.42 (k=20)**	—	—	**58.55**	**59.51 (k=25)**	—
OSNet-AIN+ResNet50	—	67.34	67.34 (k=20)	—	—	52.19	52.88 (k=25)	—
OSNet-IBN+ResNet50	—	64.61	64.61 (k=20)	—	—	49.45	50.40 (k=25)	—
OSNet-AIN+OSNet-IBN+ResNet50	—	68.20	68.53 (k=15)	—	—	54.35	55.11 (k=25)	—
					Duke			
		Re-Ranking				**Re-Ranking**		
HACNN [56]	42.19	50.31	50.99 (k=25)	+20.85%	24.37	39.32	40.42 (k=25)	+65.88%
MLFN [57]	48.65	56.06	56.73 (k=25)	+16.61%	28.00	44.00	45.39 (k=25)	+62.13%
OSNet-AIN [50]	71.14	75.67	76.84 (k=25)	+8.01%	51.68	66.60	68.31 (k=30)	+32.19%
OSNet-IBN [50]	67.41	73.88	75.00 (k=25)	+11.25%	44.66	63.60	64.81 (k=25)	+45.12%
ResNet50 [45]	52.29	60.50	62.57 (k=30)	+19.66%	31.00	48.77	50.67 (k=25)	+63.45%
		Rank-Aggregation				**Rank-Aggregation**		
OSNet-AIN+OSNet-IBN	—	**76.21**	**77.69 (k=25)**	—	—	**67.46**	**69.21 (k=25)**	—
OSNet-AIN+ResNet50	—	72.80	74.55 (k=30)	—	—	63.71	65.50 (k=25)	—
OSNet-IBN+ResNet50	—	72.26	74.10 (k=30)	—	—	62.65	64.09 (k=25)	—
OSNet-AIN+OSNet-IBN+ResNet50	—	74.69	76.17 (k=25)	—	—	65.74	67.02 (k=30)	—

TABLE VI: Semi-supervised classification (**accuracy**) on **Flowers** dataset for different features. We compare the training that used the original features with the one that used embeddings generated by the proposed RFE. The best result for each classifier is highlighted in bold and the best for each dataset is highlighted in red.

Mode	Descriptor	SVM [58]	GCN-Net [59]	GCN-Gat [60]	GCN-SGC [61]	GCN-APPNP [62]	GCN-ARMA [63]
Original	**CNN-ResNet [45]**	82.467%	69.386%	71.211%	78.649%	72.186%	60.475%
	CNN-DPNet [51]	79.812%	72.954%	18.874%	76.292%	70.539%	56.539%
	CNN-SENet [52]	76.193%	68.895%	63.18%	72.835%	66.797%	60.649%
Our Embeddings	**CNN-ResNet [45]**	**82.565%**	**82.593%**	**82.966%**	**84.948%**	**83.974%**	**75.160%**
	CNN-DPNet [51]	80.131%	80.003%	41.237%	81.603%	81.029%	67.784%
	CNN-SENet [52]	76.716%	76.618%	73.454%	77.559%	77.260%	70.382%
Relative Gain	**CNN-ResNet [45]**	+0.12%	+19.03%	+16.51%	+8.01%	+16.33%	+24.28%
	CNN-DPNet [51]	+0.40%	+9.66%	+118.49%	+6.96%	+14.87%	+19.89%
	CNN-SENet [52]	+0.69%	+11.21%	+16.26%	+6.49%	+15.66%	+16.05%

TABLE VII: Semi-supervised classification (**accuracy**) on **Corel5k** dataset for different features. We compare the training that used the original features with the one that used embeddings generated by the proposed RFE. The best result for each classifier is highlighted in bold and the best for each dataset is highlighted in red.

Mode	Descriptor	SVM [58]	GCN-Net [59]	GCN-Gat [60]	GCN-SGC [61]	GCN-APPNP [62]	GCN-ARMA [63]
Original	**CNN-ResNet [45]**	89.504%	78.066%	87.68%	90.288%	86.679%	73.621%
	CNN-DPNet [51]	87.662%	84.733%	18.349%	87.389%	85.653%	72.883%
	CNN-SENet [52]	88.613%	88.627%	87.292%	90.404%	88.76%	83.447%
Our Embeddings	**CNN-ResNet [45]**	**89.602%**	90.008%	91.003%	91.54%	91.507%	89.212%
	CNN-DPNet [51]	87.933%	89.488%	52.374%	90.515%	91.061%	85.135%
	CNN-SENet [52]	88.776%	**91.299%**	**91.441%**	**91.97%**	**92.198%**	**90.924%**
Relative Gain	**CNN-ResNet [45]**	+0.11%	+15.3%	+3.79%	+1.39%	+5.57%	+21.18%
	CNN-DPNet [51]	+0.31%	+5.61%	+185.43%	+3.58%	+6.31%	+16.81%
	CNN-SENet [52]	+0.18%	+3.01%	+4.75%	+1.73%	+3.87%	+8.96%

are referred to as external or unseen queries. To assess the proposed approach in such cases, we conducted experiments on the Flowers, Corel5k, and ALOI datasets, which are presented in Table VIII.

We generated a set of unseen queries by randomly removing elements from the original dataset. To ensure a balanced analysis, we generated 10 samples per dataset, with each sample containing one element from each class. The reported MAP (both original and RFE) reflects the effectiveness of the approach in handling unseen queries, where the improvement is visible for all datasets and features.

TABLE VIII: Evaluation of RFE on **unseen queries** considering **MAP (%)**. The reported results are the average of 10 executions, each conducted on a different set of unseen queries randomly sampled from the dataset.

Dataset	Descriptor	Original	RFE
Flowers	CNN-ResNet	52.3226	65.4526
	VIT-B16	89.0063	93.3823
	SWIN-TF	93.0988	95.3603
Corel5k	CNN-ResNet	63.2227	76.3823
	VIT-B16	75.2124	84.8642
	SWIN-TF	72.3914	82.5962
ALOI	CNN-ResNet	82.5268	88.4239
	VIT-B16	80.1258	85.8109
	SWIN-TF	89.7562	93.1862

G. Comparison with State-of-the-art for Unsupervised Image Retrieval

This section aims to present the comparisons of the best results obtained by the proposed RFE (reported in Section IV-D) in relation to the most recent baselines and state-of-the-art approaches on unsupervised image retrieval.

Table IX presents the results for ORL and MPEG-7 datasets, which are two traditional benchmark datasets. These datasets are used for comparison with different diffusion methods. The ORL consists of images of faces, while the MPEG-7 is composed of images of shapes and contours. In order to keep consistency with the baselines, the same features were used for all the approaches: the IDSC [48] for MPEG-7 and raw images for ORL. The result with the original features is reported as "Our Baseline". The best values are highlighted in bold for each dataset. Notice that RFE achieved the best result for ORL and comparable ones for MPEG-7.

TABLE IX: **State-of-the-art (SOTA)** comparison with other variants of diffusion process on the **ORL (R@15)** and the **MPEG-7 (R@40)** datasets.

Methods	**ORL**	**MPEG-7**
Baseline [64]	62.35	85.40
SD [65]	71.67	83.09
LCDP [66]	74.25	89.45
TPG [67]	73.90	89.06
MR [68]	77.05	89.26
MR* [68]	77.58	92.61
GDP [69]	77.42	90.96
RDP (Y=I) [64]	78.53	**93.77**
RDP (Y=W) [64]	79.27	**93.78**
Our Baseline	74.32	85.40
RFE **(our method)**	**90.62** **(k=10)**	**93.54** **(k=20)**

The state-of-the-art comparison also encompasses the Flowers, Corel5k, and ALOI datasets; which is shown in Table X. Our method outperformed all other recent approaches, achieving the best results on all three datasets. The values reveal the effectiveness of RFE for both small and large datasets (Flowers and ALOI contain 13060 and 10200 images, respectively), with MAP always above 96.79%. This is a really significant result since the baselines also consider rank-aggregation of different features, especially Unsupervised Genetic Algorithm Framework for Rank Selection and fusion (UGAF-RSF) [70] and Unsupervised Selective Rank Fusion (USRF) [71] that combine more than 10 features.

Tables XI and XII compare the RFE results to state-of-the-art methods on Holidays and Ukbench datasets, respectively. These datasets are widely used as benchmarks for many retrieval algorithms. We compare RFE to at least 15 approaches

TABLE X: **State-of-the-art (SOTA)** comparison on **Flowers, Corel5k, and ALOI** datasets (**MAP %**).

Method	Flowers	Corel5k	ALOI
CPRR [33]	—	—	76.90
RL-Sim [72]	—	—	78.84
RL-Recom [73]	—	—	80.35
LHRR [14]	—	73.34	88.42
BFSTree [5]	—	53.00	91.15
RDPAC [16]	—	56.00	91.31
UGAF-RSF [70]	80.92	91.45	—
USRF [71]	81.71	90.32	—
RFE (Our Method)	**99.65**	**96.79**	**97.73**

TABLE XI: **State-of-the-art (SOTA)** comparison on **Holidays** dataset (**MAP**).

MAP for state-of-the-art methods				
Jégou *et al.* [34]	Tolias *et al.* [74]	Paulin *et al.* [75]	Qin *et al.* [76]	Zheng *et al.* [77]
75.07%	82.20%	82.90%	84.40%	85.20%
Sun *et al.* [78]	Zheng *et al.* [79]	Pedronette *et al.* [19]	Arandjelovic *et al.* [80]	Li *et al.* [81]
85.50%	85.80%	86.16%	87.50%	89.20%
Razavian *et al.* [82]	Pedronette *et al.* [5]	Gordo *et al.* [83]	Valem *et al.* [71]	Valem *et al.* [84]
89.60%	90.02%	90.30%	90.51%	90.51%
Liu *et al.* [81]	Pedronette *et al.* [14]	Pedronette *et al.* [16]	Yu *et al.* [85]	Berman *et al.* [86]
90.89%	90.94%	91.25%	91.40%	91.80%

RFE (Our Method)
91.97%

for each dataset. Notice, that the results achieved by RFE are higher than the baselines in both cases. We achieved a N-S Score of 3.97 (the maximum possible value is 4.00).

TABLE XII: **State-of-the-art (SOTA)** comparison on **UK-Bench** dataset (**NS-Score**).

N-S-Scores for state-of-the-art methods				
Qin *et al.* [17]	Zhang *et al.* [87]	Zheng *et al.* [88]	Bai *et al.* [89]	Xie *et al.* [90]
3.67	3.83	3.84	3.86	3.89
Lv *et al.* [91]	Liu *et al.* [92]	Pedronette *et al.* [19]	Bai *et al.* [93]	Liu *et al.* [94]
3.91	3.92	3.93	3.93	3.93
Valem *et al.* [84]	Bai *et al.* [95]	Valem *et al.* [71]	Valem *et al.* [70]	Chen *et al.* [96]
3.93	3.94	3.94	3.95	3.96

RFE (Our Method)
3.97

Table XIII presents the results of different approaches on the Re-ID datasets considering both R1 and MAP. Our results (RFE) are marked with a gray background and correspond to the best ones according to Table V. The abbreviations in parentheses indicate the datasets used for training (C03 = CUHK03, M = Market1501, D = DukeMTMC, MT = MSMT17). For example, the use of (D, M) indicates that the reported result corresponds to training done either on Duke or on Market dataset. The results reported on Market were trained on Duke and the results reported on Duke were trained on Market. None of the presented methods were trained using labels of the target dataset. The abbreviations were omitted for multi-source baselines, but they can be consulted in their papers. The best results for each dataset are highlighted in bold. Notice, that our results are among the best in all the cases and are above all of the baselines for DukeMTMC considering MAP.

TABLE XIII: **State-of-the-art (SOTA)** comparison for **person Re-ID** datasets considering **MAP** (%) and **R-01** (%). The abbreviations in parentheses indicate the datasets used for training (C03 = CUHK03, M = Market1501, D = DukeMTMC, MT = MSMT17). For example, the use of (D, M) indicates that the reported result corresponds to training done either on Duke or on Market dataset. The results reported on Market were trained on Duke and the results reported on Duke were trained on Market. None of the presented methods were trained using labels of the target dataset.

Method	**Year**	**Datasets**					
		Market1501		**DukeMTMC**		**CUHK03**	
		R1	**MAP**	**R1**	**MAP**	**R1**	**MAP**
Unsupervised Methods							
ARN [97]	2018	70.3	39.4	60.2	33.4	—	—
EANet [98]	2018	66.4	40.6	45.0	26.4	51.4	31.7
ECN [99]	2019	75.1	43.0	63.3	40.4	—	—
TAUDL [100]	2018	63.7	41.2	61.7	43.5	44.7	31.2
UTAL [101]	2019	69.2	46.2	62.3	44.6	**56.3**	**42.3**
SSL [102]	2020	71.7	37.8	52.5	28.6	—	—
HCT [103]	2020	80.0	56.4	69.6	50.7	—	—
CAP [104]	2021	**91.4**	**79.2**	**81.1**	67.3	—	—
IICS [105]	2021	89.5	72.9	80.0	64.4	—	—
Domain Adaptive Methods							
HHL (D,M) [106]	2018	62.2	31.4	46.9	27.2	—	—
HHL (C03) [106]	2018	56.8	29.8	42.7	23.4	—	—
ATNet (D,M) [107]	2019	55.7	25.6	45.1	24.9	—	—
CSGLP (D,M) [108]	2019	63.7	33.9	56.1	36.0	—	—
ISSDA (D,M) [109]	2019	81.3	63.1	72.8	54.1	—	—
ECN++ (D,M) [110]	2020	84.1	63.8	74.0	54.4	—	—
MMCL (D,M) [111]	2020	84.4	60.4	72.4	51.4	—	—
Cross-Domain Methods (single-source)							
EANet (C03) [98]	2018	59.4	33.3	39.3	22.0	—	—
EANet (D,M) [98]	2018	61.7	32.9	51.4	31.7	—	—
SPGAN (D,M) [112]	2018	43.1	17.0	33.1	16.7	—	—
DAAM (D,M) [113]	2019	42.3	17.5	29.3	14.5	—	—
AF3 (D,M) [114]	2019	67.2	36.3	56.8	37.4	—	—
AF3 (MT) [114]	2019	68.0	37.7	66.3	46.2	—	—
PAUL (MT) [115]	2019	68.5	40.1	72.0	53.2	—	—
Cross-Domain Methods (multi-source)							
EMTL [116]	2018	52.8	25.1	39.7	22.3	—	—
CAMEL [117]	2017	54.5	26.3	—	—	31.9	—
Baseline by [118]	2019	80.5	56.8	67.4	46.9	29.4	27.4
Our Proposed Method							
Our Method		**72.42**	**59.51**	**77.69**	**69.21**	**36.89**	**39.24**

H. Comparison with State-of-the-art for Semi-Supervised Image Classification

This section compares the semi-supervised image classification results reported in Section IV-E to various state-of-the-art approaches. Table XIV presents the comparisons considering different features (CNN-ResNet [45] and CNN-SENet [52]). The best result for each feature and dataset is highlighted in bold. The gray rows indicate the results that correspond to our method. We employed the same protocol adopted for RFE in all baselines: 5 executions of 10 folds. The only exception is CoMatch [119], where only 3 executions were reported for Corel5k due to the long time required to train this approach. Different from others, CoMatch takes images as input. However, it uses CNN-ResNet as its backbone.

For all the methods, we considered the default parameters and implementation provided by the original authors or the one in *Python Sklearn*. Regarding parameters, we used $k = 20$ for methods that require a size for the neighborhood set (i.e, kNN, GNN-LDS, GNN-KNN-LDS, and WSEF). The Label Spreading (LS) [122] was used combined with different classifiers once it can be used to generate pseudo-labels for further expanding the training set. The results achieved by RFE are the best ones for the SENet features and very comparable to the best for the ResNet features.

TABLE XIV: **Accuracy** comparison (%) for baselines on **Flowers** and **Corel5k** datasets. We compared our approach with **semi-supervised classification baselines**. The methods are compared with different input features. The results of our method are highlighted with a gray background; the best results for each pair of features and dataset are marked in bold.

Method	Input	Flowers	Corel5k
CoMatch [119]	**Images**	82.55	*85.70*
kNN	**ResNet Features**	63.67	76.80
SVM [58]		80.54	88.73
OPF [120]		71.77	83.56
SL-Perceptron		75.44	83.56
ML-Perceptron		78.88	87.10
PseudoLabel+SGD [121]		82.69	89.76
LS+kNN [122]		73.49	83.98
LS+SVM [58], [122]		73.53	83.26
LS+OPF [120], [122]		72.66	82.32
LS+SL-Perceptron [122]		72.34	82.38
LS+ML-Perceptron [122]		73.03	82.53
GNN-LDS [123]		54.98	62.69
GNN-KNN-LDS [123]		79.32	88.94
WSEF [124]		**85.12**	**91.68**
RFE (Our Method)		84.95	91.54
kNN	**SENet Features**	48.71	58.78
SVM [58]		73.30	85.89
OPF [120]		64.00	81.33
SL-Perceptron		71.84	82.28
ML-Perceptron		72.62	86.90
PseudoLabel+SGD [121]		76.87	89.85
LS+kNN [122]		58.05	72.16
LS+SVM [58], [122]		59.84	72.79
LS+OPF [120], [122]		59.25	72.20
LS+SL-Perceptron [122]		59.27	72.19
LS+ML-Perceptron [122]		59.39	72.24
GNN-LDS [123]		52.24	65.80
GNN-KNN-LDS [123]		73.69	89.95
WSEF [124]		76.16	89.74
RFE (Our Method)		**77.56**	**92.20**

I. Visual Analysis

In addition to the numerical analyses, qualitative experiments are also important for understanding the results achieved by the proposed approach. For better visualization of the improvements provided by RFE in the semi-supervised classification experiments, Figure 5 illustrates feature spaces on Flowers17 dataset with CNN-ResNet descriptor for three different cases: *(a)* features extracted by the CNN-ResNet descriptor; *(b)* GCN-Net output features after being trained on the CNN-ResNet features; and *(c)* GCN-Net output features after being trained on the CNN-ResNet features combined to the RFE embeddings. The TSNE method was used in order to compute the coordinates in the 2D space. While each dot represents a different element of the dataset, each combination of color and shape corresponds to a distinct class. Notice that *(c)* presents the best correspondence among the visual groups formed by the dots and the original dataset classes. This evinces our hypothesis that the RFE embeddings improve the classification of GCNs.

Experiments were also conducted to visualize the performance of RFE in retrieval tasks. Figure 6 presents examples of ranked lists before and after the execution of our proposed method. These results were obtained on different datasets (CNN-ResNet for Flowers and Corel5k; and OSNET-AIN for Duke) with the default parameters and k. The query images are presented with green borders and the incorrect ones with red borders. It clearly shows the significant improvements for all the queries.

V. Conclusion

In this work, we have proposed Rank Flow Embedding (RFE). The method is based on different techniques (hypergraph, Cartesian product, connected components) and can be used for improving both retrieval and classification tasks. An extensive experimental evaluation was conducted on 10 datasets, including 7 general purpose and 3 person re-identification datasets. The results are very promising for the vast majority of cases when compared to the state-of-the-art. In future work, we intend to investigate new strategies for graph modeling and embedding generation. We also intend to apply our method to other types of multimedia data.

References

[1] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influences, and trends of the new age," *ACM Computing Surveys*, vol. 40, no. 2, pp. 5:1–5:60, 2008.

[2] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. W. Fieguth, L. Liu, and M. S. Lew, "Deep image retrieval: A survey," *CoRR*, vol. abs/2101.11282, 2021.

[3] W. Zhou, H. Li, and Q. Tian, "Recent advance in content-based image retrieval: A literature survey," *CoRR*, vol. abs/1706.06064, 2017.

[4] Y. Zhao, L. Wang, L. Zhou, Y. Shi, and Y. Gao, "Modelling diffusion process by deep neural networks for image retrieval," in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*. BMVA Press, 2018, p. 161.

[5] D. C. G. Pedronette, L. P. Valem, and R. da S. Torres, "A bfs-tree of ranking references for unsupervised manifold learning," *Pattern Recognition*, vol. 111, p. 107666, 2021.

[6] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1213–1226, 2019.

[7] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, on-line, To appear.

[8] F. Yang, R. Hinami, Y. Matsui, S. Ly, and S. Satoh, "Efficient image retrieval via decoupling diffusion into online and offline processing," in *Conference on Artificial Intelligence, AAAI 2019*. AAAI Press, 2019, pp. 9087–9094.

[9] A. Iscen, Y. Avrithis, G. Tolias, T. Furon, and O. Chum, "Fast spectral ranking for similarity search," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7632–7641.

[10] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *CVPR*, 2013, pp. 1320–1327.

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

[12] D. C. G. Pedronette and R. da S. Torres, "Image re-ranking and rank aggregation based on similarity of ranked lists," *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.

[13] X. Yang, L. Prasad, and L. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE TPAMI*, vol. 35, no. 1, pp. 28–38, 2013.

[14] D. C. G. Pedronette, L. P. Valem, J. Almeida, and R. da S. Torres, "Multimedia retrieval through unsupervised hypergraph-based manifold ranking," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5824–5838, 2019.

[15] A. Iscen, G. Tolias, Y. Avrithis, T. Furon, and O. Chum, "Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations," in *CVPR*, 2017.

[16] D. C. Guimarães Pedronette, L. Pascotti Valem, and L. J. Latecki, "Efficient rank-based diffusion process with assured convergence," *Journal of Imaging*, vol. 7, no. 3, 2021.

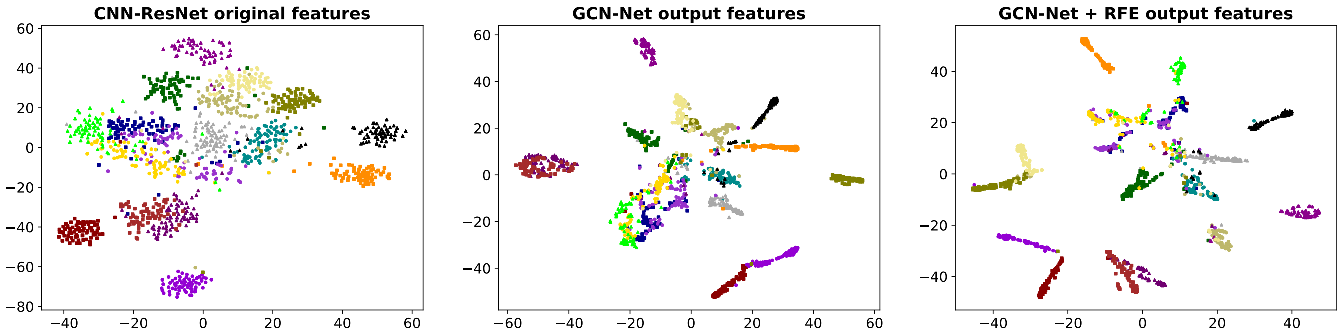
[17] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *CVPR*, june 2011, pp. 777 –784.

[18] X. Bai, S. Bai, and X. Wang, "Beyond diffusion process: Neighbor set similarity for fast re-ranking," *Information Sciences*, vol. 325, pp. 342 – 354, 2015.

[19] D. C. G. Pedronette, F. M. F. Gonçalves, and I. R. Guilherme, "Unsupervised manifold learning through reciprocal kNN graph and Connected Components for image retrieval tasks," *Pattern Recognition*, vol. 75, pp. 161 – 174, 2018.

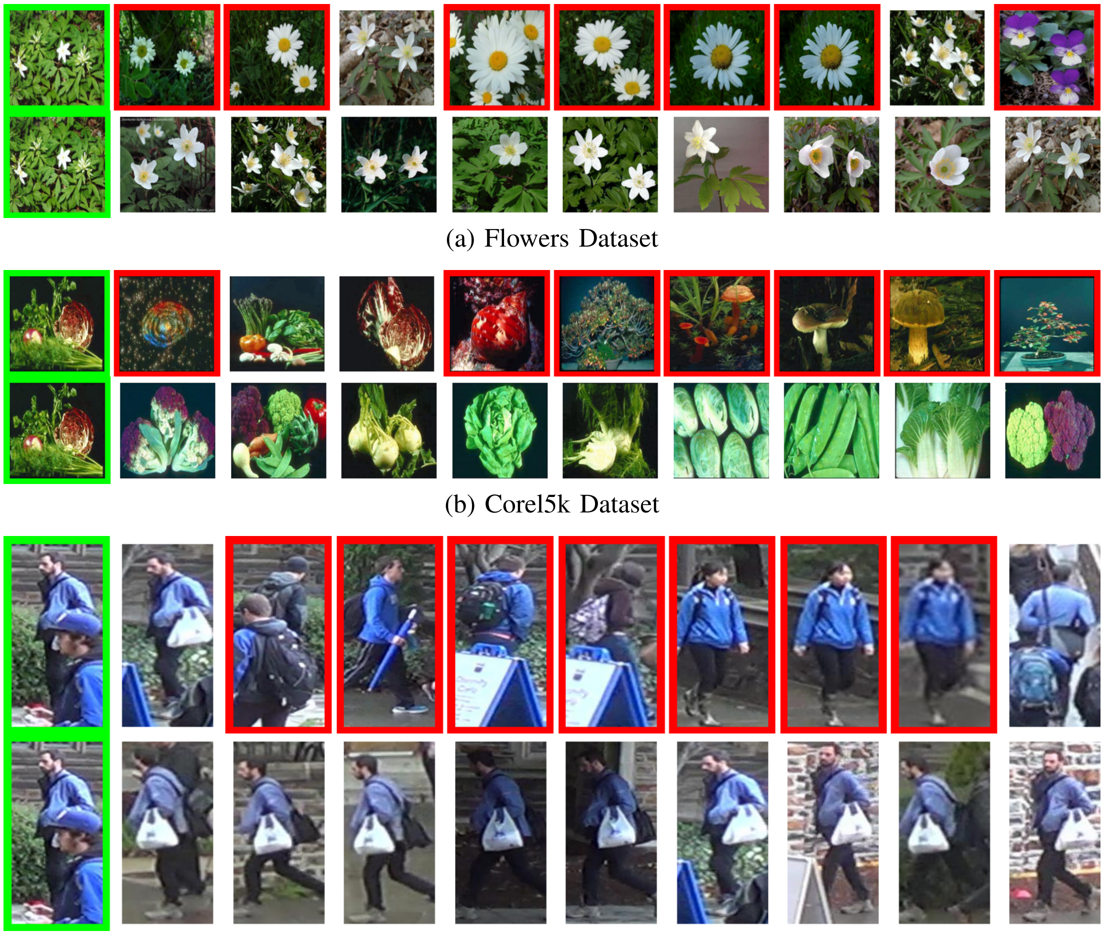
[20] D. C. G. Pedronette, J. Almeida, and R. da S. Torres, "A scalable re-ranking method for content-based image retrieval," *Information Sciences*, vol. 265, no. 1, pp. 91–104, 2014.

[21] A. Delvinioti, H. Jégou, L. Amsaleg, and M. E. Houle, "Image retrieval with reciprocal and shared nearest neighbors," in *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, vol. 2, Jan 2014, pp. 321–328.



(a) **CNN-ResNet** (b) **GCN-Net** (c) **GCN-Net + RFE**

Fig. 5: Feature space illustrations computed by TSNE on Flower dataset with CNN-ResNet descriptor. It shows the (a) original feature space, (b) feature space obtained with the GCN, and (c) feature space obtained by the GCN using the RFE (our proposed approach) embeddings.



(a) Flowers Dataset

(b) Corel5k Dataset

(c) Duke Re-ID Dataset

Fig. 6: Examples of ranked lists before and after RFE was applied for three datasets. Query images are highlighted with green borders and wrong results are with red borders.

[22] D. C. G. Pedronette, O. A. Penatti, and R. da S. Torres, "Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks," *Image and Vision Computing*, vol. 32, no. 2, pp. 120 – 130, 2014.

[23] D. C. G. Pedronette and R. da S. Torres, "Image re-ranking and rank aggregation based on similarity of ranked lists," *Pattern Recognition*, vol. 46, no. 8, pp. 2350–2360, 2013.

[24] L. P. Valem, C. R. D. Oliveira, D. C. G. Pedronette, and J. Almeida, "Unsupervised similarity learning through rank correlation and knn sets," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 14, no. 4, pp. 80:1–80:23, 2018.

[25] J. Wang, Y. Li, X. Bai, Y. Zhang, C. Wang, and N. Tang, "Learning context-sensitive similarity by shortest path propagation," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2367–2374, 2011.

[26] D. C. G. a. Pedronette and R. d. S. Torres, "A correlation graph approach for unsupervised manifold learning in image retrieval tasks," *Neurocomputing*, vol. 208, no. Supplement C, pp. 66 – 79, 2016, sI: BridgingSemantic.

[27] A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Mining on manifolds: Metric learning without labels," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7642–7651.

[28] S. Karanam, M. Gou, Z. Wu, A. Rates-Borras, O. Camps, and

R. J. Radke, "A systematic evaluation and benchmark for person re-identification: Features, metrics, and datasets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 523–536, March 2019.

[29] M. Wang and T. Song, "Remote sensing image retrieval by scene semantic matching," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 2874–2886, May 2013.

[30] M. Agarwal and J. Mostafa, "Content-based image retrieval for alzheimer's disease detection," in *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, June 2011, pp. 13–18.

[31] X. SHEN, Y. Xiao, S. X. Hu, O. Sbai, and M. Aubry, "Re-ranking for image retrieval and transductive few-shot classification," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, 2021, pp. 25 932–25 943.

[32] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *IEEE Conference on Conference on Computer Vision and Pattern Recognition (CVPR'10)*, June 2010, pp. 3376–3383.

[33] L. P. Valem, D. C. G. Pedronette, and J. Almeida, "Unsupervised similarity learning through cartesian product of ranking references," *Pattern Recognition Letters*, vol. 114, pp. 41 – 52, 2018.

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

[35] Z. Zhong, L. Zheng, D. Cao, and S. Li, "Re-ranking person re-identification with k-reciprocal encoding," in *CVPR*, 2017.

[36] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, "Scalable person re-identification: A benchmark," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1116–1124.

[37] Z. Zheng, L. Zheng, and Y. Yang, "Unlabeled samples generated by gan improve the person re-identification baseline in vitro," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, p. 3754–3762.

[38] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, p. 1457–1469, Dec. 2004.

[39] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1447–1454.

[40] L. J. Latecki, R. Lakamper, and U. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *CVPR*, 2000, pp. 424–429.

[41] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," *Pattern Recognition*, vol. 46, no. 1, pp. 188 – 198, 2013.

[42] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'2006)*, vol. 2, 2006, pp. 2161–2168.

[43] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 152–159.

[44] J.-M. Geusebroek, G. J. Burghouts, and A. W. M. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

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

[46] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *CoRR*, vol. abs/2103.14030, 2021.

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[48] H. Ling and D. W. Jacobs, "Shape classification using the inner-distance," *IEEE TPAMI*, vol. 29, no. 2, pp. 286–299, 2007.

[49] D. C. G. Pedronette and R. da S. Torres, "Shape retrieval using contour features and distance optmization," in *VISAPP*, vol. 1, 2010, pp. 197 – 202.

[50] K. Zhou, Y. Yang, A. Cavallaro, and T. Xiang, "Learning generalisable omni-scale representations for person re-identification," *arXiv preprint arXiv:1910.06827*, 2019.

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

[52] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[53] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1800–1807.

[54] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token vit: Training vision transformers from scratch on imagenet," *arXiv preprint arXiv:2101.11986*, 2021.

[55] K. Reddy Mopuri and R. Venkatesh Babu, "Object level deep feature pooling for compact image representation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2015.

[56] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[57] X. Chang, T. M. Hospedales, and T. Xiang, "Multi-level factorisation net for person re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[58] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[59] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[60] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[61] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *ICML*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6861–6871.

[62] J. Klicpera, A. Bojchevski, and S. Günnemann, "Combining neural networks with personalized pagerank for classification on graphs," in *International Conference on Learning Representations*, 2019.

[63] F. M. Bianchi, D. Grattarola, L. Livi, and C. Alippi, "Graph neural networks with convolutional arma filters," *IEEE TPAMI*, pp. 1–1, 2021.

[64] S. Bai, X. Bai, Q. Tian, and L. J. Latecki, "Regularized diffusion process on bidirectional context for object retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 5, pp. 1213–1226, 2019.

[65] B. Wang and Z. Tu, "Affinity learning via self-diffusion for image segmentation and clustering," in *2012 IEEE CVPR*, 2012, pp. 2312–2319.

[66] X. Yang, S. Koknar-Tezel, and L. J. Latecki, "Locally constrained diffusion process on locally densified distance spaces with applications to shape retrieval," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 357–364.

[67] X. Yang, L. Prasad, and L. J. Latecki, "Affinity learning with diffusion on tensor product graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 28–38, 2013.

[68] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *NeurIPS*, ser. NIPS'03. Cambridge, MA, USA: MIT Press, 2003, p. 169–176.

[69] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1320–1327.

[70] L. P. Valem and D. C. G. a. Pedronette, "An unsupervised genetic algorithm framework for rank selection and fusion on image retrieval," in *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ser. ICMR '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 58–62.

[71] L. P. Valem and D. C. G. Pedronette, "Unsupervised selective rank fusion for image retrieval tasks," *Neurocomputing*, vol. 377, pp. 182 – 199, 2020.

[72] D. C. Guimarães Pedronette, J. Almeida, and R. da S. Torres, "A scalable re-ranking method for content-based image retrieval," *Information Sciences*, vol. 265, pp. 91–104, 2014.

[73] L. P. Valem, D. C. G. a. Pedronette, R. d. S. Torres, E. Borin, and J. Almeida, "Effective, efficient, and scalable unsupervised distance learning in image retrieval tasks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ser. ICMR '15. New York, NY, USA: ACM, 2015, pp. 51–58.

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

[75] M. Paulin, J. Mairal, M. Douze, Z. Harchaoui, F. Perronnin, and C. Schmid, "Convolutional patch representations for image retrieval: An unsupervised approach," *Int. Journal of Computer Vision*, 2017.

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

[77] L. Zheng, S. Wang, and Q. Tian, "Coupled binary embedding for large-scale image retrieval," *IEEE Transactions on Image Processing (TIP)*, vol. 23, no. 8, pp. 3368–3380, 2014.

[78] S. Sun, Y. Li, W. Zhou, Q. Tian, and H. Li, "Local residual similarity for image re-ranking," *Information Sciences*, vol. 417, no. Sup. C, pp. 143 – 153, 2017.

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

[80] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 5297–5307.

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

[82] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," *ITE Transactions on Media Technology and Applications*, vol. 4, no. 3, pp. 251–258, 2016.

[83] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, 2017.

[84] L. P. Valem and D. C. G. Pedronette, "Graph-based selective rank fusion for unsupervised image retrieval," *Pattern Recognition Letters*, vol. 135, pp. 82–89, 2020.

[85] W. Yu, K. Yang, H. Yao, X. Sun, and P. Xu, "Exploiting the complementary strengths of multi-layer cnn features for image retrieval," *Neurocomputing*, vol. 237, pp. 235–241, 2017.

[86] M. Berman, H. Jégou, V. Andrea, I. Kokkinos, and M. Douze, "MultiGrain: a unified image embedding for classes and instances," *arXiv e-prints*, Feb 2019.

[87] S. Zhang, M. Yang, T. Cour, K. Yu, and D. Metaxas, "Query specific rank fusion for image retrieval," *IEEE TPAMI*, vol. 37, no. 4, pp. 803–815, April 2015.

[88] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian, "Query-adaptive late fusion for image search and person re-identification," in *CVPR*, 2015.

[89] S. Bai and X. Bai, "Sparse contextual activation for efficient visual re-ranking," *IEEE Trans. on Image Processing (TIP)*, vol. 25, no. 3, pp. 1056–1069, 2016.

[90] L. Xie, R. Hong, B. Zhang, and Q. Tian, "Image classification and retrieval are one," in *ACM ICMR'2015*, 2015, pp. 3–10.

[91] Y. Lv, W. Zhou, Q. Tian, S. Sun, and H. Li, "Retrieval oriented deep feature learning with complementary supervision mining," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4945–4957, 2018.

[92] Z. Liu, S. Wang, L. Zheng, and Q. Tian, "Robust imagegraph: Rank-level feature fusion for image search," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3128–3141, 2017.

[93] S. Bai, Z. Zhou, J. Wang, X. Bai, L. J. Latecki, and Q. Tian, "Ensemble diffusion for retrieval," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 774–783.

[94] G. Lao, S. Liu, C. Tan, Y. Wang, G. Li, L. Xu, L. Feng, and F. Wang, "Three degree binary graph and shortest edge clustering for re-ranking in multi-feature image retrieval," *Journal of Visual Communication and Image Representation*, vol. 80, p. 103282, 2021.

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

[96] X. Chen and Y. Li, "Deep feature learning with manifold embedding for robust image retrieval," *Algorithms*, vol. 13, no. 12, 2020.

[97] Y.-J. Li, F.-E. Yang, Y.-C. Liu, Y.-Y. Yeh, X. Du, and Y.-C. Frank Wang, "Adaptation and re-identification network: An unsupervised deep transfer learning approach to person re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[98] H. Huang, W. Yang, X. Chen, X. Zhao, K. Huang, J. Lin, G. Huang, and D. Du, "Eanet: Enhancing alignment for cross-domain person re-identification," *arXiv preprint arXiv:1812.11369*, 2018.

[99] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Invariance matters: Exemplar memory for domain adaptive person re-identification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[100] M. Li, X. Zhu, and S. Gong, "Unsupervised person re-identification by deep learning tracklet association," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 737–753.

[101] ——, "Unsupervised tracklet person re-identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[102] Y. Lin, L. Xie, Y. Wu, C. Yan, and Q. Tian, "Unsupervised person re-identification via softened similarity learning," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3387–3396.

[103] K. Zeng, M. Ning, Y. Wang, and Y. Guo, "Hierarchical clustering with hard-batch triplet loss for person re-identification," in *CVPR*, 2020, pp. 13 654–13 662.

[104] M. Wang, B. Lai, J. Huang, X. Gong, and X.-S. Hua, "Camera-aware proxies for unsupervised person re-identification," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[105] S. Xuan and S. Zhang, "Intra-inter camera similarity for unsupervised person re-identification," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 921–11 930.

[106] Z. Zhong, L. Zheng, S. Li, and Y. Yang, "Generalizing a person retrieval model hetero- and homogeneously," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[107] J. Liu, Z.-J. Zha, D. Chen, R. Hong, and M. Wang, "Adaptive transfer network for cross-domain person re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[108] C. Ren, B. Liang, and Z. Lei, "Domain adaptive person re-identification via camera style generation and label propagation," *CoRR*, vol. abs/1905.05382, 2019.

[109] H. Tang, Y. Zhao, and H. Lu, "Unsupervised person re-identification with iterative self-supervised domain adaptation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[110] Z. Zhong, L. Zheng, Z. Luo, S. Li, and Y. Yang, "Learning to adapt invariance in memory for person re-identification," *TPAMI*, pp. 1–1, 2020.

[111] D. Wang and S. Zhang, "Unsupervised person re-identification via multi-label classification," in *CVPR*, 2020, pp. 10 978–10 987.

[112] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, "Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification," in *CVPR*, 2018.

[113] Y. Huang, P. Peng, Y. Jin, J. Xing, C. Lang, and S. Feng, "Domain adaptive attention model for unsupervised cross-domain person re-identification," *CoRR*, vol. abs/1905.10529, 2019.

[114] H. Liu, J. Cheng, S. Wang, and W. Wang, "Attention: A big surprise for cross-domain person re-identification," *CoRR*, vol. abs/1905.12830, 2019.

[115] Q. Yang, H.-X. Yu, A. Wu, and W.-S. Zheng, "Patch-based discriminative feature learning for unsupervised person re-identification," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[116] Y. Xian and H. Hu, "Enhanced multi-dataset transfer learning method for unsupervised person re-identification using co-training strategy," *IET Computer Vision*, vol. 12, no. 8, pp. 1219–1227, 2018.

[117] H.-X. Yu, A. Wu, and W.-S. Zheng, "Cross-view asymmetric metric learning for unsupervised person re-identification," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[118] D. Kumar, P. Siva, P. Marchwica, and A. Wong, "Fairest of them all: Establishing a strong baseline for cross-domain person reid," *CoRR*, vol. abs/1907.12016, 2019.

[119] J. Li, C. Xiong, and S. C. H. Hoi, "Comatch: Semi-supervised learning with contrastive graph regularization," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9455–9464.

[120] W. P. Amorim, A. X. Falcão, and M. H. d. Carvalho, "Semi-supervised pattern classification using optimum-path forest," in *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, Aug 2014, pp. 111–118.

[121] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on challenges in representation learning, ICML*, vol. 3, no. 2, 2013.

[122] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information Processing Systems 16*. MIT Press, 2004, pp. 321–328.

[123] L. Franceschi, M. Niepert, M. Pontil, and X. He, "Learning discrete structures for graph neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[124] J. G. C. Presotto, L. P. Valem, N. G. de Sá, D. C. G. Pedronette, and J. P. Papa, "Weakly supervised learning through rank-based contextual measures," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5752–5759.