

Information Ranking Using Optimum-Path Forest

Nathalia Q. Ascenção* Luis C. S. Afonso†, Danilo Colombo‡, Luciano Oliveira§ and João P. Papa*

*UNESP - Univ. Estadual Paulista, School of Sciences, Bauru, Brazil

Email: {nathalia.ascencao,joao.papa}@unesp.br

†UFSCar - Federal University of São Carlos, Department of Computing, São Carlos, Brazil

Email: sugi.luis@ufscar.br

‡Cenpes, Petróleo Brasileiro S.A.,

Email: colombo.danilo@petrobras.com.br

§UFBA - Federal University of Bahia, Salvador, Brazil

Email: lrebouca@ufba.br

Abstract—The task of learning to rank has been widely studied by the machine learning community, mainly due to its use and great importance in information retrieval, data mining, and natural language processing. Therefore, ranking accurately and learning to rank are crucial tasks. Context-Based Information Retrieval systems have been of great importance to reduce the effort of finding relevant data. Such systems have evolved by using machine learning techniques to improve their results, but they are mainly dependent on user feedback. Although information retrieval has been addressed in different works along with classifiers based on Optimum-Path Forest (OPF), these have so far not been applied to the learning to rank task. Therefore, the main contribution of this work is to evaluate classifiers based on Optimum-Path Forest, in such a context. Experiments were performed considering the image retrieval and ranking scenarios, and the performance of OPF-based approaches was compared to the well-known SVM-Rank pairwise technique and a baseline based on distance calculation. The experiments showed competitive results concerning precision and outperformed traditional techniques in terms of computational load.

I. INTRODUCTION

Information Retrieval stands for a field of knowledge that aims to return relevant data given an input query, which can be a multimedia content such as an image, audio, video, or a text-based information [1]. Nowadays, the amount of data that has been generated has increased considerably. E-mails and multimedia data are interchanged among millions of users daily, thus contributing to spreading communication and increasing the workload in Internet traffic. Therefore, it is highly desired to handle such amount of data efficiently, i.e., to store and further retrieve the relevant information only.

Images have a crucial role in several fields of research, such as medicine, advertising, education, and entertainment, among others [2]. Content-based Image Retrieval (CBIR) systems come to reduce the costs of manually retrieving relevant data since it is a laborious and very much time-consuming task. CBIR-driven systems aim at retrieving relevant images from a dataset based on features such as color, shape, and texture, and have been assisting in a broad range of applications [3]–[7].

The authors are grateful to Petrobras grant #2014/00545-0, FAPESP grants #2013/07375-0, #2014/12236-1, #2017/25908-6, #2018/15597-6, #2019/07665-4, as well as CNPq grants #307066/2017-7, #307550/2018-4, and #427968/2018-6.

However, CBIR systems face the problem of being pretty much user-dependent, which means that it is not straightforward to learn models that can generalize well for every kind of input query. Such techniques make use of relevance feedback from the user to overcome such dependence. The user indicates the images that are relevant (and non-relevant) to their needs, and the process is repeated until the retrieved data is satisfactory [8]–[11]. Hence, ranking accurately is crucial for CBIR systems.

In this sense, machine learning techniques have been applied in the context of learning to rank. Younus et al. [12] used image features such as color histogram, color moment, co-occurrence matrices, and wavelet moment for measuring the similarity among samples. Then, they applied k -means with the Particle Swarm Optimization algorithm for retrieving images. Irtaza et al. [13] proposed a method that uses an in-depth texture analysis, a learning scheme based on k -nearest neighbors and a neural network to the retrieval task. A Bayesian network based on feedback relevance was applied in [14] on medical image retrieval using feature analysis such as color, texture, and shape jointly with some visual descriptors.

In [15], an artificial neural network along with Support Vector Machine (SVM) was employed on face image retrieval and recognition. Support Vector Machine also has been applied in content-based image retrieving [16]–[24] to reduce the semantic gap problem by enhancing the image classification process; as an active classifier combining classification with active learning based on feedback relevance; together with visual descriptors to improve retrieval performance, and also in deep learning as proposed in [25], where a framework based on Convolutional Neural Networks and SVM was used on feature extraction, classification, and image retrieval.

A few years ago, the Optimum-Path Forest (OPF) framework was proposed to handle the problem of pattern classification as a graph partitioning task. The framework comprises supervised [26]–[28], semi-supervised [29], and unsupervised versions [30]. Such approaches work by mapping the classification problem as a graph partition task, where the nodes stand for the samples that are represented by their corresponding feature vectors and are connected through some adjacency

relation that has been defined previously. In a nutshell, nodes are classified based on a competitive process among key samples that try to conquer others offering them optimum-path costs. The key samples, also called prototypes, are the ones that best represent different classes and are chosen based on a specific heuristic.

Although OPF has been used in several areas, only a few works have applied such a technique to the context of image retrieval. Tavares et al. [8] proposed two approaches for content-based image retrieval based on OPF, where the main idea is to ask the user to mark some relevant images which are further used as prototypes for a new training step. Later on, the remaining data is classified and sorted in such a way that contains only the relevant images, which are presented to the user once more. This process is repeated until the user is satisfied.

Dhawale and Joglekar [31] compared OPF against other classifiers for image retrieval purposes, and concluded that it could be much faster than widely used techniques such as Artificial Neural Networks, Support Vector Machines, and the k -Nearest Neighbours (k -NN) classifier. However, those works are based on user feedback only. Therefore, the main contribution of our work is to introduce the OPF-Ranking (OPF-R), which is an OPF-based approach that can rank images automatically, i.e., without user intervention. The OPF-R was evaluated under three well-known image datasets where there were considered two relevance metrics. The performance was compared against a baseline technique and the SVM-Rank, in which OPF-R showed competitive results and lower ranking computing times.

The remainder of this paper is organized as follows. Section II reviews the theoretical background concerning OPF-based classifiers. The proposed approach and methodology are presented in Sections III and IV, respectively. Section V discusses the experiments and results. Finally, Section VI states conclusions and future works.

II. OPTIMUM-PATH FOREST

The Optimum-Path Forest is a framework for the design of graph-based classifiers. Let \mathcal{Z} be a labeled dataset, such that $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$, in which \mathcal{Z}_1 and \mathcal{Z}_2 are training and testing sets, respectively. OPF encodes each sample $\mathbf{u} \in \mathcal{Z}$ as a graph node, and the graph is initially designed based on a predefined adjacency relation \mathcal{A} with edges weighted by the distance between the feature vectors of their connecting nodes.

The general OPF training algorithm is divided into two parts: (i) to find a set of samples called prototypes, and (ii) to compute optimum-path trees (OPTs) rooted at them. The prototypes are the most representative samples from each class. The definition of *most representative sample*, as well as the method for computing prototypes, are different for each variant of OPF, and they are explained in the following sections.

Let \mathcal{P} be the set of prototypes such that $\mathcal{P} \subset \mathcal{Z}_1$. The OPTs are built through a competitive process in which samples try to “conquer” each other by offering costs. The competition starts

at the prototypes that offer their best cost to the remaining training samples (i.e., non-prototype samples). The costs are defined by a path-cost function, which is also different for each OPF variant.

Let $\mathbf{p} \in \mathcal{P}$ and $\mathbf{s} \in \mathcal{Z}_1 \setminus \mathcal{P}$ be some prototype and non-prototype samples, respectively. Suppose that sample \mathbf{s} is conquered by a sample \mathbf{p} that offers to it the best cost. Upon such an assumption, \mathbf{p} assigns its label to \mathbf{s} , and \mathbf{s} is added to tree rooted at \mathbf{p} . Notice that prototypes cannot be conquered, and a class is represented by at least one tree. The outcome of the training step is a set of optimum-path trees (i.e., optimum-path forest) $\mathcal{G}_{tr} = (\mathcal{Z}_1, \mathcal{A})$.

Concerning the classification step, a node $\mathbf{v} \in \mathcal{Z}_2$ is connected to \mathcal{G}_{tr} according to \mathcal{A} (i.e., if \mathcal{A} is a k -NN adjacency relation, then \mathbf{v} is first connected to the k -nearest samples from \mathcal{G}_{tr}). Besides, the sample $\mathbf{u} \in \mathcal{Z}_1$ that offers the best cost to \mathbf{v} assigns its label to it. This work makes use of two variants of the supervised version whose working mechanisms are further explained.

A. OPF with Complete Graph (CG-OPF)

This supervised variant was introduced by Papa et al. [26] and implements the complete graph as the adjacency relation (i.e., all nodes are connected). As aforementioned, the first step in the training phase is to find the set of prototypes \mathcal{P} , which is computed through the minimum spanning tree (MST) over \mathcal{G}_{tr} . The prototype nodes are the samples belonging to the intersection region among classes since they are more likely to be misclassified.

The following step is the competition process, which is carried out using Equations 1 and 2 that stand for the initialization and propagation of costs, respectively:

$$f_{\max}(\langle \mathbf{p} \rangle) = \begin{cases} 0 & \text{if } \mathbf{p} \in \mathcal{P}, \\ +\infty & \text{otherwise} \end{cases} \quad (1)$$

and

$$f_{\max}(\pi_{\mathbf{p}} \cdot \langle \mathbf{p}, \mathbf{v} \rangle) = \max\{f_{\max}(\pi_{\mathbf{p}}), d(\mathbf{p}, \mathbf{v})\}, \quad (2)$$

in which f is a real-valued path-cost function, $\pi_{\mathbf{p}} \cdot \langle \mathbf{p}, \mathbf{v} \rangle$ stands for the concatenation of path $\pi_{\mathbf{p}}$ (i.e., a sequence of adjacent nodes starting from any node and with terminus at node \mathbf{p}) with an edge $\langle \mathbf{p}, \mathbf{v} \rangle$, and d denotes a distance function.

The conquering of samples happens during the propagation of costs where prototypes offer their optimum-path costs to other samples. The sample \mathbf{v} is conquered by sample \mathbf{p} that minimizes (2). The classification is based on the connecting force between samples from \mathcal{Z}_1 and testing samples from \mathcal{Z}_2 , and works similarly to the training phase. Each $\mathbf{t} \in \mathcal{Z}_2$ is connected to \mathcal{G}_{tr} obeying \mathcal{A} . Then, it is evaluated the sample $\mathbf{v}^* \in \mathcal{Z}_1$ that satisfies Equation 3:

$$C(\mathbf{t}) = \arg \min_{\mathbf{v} \in \mathcal{Z}_1} \max\{C(\mathbf{v}), d(\mathbf{v}, \mathbf{t})\}, \quad (3)$$

where C is the cost of the sample. Figure 1 depicts the training and classification processes.

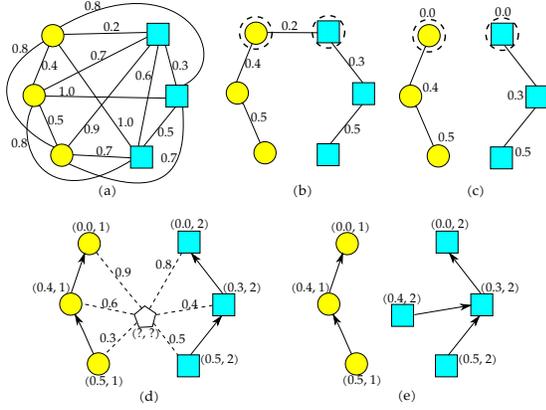


Fig. 1. OPF with complete graph. Training phase: (a) a two-class training graph with weighted arcs, (b) an MST with prototypes highlighted, and (c) optimum-path forest generated during the training phase with costs over the nodes (notice the prototypes have zero cost). Classification phase: (d) sample pentagon is connected to all training nodes, and (e) pentagon is conquered by a sample from the class “square”, and it receives the “square” label.

B. OPF with k -nearest neighbors Graph (k -NN-OPF)

The OPF with k -nearest neighbors Graph was proposed by Papa et al. [32]–[34], whose main differences to CG-OPF are the adjacency relation, the weighting of nodes, and how prototypes are computed. The k -NN-OPF employs the k -nearest neighbors as the adjacency relation, and the nodes are now weighted by a probability density function. The method for building the set of prototypes must be changed since a k -NN adjacency relation does not guarantee a connected graph. Instead, the set \mathcal{P} is computed based on region density values where samples of higher density are selected as prototypes. Such approach is similar to selecting the centroids of clusters [35]. Hence, k -NN-OPF is understood as a “dual version” of CG-OPF (minimization problem) since it aims at maximizing the cost of each sample according to Equation 4:

$$\max f(\pi_{\mathbf{u}}), \forall \mathbf{u} \in \mathcal{Z}_1. \quad (4)$$

Besides, samples $\mathbf{u} \in \mathcal{Z}_1$ are weighted by a function $\rho(\mathbf{u})$ that computes the probability density value as follows:

$$\rho(\mathbf{u}) = \frac{1}{\sqrt{2\pi\sigma^2k}} \sum_{\mathbf{v} \in \mathcal{A}_k(\mathbf{u})} \exp\left(\frac{-d(\mathbf{u}, \mathbf{v})}{2\sigma^2}\right), \quad (5)$$

where $\mathcal{A}_k(\mathbf{u})$ stands for the k -nearest neighbors of sample \mathbf{u} , $d_{\max} = \max\{d(\mathbf{u}, \mathbf{v}) \in G_{tr}\}$, and $\sigma = d_{\max}/3$. After computing the density values for all training samples, k -NN-OPF starts the competition process through the path-cost function f_{\min} defined as follows:

$$\begin{aligned} f_{\min}(\langle \mathbf{v} \rangle) &= \begin{cases} \rho(\mathbf{v}) & \text{if } \mathbf{v} \in \mathcal{P} \\ \rho(\mathbf{v}) - 1 & \text{otherwise} \end{cases} \\ f_{\min}(\pi_{\mathbf{u}} \cdot \langle \mathbf{u}, \mathbf{v} \rangle) &= \min\{f_{\min}(\pi_{\mathbf{u}}), \rho(\mathbf{v})\}. \end{aligned} \quad (6)$$

The upper formulation stands for the proper initialization of the training nodes. The term $\rho(\mathbf{v}) - 1$ must be used to avoid over clustering since a plateau of densities may occur.

The competition itself stands for the propagation of costs among samples. Since the prototype holds the higher cost of its optimum-path tree, the idea is to conquer samples with lower costs. Finally, the sample that maximizes f_{\min} for a given sample \mathbf{v} will be the one to conquer it.

The classification of samples in \mathcal{Z}_2 is performed similarly to the conquering process. The first step computes the k -nearest neighbors from \mathcal{Z}_1 to a testing sample $t \in \mathcal{Z}_2$. Finally, it is verified which node $\mathbf{s}^* \in \mathcal{Z}_1$ satisfies the equation below:

$$C(\mathbf{t}) = \arg \max_{\mathbf{v} \in \mathcal{Z}_1} \min\{C(\mathbf{v}), \rho(\mathbf{t})\}. \quad (7)$$

Figure 2 depicts the classification process.

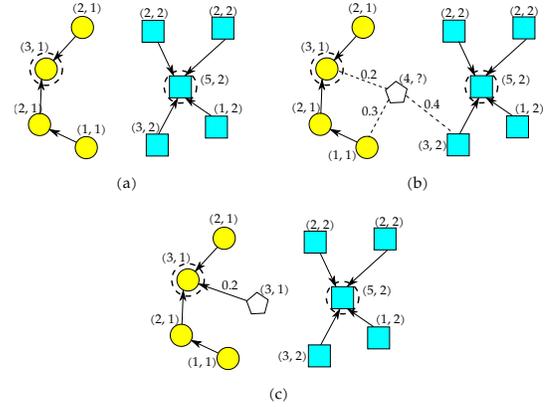


Fig. 2. OPF with k -NN graph classification: (a) optimum-path forest generated during the training phase with the pair (cost, label) over the nodes, (b) sample pentagon is connected to its k -nearest training nodes, and (c) pentagon is conquered by a sample from the class “circle”, and it receives the “circle” label.

III. PROPOSED APPROACH

The idea of ranking in CBIR is to retrieve the most similar images for a given query. However, OPF does not rely on any similarity metric solely. Instead, the ranking problem is designed as a connectivity problem, where a testing sample is assigned to a class based on the strength of its connection to each sample in \mathcal{G}_{tr} , where the path cost defines the strength level (i.e., the better the path-cost, the stronger is the strength). Therefore, (OPF-R) maps the similarity as the strength of the query to the samples from the trained model (i.e., the stronger is the connection between a query and a model sample, the higher is their similarity).

After training OPF-R, a given query q is further connected to \mathcal{G}_{tr} according to the adjacency relation \mathcal{A} , i.e., if \mathcal{A} is the complete graph type, then q is connected to all samples in \mathcal{A} ; if \mathcal{A} is the k -NN graph type, then q is connected to the k -nearest samples in \mathcal{A} . Notice that the training is performed as described in Section II.

The next step is to perform the competition process, similarly to the classification process. The difference is that both CG-OPF and k -NN-OPF store all path-costs offered to q . In the end, the path-costs are sorted, and the r best costs comprise the ranking list. Figure 3 depicts the ranking approach through OPF-R with a complete graph (CG-OPF-R) considering the Top-5 results. One interesting point is that OPF ranks samples natively, i.e., there is no need to perform complex changes in the original classifier to support both ranking and retrieval.

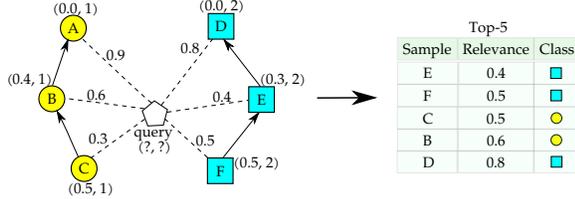


Fig. 3. Ranking by OPF. The query sample is connected to the training set according and costs are further computed. Samples of lower cost are the most strongly connected ones.

IV. METHODOLOGY

This section presents the data and techniques employed to validate the proposed approach.

A. Datasets

The proposed approach was evaluated in three datasets with detailed information presented in Table I. Notice that Brodatz dataset has originally a total of 112 images, and was later expanded to 1,792 images by dividing the original ones into 16 parts. This strategy was applied to increase the number of samples per class.

TABLE I
DATASET INFORMATION.

dataset	type	# images	# classes	images per classes
Brodatz	texture	1,792	112	16
Caltech101	objects/scenes	8,677	101	40 – 800
MPEG-7	shape	1,500	70	20

B. Features

The experiments considered a total of seven features. However, they are not applied to all datasets since their nature is not suitable for all cases. For instance, Brodatz is a dataset characterized by texture images. Therefore, only Local Binary Pattern (LBP) [36] and Statistical Analysis of Structural Information (SASI) [37] features are computed. Table II presents the seven extracted features.

C. Experimental Setup

The experiments were carried out using three configurations of training and testing set (querying set) sizes: (i) 25% samples for training and 75% samples for classification, (ii) 50% samples for training and 50% samples for classification, and (iii) 75% samples for training and 25% samples for

TABLE II
FEATURE INFORMATION.

feature	type	dataset
Auto-color correlation (ACC) [38]	color	Caltech101
Border/Interior Pixel Classification (BIC) [39]	color	Caltech101
Color Coherence Vectors (CCV) [40]	color	Caltech101
Local Color Histogram (LCH) [41]	color	Caltech101
Local Binary Pattern (LBP) [36]	texture	Brodatz
Statistical Analysis of Structural Information (SASI) [37]	texture	Brodatz
Spherical Pyramid-Technique (SPYTEC) [42]	shape	MPEG-7

classification. The sets were randomly generated. Table III presents the parameter values used in this work¹.

TABLE III
PARAMETERS.

technique	parameters
CG-OPF	–
k -NN-OPF	$k_{\max} = 20$
SVM-Rank	$c = 0.05, l = 0, e = 0.001$

The remaining parameters of SVM-Rank were set in their default values as defined in [43].

D. Relevance Metric

The accuracy is measured by two metrics as described:

- Normalized Discounted Cumulative Gain (NDCG): The NDCG computes the cumulative gain (i.e., the sum of the relevance score of the candidate samples) considering the ordering in the ranking. The metric is based on the Discounted Cumulative Gain (DCG) that reduces the score of relevant data in a logarithmical proportional to their position in the ranking, Equation 8. The lower the position, the higher is the penalization:

$$DCG_r = \sum_{i=1}^r \frac{2^{rel_i} - 1}{\log_2(i + 1)}, \quad (8)$$

where rel_i is the graded relevance of the result at position i . Thus, NDCG is defined as follows:

$$NDCG_r = \frac{DCG_r}{DCG_{r_{ideal}}}, \quad (9)$$

where $DCG_{r_{ideal}}$ stands for the case where the data is sorted by their relevance score.

- Mean Average Precision (MAP): The MAP is commonly used to evaluate ranking methods applied to binary score problems and computes the mean average precision for each query as follows:

$$MAP = \frac{1}{n} \sum_{q=1}^n AP_q, \quad (10)$$

where n is the number of queries and AP is the average precision of a single query and defined as:

$$AP = \frac{\sum_{r=1}^n (P@r \times I(rel_r == 1))}{\sum_{r=1}^n I(rel_r == 1)}, \quad (11)$$

¹These parameters were obtained empirically.

being $P@r$ the precision:

$$P@r = \frac{1}{r} \sum_{i=1}^r (rel_i == 1), \quad (12)$$

which is the normalized number of relevant candidates in the first r positions.

Both metrics use the absolute relevance (i.e., a label of 0 or 1 is assigned to the candidate based on its relevance). The relevance of a candidate is defined by comparing its label to the query after the ranking. The candidate is assigned a relevance 0 if it does not have the same label as the query, or 1 if the candidate and query share the same label.

V. EXPERIMENTAL RESULTS

The experimental results are organized by datasets with tables presenting the NDGC and MAP values for each scenario (i.e., dataset \times training/testing sets configuration \times descriptor). Besides evaluating OPF-Rank in a great variety of cases, we also compared its performance against a distance-based technique (Distance) and the well-known SVM-Rank considering the top-10, top-15, and top-20 rankings. Notice that the Distance technique computes the ranking based on the distance function suggested by the author of the descriptor (i.e., each descriptor has its most appropriate distance function). The best results (i.e., for each configuration, descriptor and top-r) are shown in bold, according to the Wilcoxon signed-rank test with the significance of 0.05. Besides, a hold-out approach with 10 runs with randomly generated training and testing sets was applied for validation purposes.

A. Brodatz

The experimental results are presented in Table IV. One can observe that the best results achieved by CG-OPF and k -NN-OPF were using the SASI descriptor regardless of the training/testing sets configuration. Concerning the comparison between the OPF-based approaches, CG-OPF obtained a better performance over k -NN-OPF in the $25\% \times 75\%$ configuration, whereas the performance was very close in the remaining configurations. Among all techniques, SVM-Rank presented the best overall results.

B. Caltech101

The results are presented in Table V, in which LCH and BIC provided the best results. CG-OPF showed better relevance values over k -NN-OPF in all considered configurations. Except for Distance, all techniques are benefited by increasing the number of training samples. The best overall results were achieved by SVM-Rank.

C. MPEG-7

The results are presented in Table VI. Once again, CG-OPF outperformed k -NN-OPF in all configurations. An interesting behavior is that changing the configuration $25\% \times 75\%$ to $50\% \times 50\%$ provided a very small increase in the results, whereas the configuration $75\% \times 25\%$ caused a drop in the results. In this dataset, CG-OPF showed competitive results when compared to SVM-Rank.

TABLE IV
RESULTS CONCERNING BRODATZ DATASET.

		25% \times 75%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	LBP	0.370	0.225	0.377	0.182	0.382	0.152
	SASI	0.392	0.260	0.393	0.170	0.396	0.140
CG-OPF	LBP	0.354	0.162	0.371	0.124	0.375	0.102
	SASI	0.405	0.199	0.410	0.154	0.411	0.127
k -NN-OPF	LBP	0.328	0.135	0.345	0.106	0.349	0.087
	SASI	0.378	0.189	0.381	0.146	0.383	0.120
SVM-Rank	LBP	0.362	0.145	0.383	0.115	0.388	0.094
	SASI	0.431	0.328	0.434	0.264	0.436	0.225
		50% \times 50%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	LBP	0.368	0.202	0.378	0.164	0.386	0.140
	SASI	0.406	0.289	0.411	0.236	0.411	0.201
CG-OPF	LBP	0.391	0.214	0.402	0.174	0.410	0.148
	SASI	0.428	0.222	0.431	0.176	0.432	0.148
k -NN-OPF	LBP	0.361	0.197	0.370	0.160	0.378	0.137
	SASI	0.432	0.308	0.435	0.251	0.437	0.212
SVM-Rank	LBP	0.409	0.224	0.420	0.182	0.429	0.155
	SASI	0.451	0.321	0.455	0.262	0.457	0.222
		75% \times 25%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	LBP	0.341	0.145	0.359	0.118	0.364	0.099
	SASI	0.406	0.289	0.411	0.236	0.411	0.201
CG-OPF	LBP	0.410	0.255	0.418	0.210	0.423	0.179
	SASI	0.402	0.209	0.405	0.166	0.407	0.139
k -NN-OPF	LBP	0.378	0.235	0.385	0.194	0.389	0.165
	SASI	0.403	0.334	0.403	0.274	0.406	0.236
SVM-Rank	LBP	0.429	0.267	0.437	0.220	0.442	0.187
	SASI	0.457	0.379	0.457	0.311	0.460	0.268

D. Discussion

In this section, we present a discussion concerning the results obtained in the experiments. Besides, we also provided an additional study concerning the computational load as well. Tables VII, VIII, and IX present the results concerning Brodatz, Caltech101, and MPEG-7 datasets, respectively. The results stand for the average (seconds) ranking time over 10 runs. As one can observe, OPF-based approaches are pretty much faster than Distance and SVM-Rank techniques (e.g., 1.8 to 2.2 times faster). However, SVM-Rank achieved the best results in most cases but with up 4% of superiority over OPF-based techniques.

If one takes into account the trade-off between ranking relevance and retrieving time, OPF-based approaches figure as the most prominent ones, since they achieved results close to the SVM-Rank, but faster. Another point that should be highlighted is that the proposed approach was little modified to handle ranking problems; meanwhile, SVM-Rank needed a considerable adaptation in its working mechanism. In other others, we expect to achieve better results with a more in-depth change in OPF-based competition process to better adapt to ranking-driven applications.

Also, we are not taking into account the training time, which is supposed to be even faster concerning OPF-based approaches. Particular attention is given to CG-OPF, which

TABLE V
RESULTS CONCERNING CALTECH101 DATASET.

		25% × 75%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	ACC	0.252	0.151	0.270	0.143	0.283	0.136
	BIC	0.264	0.162	0.281	0.155	0.294	0.149
	CCV	0.242	0.140	0.263	0.132	0.277	0.126
CG-OPF	LCH	0.269	0.162	0.289	0.151	0.302	0.142
	ACC	0.250	0.141	0.274	0.132	0.286	0.125
	BIC	0.259	0.151	0.284	0.142	0.267	0.135
k -NN-OPF	CCV	0.238	0.125	0.260	0.118	0.276	0.112
	LCH	0.264	0.156	0.282	0.144	0.299	0.138
	ACC	0.232	0.132	0.254	0.124	0.266	0.116
SVM-Rank	BIC	0.241	0.142	0.259	0.133	0.273	0.126
	CCV	0.222	0.118	0.242	0.111	0.257	0.105
	LCH	0.246	0.146	0.262	0.136	0.277	0.128
	ACC	0.260	0.146	0.258	0.137	0.298	0.129
	BIC	0.270	0.157	0.263	0.147	0.306	0.140
	CCV	0.248	0.130	0.244	0.122	0.288	0.116
LCH	0.275	0.174	0.294	0.164	0.311	0.156	

		50% × 50%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	ACC	0.283	0.182	0.301	0.174	0.314	0.167
	BIC	0.295	0.186	0.312	0.186	0.325	0.180
	CCV	0.273	0.171	0.294	0.163	0.309	0.157
CG-OPF	LCH	0.300	0.193	0.320	0.182	0.333	0.173
	ACC	0.281	0.172	0.305	0.163	0.317	0.157
	BIC	0.290	0.182	0.310	0.173	0.325	0.175
k -NN-OPF	CCV	0.269	0.156	0.291	0.149	0.307	0.143
	LCH	0.295	0.187	0.313	0.175	0.330	0.172
	ACC	0.263	0.163	0.285	0.155	0.297	0.147
SVM-Rank	BIC	0.272	0.173	0.290	0.164	0.304	0.157
	CCV	0.253	0.149	0.273	0.142	0.287	0.137
	LCH	0.277	0.177	0.293	0.167	0.310	0.159
	ACC	0.291	0.177	0.316	0.169	0.329	0.160
	BIC	0.301	0.190	0.321	0.180	0.337	0.171
	CCV	0.264	0.146	0.287	0.138	0.304	0.132
LCH	0.306	0.205	0.325	0.195	0.342	0.187	

		75% × 25%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	ACC	0.340	0.239	0.358	0.231	0.371	0.224
	BIC	0.352	0.241	0.369	0.243	0.382	0.237
	CCV	0.330	0.228	0.351	0.221	0.366	0.214
CG-OPF	LCH	0.357	0.250	0.377	0.239	0.390	0.230
	ACC	0.338	0.229	0.362	0.220	0.374	0.213
	BIC	0.347	0.239	0.367	0.230	0.382	0.223
k -NN-OPF	CCV	0.326	0.213	0.348	0.206	0.364	0.201
	LCH	0.352	0.248	0.372	0.232	0.387	0.230
	ACC	0.321	0.220	0.342	0.212	0.354	0.204
SVM-Rank	BIC	0.329	0.230	0.347	0.221	0.361	0.216
	CCV	0.310	0.206	0.330	0.199	0.338	0.193
	LCH	0.336	0.235	0.351	0.224	0.367	0.216
	ACC	0.348	0.234	0.373	0.226	0.386	0.217
	BIC	0.358	0.247	0.378	0.237	0.396	0.228
	CCV	0.336	0.218	0.359	0.210	0.376	0.205
LCH	0.363	0.262	0.382	0.252	0.399	0.253	

does not comprise any parameter beforehand, thus turning out to be easier to be set up and with no need for a fine-tuning step. Additionally, although k -NN-OPF figures one parameter, its training time is faster than SVM-Rank, in which the number of parameters depends on the kernel function used.

Based on the above assumptions, we conclude that OPF-based classifiers are suitable for ranking purposes, even with its native version. We expect that better results may come with more in-depth modifications that shall affect little the efficiency of the methods.

TABLE VI
RESULTS CONCERNING MPEG-7 DATASET.

		25% × 75%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	SPYTEC	0.061	0.030	0.071	0.027	0.074	0.026
		0.076	0.078	0.071	0.030	0.088	0.029
		0.066	0.031	0.072	0.028	0.082	0.027
		0.075	0.034	0.082	0.031	0.092	0.031

		50% × 50%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	SPYTEC	0.087	0.051	0.093	0.051	0.108	0.052
		0.094	0.054	0.098	0.053	0.113	0.055
		0.086	0.050	0.093	0.050	0.107	0.052
		0.095	0.055	0.101	0.056	0.118	0.057

		75% × 25%					
technique	descriptor	top-r					
		10		15		20	
		NDGC	MAP	NDGC	MAP	NDGC	MAP
Distance	SPYTEC	0.134	0.097	0.140	0.095	0.147	0.095
		0.130	0.103	0.141	0.101	0.145	0.102
		0.126	0.099	0.136	0.098	0.138	0.098
		0.137	0.103	0.145	0.102	0.150	0.103

TABLE VII
COMPUTATIONAL LOAD [S] CONCERNING BRODATZ DATASET.

		top-r								
technique	descriptor	10		15		20				
		25x75	50x50	75x25	25x75	50x50	75x25	25x75	50x50	75x25
Distance	LBP	31.01	33.01	32.14	32.00	35.00	32.37	33.06	36.83	32.95
	SASI	31.49	34.65	33.33	33.26	35.39	34.03	34.51	37.41	34.11
CG-OPF	LBP	19.15	18.23	16.22	20.50	19.45	16.47	21.15	19.90	17.33
	SASI	16.74	18.85	16.64	17.90	19.95	16.75	18.20	20.20	17.95
k -NN-OPF	LBP	17.29	19.34	17.24	18.40	20.45	17.57	18.73	20.74	18.32
	SASI	18.46	20.45	18.37	19.61	21.61	18.55	20.12	22.06	19.18
SVM-Rank	LBP	32.35	34.35	34.50	33.36	35.36	35.01	34.05	35.97	35.21
	SASI	33.12	35.12	36.19	34.49	36.50	36.49	34.87	36.83	37.11

VI. CONCLUSIONS

In this work, we introduced two OPF variants to the context of content-based image retrieval and ranking. Both approaches, i.e., CG-OPF and k -NN-OPF, achieved promising results when compared to SVM-Rank, but being faster for ranking purposes. Although the latter one figured as the most accurate technique in almost all simulations, the best trade-off between effectiveness and efficiency was achieved by OPF.

As future works, we intend to change the OPF working mechanism and adapt some parts to handle better the problem of image ranking, as well as to consider different distance functions for the arc-weights.

REFERENCES

- [1] 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.
- [2] S. M. Youssef, "Ictedct-cbir: Integrating curvelet transform with enhanced dominant colors extraction and texture analysis for efficient content-based image retrieval," *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1358 – 1376, 2012, special issue on Recent Advances in Security and Privacy in Distributed Communications and Image processing.
- [3] M. Iliadis, S. Yoo, X. Xin, and A. K. Katsaggelos, "Virtual touring: A content based image retrieval application," in *2013 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, 2013, pp. 1–4.

TABLE VIII
COMPUTATIONAL LOAD [S] CONCERNING CALTECH101 DATASET.

technique	descriptor	top-r								
		10			15			20		
		25x75	50x50	75x25	25x75	50x50	75x25	25x75	50x50	75x25
Distance	ACC	40.98	40.16	39.25	41.72	40.95	40.09	42.22	41.38	40.49
	BIC	40.66	40.29	39.35	41.06	41.48	40.71	41.62	41.81	40.91
	CCV	41.42	39.16	37.66	41.74	39.48	37.98	42.19	39.69	38.09
CG-OPF	LCH	39.46	38.07	36.87	40.53	38.10	36.60	40.64	38.84	37.38
	ACC	21.03	21.71	20.66	21.86	22.12	20.98	22.55	22.52	21.44
	BIC	22.31	22.60	21.77	23.03	23.06	21.98	23.40	23.54	22.43
k-NN-OPF	CCV	21.30	21.71	20.77	22.06	22.29	21.32	22.73	23.09	22.20
	LCH	22.55	22.82	21.76	14.09	23.10	21.97	23.38	23.46	22.32
	ACC	22.62	22.69	21.55	13.96	23.12	21.97	14.26	23.49	22.35
SVM-Rank	BIC	23.45	24.05	22.21	23.95	24.38	22.55	24.39	24.79	22.98
	CCV	22.27	23.52	21.66	14.10	24.11	22.35	23.71	24.51	22.76
	LCH	23.13	25.29	23.32	23.25	25.86	23.88	24.13	26.14	24.21
SPYTEC	ACC	41.86	43.08	41.55	42.392	43.192	42.495	43.012	44.672	43.297
	BIC	42.39	41.87	40.32	42.94	42.47	40.97	43.41	43.03	41.44
	CCV	43.50	43.18	41.89	43.93	43.61	42.35	44.27	43.98	42.77
LCH	42.61	43.93	41.44	43.64	45.06	42.77	44.30	45.37	43.09	

TABLE IX
COMPUTATIONAL LOAD [S] CONCERNING MPEG-7 DATASET.

technique	descriptor	top-r								
		10			15			20		
		25x75	50x50	75x25	25x75	50x50	75x25	25x75	50x50	75x25
SPYTEC	Distance	28.18	28.17	29.66	29.35	29.58	30.86	29.75	29.80	31.12
	CG-OPF	18.35	18.11	19.35	20.23	20.27	21.76	21.55	21.72	22.61
	k-NN-OPF	20.11	20.34	21.32	21.24	21.24	21.95	22.00	22.50	24.17
SVM-Rank	32.78	33.13	34.66	33.55	33.81	34.63	34.12	34.59	35.46	

[4] M. Milovanovic, M. Minovic, and D. Starcevic, "Walking in colors: Human gait recognition using kinect and cbir," *IEEE MultiMedia*, vol. 20, pp. 28–36, 2013.

[5] P. H. Bugatti, D. S. Kaster, M. Ponciano-Silva, C. Traina, P. M. A. Marques, and A. J. Traina, "Prosper: Perceptual similarity queries in medical cbir systems through user profiles," *Computers in Biology and Medicine*, vol. 45, pp. 8 – 19, 2014.

[6] M. H. Memon, J. P. Li, I. Memon, R. A. Shaikh, and F. A. Mangi, "Efficient object identification and multiple regions of interest using cbir based on relative locations and matching regions," in *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, 2015, pp. 247–250.

[7] B. Demir and L. Bruzzone, "A novel active learning method in relevance feedback for content-based remote sensing image retrieval," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 2323–2334, 2015.

[8] A. d. S. Tavares, A. X. Falcão, and L. P. Magalhães, "Active learning paradigms for CBIR systems based on optimum-path forest classification," *Pattern Recognition*, vol. 44, no. 12, pp. 2971 – 2978, 2011.

[9] J. Pardede and B. Sitohang, "Reduce semantic gap in content-based image retrieval." American Scientific Publishers, 2017, pp. 10664–10671.

[10] B. J. Rani, B. Prashanth, and B. Krishna, "Active learning methods for interactive image retrieval," *International Journal of Innovative Computer Science & Engineering*, vol. 4, pp. 12–17, 2017.

[11] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: A comprehensive study," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 20 – 54, 2015.

[12] Z. S. Younus, D. Mohamad, T. Saba, M. H. Alkawaz, A. Rehman, M. Al-Rodhaan, and A. Al-Dhelaan, "Content-based image retrieval using pso and k-means clustering algorithm," *Arabian Journal of Geosciences*, vol. 8, pp. 6211–6224, 2015.

[13] A. Irtaza, M. A. Jaffar, E. Aleisa, and T.-S. Choi, "Embedding neural networks for semantic association in content based image retrieval," *Multimedia Tools and Applications*, vol. 72, pp. 1911–1931, 2014.

[14] B. Satish and K. P. Supreethi, "Content based medical image retrieval using relevance feedback bayesian network," in *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2017, pp. 424–430.

[15] N. S. Devi and K. Hemachandran, "Content based feature combination method for face image retrieval using neural network and svm classifier for face recognition," *Indian Journal of Science and Technology*, vol. 10, 2017.

[16] P. Pavani and T. S. Prabha, "Content based image retrieval using machine learning approach," in *Proceedings of the International Conference on*

Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013. Springer International Publishing, 2014, pp. 173–179.

[17] X.-Y. Wang, B.-B. Zhang, and H.-Y. Yang, "Active svm-based relevance feedback using multiple classifiers ensemble and features reweighting," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 368 – 381, 2013.

[18] K. Sugamya, S. Pabboju, and A. V. Babu, "A cbir classification using support vector machines," in *2016 International Conference on Advances in Human Machine Interaction (HMI)*, 2016, pp. 1–6.

[19] L. Hu, S. Lu, and X. Wang, "A new and informative active learning approach for support vector machine," *Information Sciences*, vol. 244, pp. 142 – 160, 2013.

[20] X.-Y. Wang, H.-Y. Yang, Y.-W. Li, W.-Y. Li, and J.-W. Chen, "A new svm-based active feedback scheme for image retrieval," *Engineering Applications of Artificial Intelligence*, vol. 37, pp. 43 – 53, 2015.

[21] S. Saad, H. Kumar, and T. K. Tewari, "Efficient content based image retrieval using svm and color histogram," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, 2017, pp. 1–3.

[22] N. Seth and S. Jindal, "Local binary pattern with support vector machine to enhance image retrieval," *International Journal of Advanced Research in Computer Science*, vol. 8, 2017.

[23] Y. M. Lohite and S. J. Pawar, "A novel method for content based image retrieval using local features and svm classifier," *International Research Journal of Engineering and Technology*, vol. 4, 2017.

[24] Y. Rao, W. Liu, B. Fan, J. Song, and Y. Yang, "A novel relevance feedback method for cbir," *World Wide Web*, 2018.

[25] O. Mohamed, E. A. Khalid, O. Mohammed, and A. Brahim, "Content-based image retrieval using convolutional neural networks," in *Lecture Notes in Real-Time Intelligent Systems*. Springer International Publishing, 2018, pp. 463–476.

[26] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 120–131, 2009.

[27] J. P. Papa, A. X. Falcão, V. H. C. Albuquerque, and J. a. M. R. S. Tavares, "Efficient supervised optimum-path forest classification for large datasets," *Pattern Recognition*, vol. 45, no. 1, pp. 512–520, 2012.

[28] J. P. Papa, S. E. N. Fernandes, and A. X. Falcão, "Optimum-path forest based on k-connectivity: Theory and applications," *Pattern Recognition Letters*, vol. 87, pp. 117–126, 2017.

[29] W. P. Amorim, A. X. Falcão, J. P. Papa, and M. H. Carvalho, "Improving semi-supervised learning through optimum connectivity," *Pattern Recognition*, vol. 60, pp. 72–85, 2016.

[30] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, no. 2, pp. 50–68, 2009.

[31] S. Dhawale and B. Joglekar, "Optimum path forest approach for image retrieval based on context," 2015.

[32] J. P. Papa and A. X. Falcão, "A learning algorithm for the optimum-path forest classifier," in *Graph-Based Representations in Pattern Recognition, 7th IAPR-TC-15 International Workshop, GBRPR 2009, Venice, Italy, May 26-28, 2009. Proceedings*, 2009, pp. 195–204.

[33] —, "A new variant of the optimum-path forest classifier," in *Advances in Visual Computing, 4th International Symposium, ISVC 2008, Las Vegas, NV, USA, December 1-3, 2008. Proceedings, Part I*, 2008, pp. 935–944.

[34] J. P. Papa, S. E. N. Fernandes, and A. X. Falcão, "Optimum-path forest based on k-connectivity: Theory and applications," *Pattern Recognition Letters*, vol. 87, pp. 117–126, 2017.

[35] G. H. Rosa, K. A. P. Costa, L. A. P. Júnior, J. P. Papa, A. X. Falcão, and J. M. R. S. Tavares, "On the training of artificial neural networks with radial basis function using optimum-path forest clustering," in *22nd International Conference on Pattern Recognition*, 2014, pp. 1472–1477.

[36] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, p. 971–987, Jul. 2002.

[37] A. Çarkacıoglu and F. Yarman Vural, "Sasi: A generic texture descriptor for image retrieval," *Pattern Recognition*, vol. 36, pp. 2615–2633, 2003.

[38] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlograms," 1997, pp. 762–768.

[39] R. O. Stehling, M. A. Nascimento, and A. X. Falcão, "A compact and efficient image retrieval approach based on border/interior pixel classification." ACM, 2002, pp. 102–109.

- [40] G. Pass, R. Zabih, and J. Miller, "Comparing images using color coherence vectors," in *Proceedings of the Fourth ACM International Conference on Multimedia*. Association for Computing Machinery, 1997, p. 65–73.
- [41] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, pp. 11–32, 1991.
- [42] T. Lu and C. Chang, "Color image retrieval technique based on color features and image bitmap," *Inf. Processing and Management*, vol. 43, p. 461–472, 2007.
- [43] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 217–226.