Adaptive Content Based Image Retrieval based on RICE Algorithm Selection model

Safa Hamreras and Bachir Boucheham
Department of Computer Sciences
University 20 Août 1955
Skikda, Algeria
SafaHamreras@gmail.com
Bachir.boucheham@hotmail.com

***Abstract*—In this paper, we propose a framework for "Algorithm Selection" for image retrieval by content (CBIR). The framework is based on the model of RICE and is adapted to satisfy a given query depending on its characteristics by choosing the best classical *CBIR-Algorithm* from an *Algorithm-Portfolio*. As many as six algorithms for content based image retrieval have been included in the framework as alternatives for the different queries, including the training step. These algorithms range from RGB color moments, RGB color histogram to local binary pattern (LBP), etc. Therefore, there has been put an effort in the framework to cover the basic characteristics of images: Color and texture. Also, the framework integrates two color models to better enhance the *Algorithm-Query* adaptation process. Experimentations on the Wang (Corel 1k) database show the effectiveness of the proposed framework. Indeed, enhancements of more than 4% in precision have been obtained.**

Keywords—Adaptive framework, Content Based Image Retrieval, Algorithm Selection problem, feature selection.

I. Introduction

There exists two approaches in image retrieval: TBIR (Text Based Image Retrieval) and CBIR (Content Based Image Retrieval). The first one represents images by text annotations. This approach has many flaws: First, choosing the adequate annotation for a given image might be a difficult task, especially when this image contains a big quantity of information. Second, it is a manual task, an annotator is charged by giving images the suitable annotation which takes a lot of time and effort. Finally, the choice of image description is subjective, the same image may have multiple annotations depending on the person describing it [1]. One of the most interesting tentatives to overcome TBIR flaws was "Content Based Image Retrieval". Unlike text based image retrieval, this approach uses low level features to describe and retrieve images such as color, texture and shape.

Basic CBIR systems follow two phases: Offline indexation and query satisfaction. The first phase consists in extracting low level features to characterize an image. The search step includes feature extraction of the query image followed by the measure of similarity between the query image and all database images. Similarity is calculated using a distance function, and there exist many: Euclidian distance, Manhattan distance, Chebyshev distance, etc. After that, the CBIR system extracts the relevant images according to obtained similarity measures, the smaller the distance, the more similar images are. In other words, this step consists in "searching the k images whose feature vectors are most similar to the feature vector of the query image, namely k-nearest neighbor (k-NN) Searching" [2].

Advanced CBIR applications use many techniques and approaches to achieve high quality retrievals. These techniques include: Features selection, relevance feedback, Support Vector Machines (SVM), etc. In particular, features selection is one of the most interesting approaches in image retrieval.

Knowing that there is no feature that outperforms all others in the general case [3] [4], if we consider that one specific feature can be used as a unique feature in an image matching algorithm, then the feature selection problem can be regarded as an algorithm selection problem. However, a CBIR-Algorithm needs in fact at least two ingredients: Feature and color space. In this work we present a novel framework for feature selection based on RICE model for algorithm selection [5]. Our framework selects the best (Feature, Color space) to apply on a given query based on training data.

The rest of this paper is organized in 3 sections: First, we will expose the algorithm selection problem and its related concepts in section II, then we will explain the functioning of our framework in section III, and last we will report the obtained results and discuss them in section IV.

II. The Algorithm Selection Problem

The algorithm selection problem was defined by John RICE in 1976 [5]. It consists in choosing the best algorithm to apply to solve a specific problem instance. "The best algorithm" means here: The algorithm that has the best performance compared to all other algorithms, given a problem instance [6]. Researchers have recognized that there is no algorithm that has the best overall performance, but choosing the best algorithm to apply on each problem instance can lead to improve the average performance [7]. Fig. 1 illustrates RICE model for algorithm selection as presented in [5]:

978-1-5386-4690-8/18/$31.00 ©2018 IEEE

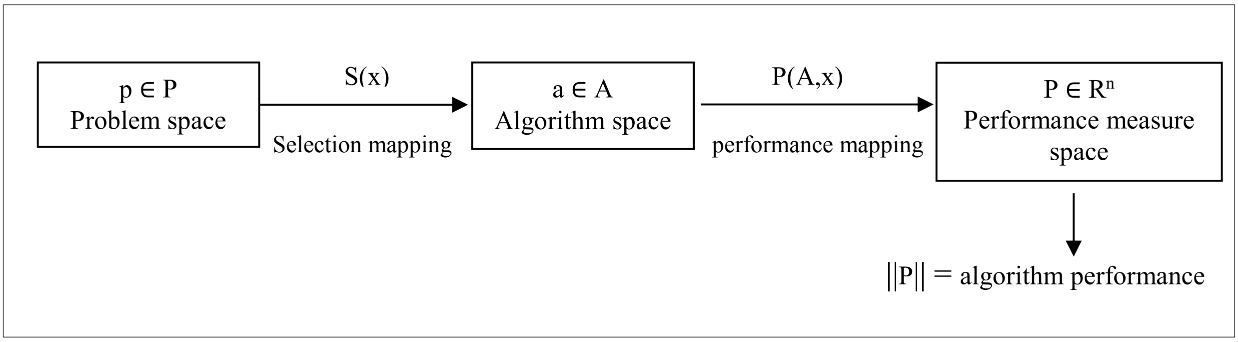

Fig. 1. RICE Model for Algorithm Selection as presented in [5]

- Problem space (P): Contains all problem instances. In our case it consists of the used database of images.
- Algorithm space (A): Contains algorithms applicable to all problem instances. Our framework's algorithm space consists of features to be applied on database images.
- Performance measure space (R^n): Contains the measures of performances when applying each algorithm on each problem instance. This set is the result of a training phase. The performance of candidate features is measured by precision.
- Selection mapping ((S(x)): Determines what algorithm to apply on what problem instance.
- Performance mapping (P(A,x)): Connects each algorithm applied on a specific problem to its performance measure, which means connecting the feature applied on a given image to its precision.

A. The no free lunch theorem

The no free lunch theorem is the essence of the algorithm selection problem. This theorem demonstrates that there is no algorithm that has the best overall performance: If an algorithm A outperforms another algorithm B given a subset of problems, then there must exist another subset of problems for which the algorithm B outperforms the algorithm A [6].

B. Algorithm portfolio

One of the fundamental concepts related to algorithm selection problem is "Algorithm Portfolio". It was first used by Huberman, Lukose and Hogg (1997) [8] to solve graph coloring problem [7]. Instead of using a single algorithm, this approach uses a set of algorithms to solve a set of problem instances. In addition to solver algorithms there exist a special algorithm called the "selector", the role of which consists in choosing the algorithm that performs best on a specific problem subset [9]. By using several algorithms, a better overall performance is reached: The candidate algorithms "exploit the lack of correlation in the best case performance of several algorithms in order to obtain improved performance in the average case" [8]. Algorithm portfolio is useful when candidate solvers are "Complementary" [9]. This concept depends on two factors: Competitiveness and potential impact.

C. Competitiveness and potential impact

An algorithm selection tool is called "effective" when it outperforms the best single strategy approach. This effectiveness is based on competitiveness and potential impact. We call a set of algorithms competitive if there exist a set of instances for which an algorithm performs best while it is outperformed for another set of problems. The more competitive algorithms are, the more useful is the algorithm selection tool. In addition to competitiveness, the potential impact affects the effectiveness of that tool. It represents the difference in performance between two algorithms on a specific problem. The larger this value, the more effective is the algorithm selection framework [10]. These two conditions are satisfied in our framework.

D. Some algorithm selection tools

Since algorithm selection problem may be applied in various fields of computer science, there exist a lot of its applications such as: An automatic algorithm selection tool for the multi-mode resource-constrained project scheduling problem (2013) [10], SATzilla (2008) which is an algorithm selection tool constructed to solve The propositional satisfiability problems [8], and MUX(2014) which is an algorithm selection tool for software model checkers [11].

III. Our Proposed Framework

Our feature selection framework follows two phases:

A. The learning phase

This step consists in applying all candidate features on each image from the learning database. The selection of the best feature is done automatically by the framework by measuring the performance of each feature on all database images. The result of this step is the training data called "The Empirical Hardness Model". This set contains triplets having the following form: (Image, Best feature, Precision). The average required time for this step is: 42 seconds for 200 images. Note that the used environment is as follows: Intel Core I3, 2GHz frequency, 4GB RAM, Windows7 and Matlab R2016b.

B. The test phase

The purpose of this step is to choose the best feature to characterize each of test database images. For that we use the training set obtained from the learning step. This task is accomplished by the algorithm selector.

The test phase is based on contemporary algorithm selection model presented in [7], Fig. 2 illustrates this model:

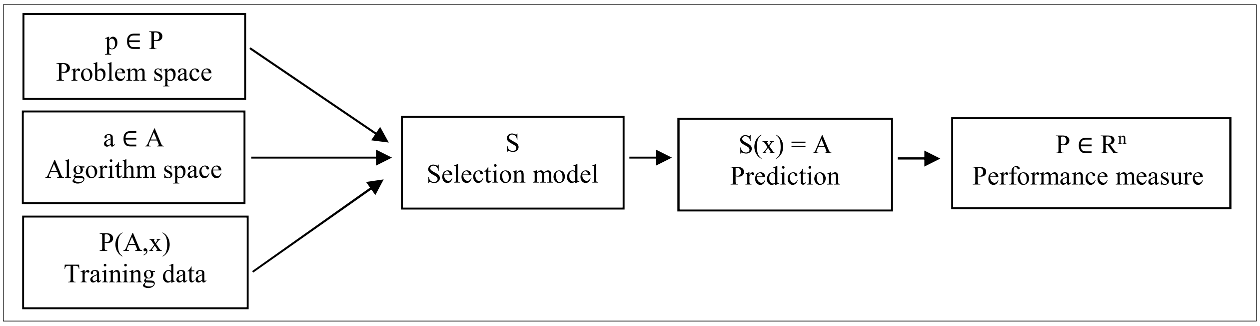

Fig. 2. Contemporary Algorithm Selection framework as presented in [7]

C. *Algorithm selector*

Our experimentations have shown that: "The more similar images are, the higher is the chance of having the same best feature". Based on this assumption, our algorithm selector follows this process:

Input: Training data, candidate features, query image.

Output: Query image characterized by its best feature.

First, our framework Extracts the most 50 similar images to the query image from the training data (The choice of this value will be discussed in the results section).

Second, it retrieves their best features saved in the training data to have a set of 50 features. The result set contains then the six candidate features with redundant values (1 to 50 occurrences for each).

Then, it predicts the best feature which is the most redundant in the result set (The best feature for most 50 similar images).

Finally, the feature is applied on the query image. Fig. 3 illustrates the complete framework:

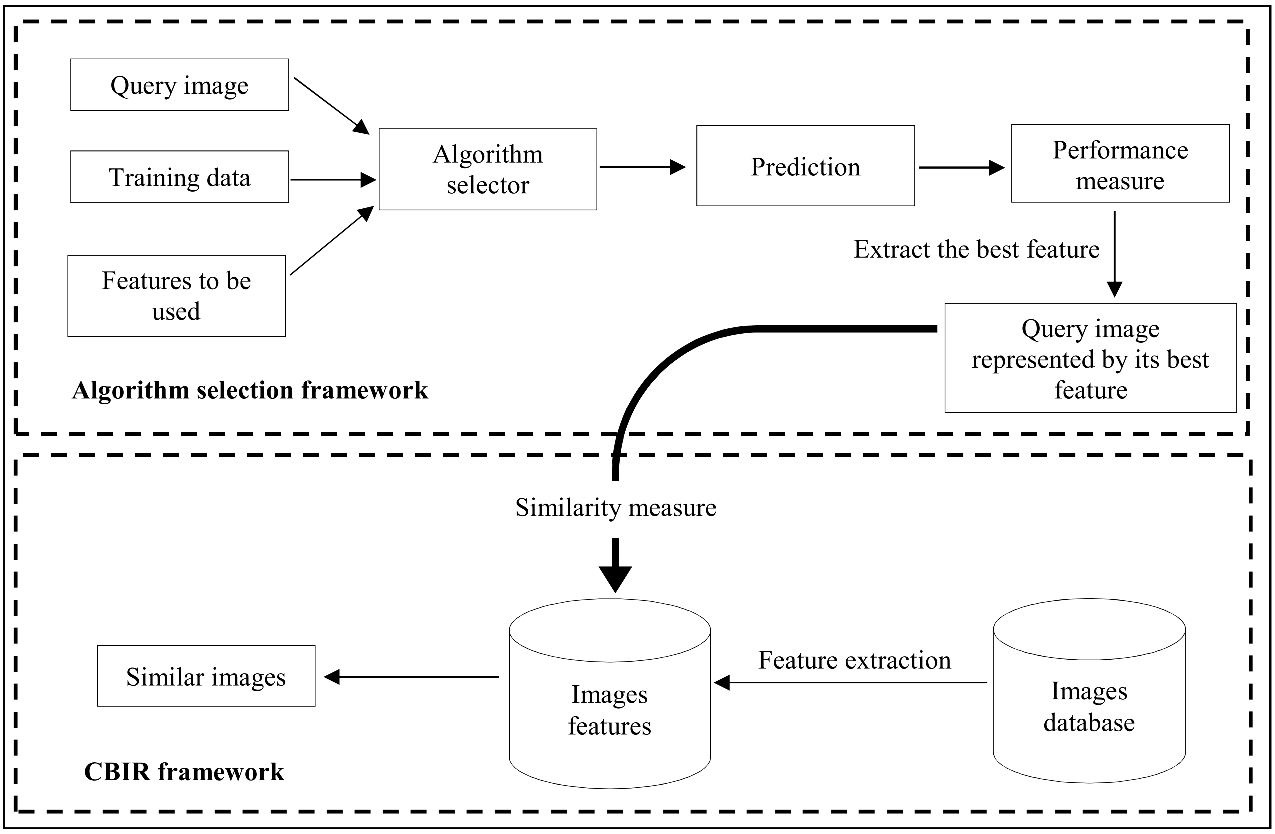

Fig. 3. The functioning of the proposed framework including algorithm selection test phase and content based image retrieval framework

IV. RESULTS

Experimentations have been established on Wang database (Corel 1k). This database includes 1000 images grouped into 10 semantic classes. Fig. 4 shows some images from this database:

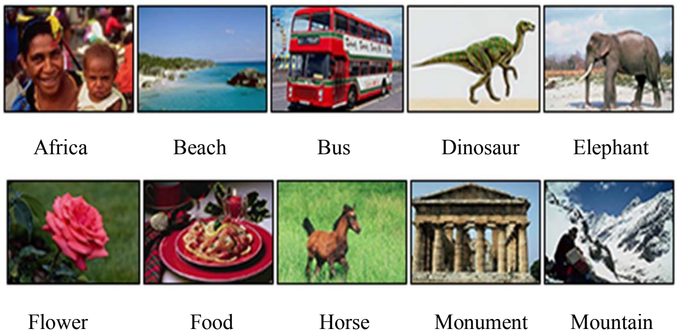

Fig. 4. A brief description of Wang database classes

Our algorithm portfolio is composed of six features: Color based includes color moments and color histogram for both RGB and HSV color spaces, and texture based includes local binary pattern (LBP) and co-occurrence matrice. For similarity measure we opted for Euclidian distance. The obtained results are presented in terms of precision:

$$\text{Precision}=\frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}}$$

Fig. 5 shows a query image from "Horse" class. Fig. 6 and Fig. 7 shows respectively the relevant images retrieved for this query using the best strategy approach and the algorithm selection framework. The obtained precision using the best single strategy approach is 0.6. However, using the algorithm selection framework a precision of 0.9 is reached.

Table I shows the precision of each candidate feature when used separately. Table II shows a comparison between our framework and the best single strategy approach. The obtained results show increase in precision of 4,245%, which confirms the outperformance of our framework on the best single strategy which is RGB color moments.

Fig. 5. Query image from "Horse" class

Fig. 6. Relevant images retrieved using the best single strategy approach

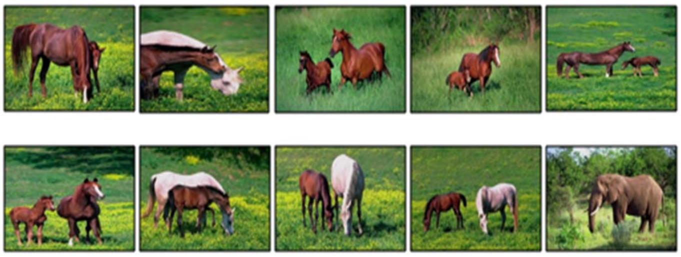

Fig. 7. Relevant images retrieved using the algorithm selection framework

Table III shows the competitiveness between the used features. These features were applied on 200 images in the algorithm selection learning phase, the result is expressed by the number of images having each candidate feature as best feature. We notice that there is a considerable competitiveness between color based features, however texture based features are less competitive. The reason is that color based features overcome texture based features for Wang database as shown in Table I.

As mentioned in the previous section: For each query image, the algorithm selector chooses the best feature based on the most 50 similar images to this query, and then predicts its best feature using the training data. Fig. 8 shows the impact of the number of similar images used in the prediction on the effectiveness of our framework in terms of precision. The best precision is reached with the value "50", which is equal to "0.4972".

Table I. Precision of each of candidate features

	Africa	*Beach*	*Bus*	*Dinosaur*	*Elephant*	*Flower*	*Food*	*Horse*	*Monument*	*Mountain*	*All*
RGB Color moments	0.3675	0.3375	0.4025	0.9925	0,3725	0.4675	0.3825	0.6975	0.23	0.2975	0.45475
HSV Color moments	0.3025	0.2875	0.53	0.9925	0,3025	0.2575	0.4	0.69	0.2875	0.36	0.4465
HSV Histogram	0.5825	0.1775	0.6075	0.75	0.2375	0.3225	0.4	0.87	0.155	0.3175	0.442
RGB Histogram	0.4575	0.195	0.4325	0.7575	0,4575	0.335	0.365	0.575	0.1675	0.2725	0.4115
Local Binary Pattern	0.1925	0.2475	0.74	0.945	0,1925	0.435	0.1875	0.4	0.2725	0.195	0.393
Co-occurrence matrice	0.335	0.1375	0.23	0.9725	0,335	0.4325	0.2875	0.3575	0.0925	0.14	0.3325

Table II. A comparison between RGB color moments and the algorithm selection framework in terms of precision

	Africa	*Beach*	*Bus*	*Dinosaur*	*Elephant*	*Flower*	*Food*	*Horse*	*Monument*	*Mountain*	*All*
RGB Color moments	0.3675	0.3375	0.4025	0.9925	0,3725	0.4675	0.3825	0.6975	0.23	0.2975	0.45475
Algorithm selection framework	0.5225	0.2975	0.5275	0.9925	0.3275	0.46	0.42	0.86	0.2575	0.3075	0.4972

Table III. Competitiveness between condidate features expressed in terms of number of images having each feature as best feature

RGB Color moments	*HSV Color histogram*	*HSV Color moments*	*RGB Color histogram*	*Co-occurrence matrice*	*Local Binary Pattern*
68	52	37	25	14	04

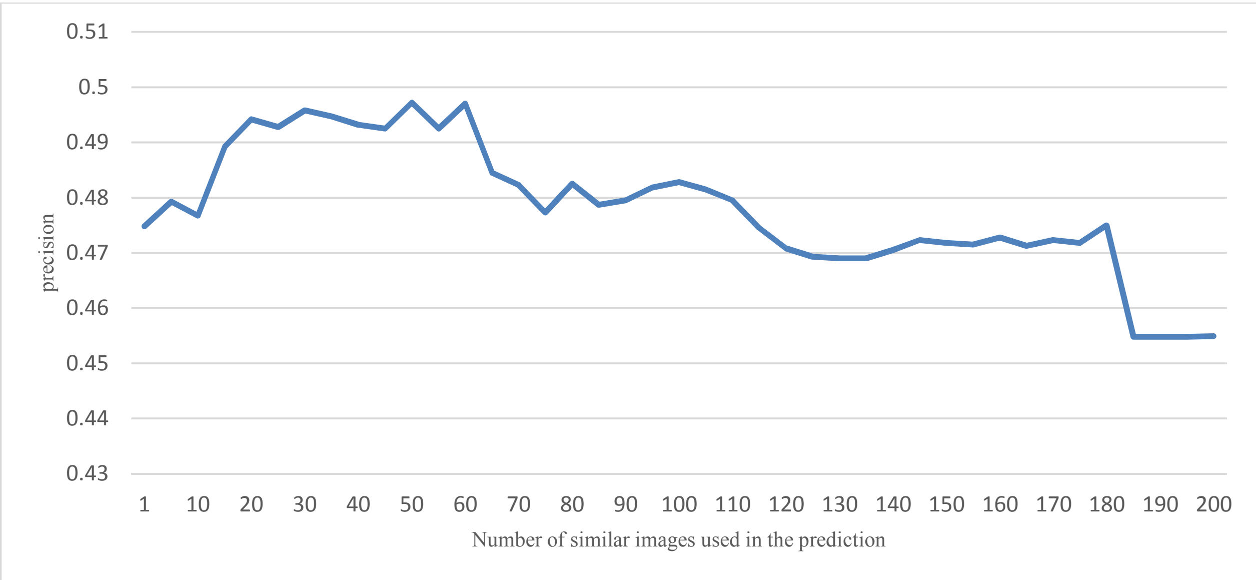

Fig. 8. Precision of the algorithm selection framework in terms of the number of similar images used in the prediction

V. Conclusion

In this paper, we proposed a new framework for feature selection based on RICE model for algorithm selection problem. The proposed framework includes six features: RGB Color moments, HSV Color moments, RGB Color histogram, HSV Color histogram, Local Binary Pattern (LBP), and Co-occurrence matrice. The experimentations were performed on Wang database (Corel 1k).Using this framework, a better overall performance was reached compared to the best single strategy approach. This work can be even more enhanced by including more features, more color spaces, and more distance functions. We also intend to combine the selected features rather than using just one. That is to say, this contribution confirms the effectiveness of algorithm selection in Content Based Image Retrieval. Therefore, it is a good step forward to create a more sophisticated framework by including the mentioned extensions.

References:

[1] Amandeep Khokher, Rajneesh Talwar. “A fast and effective image retrieval scheme using color-,texture-, and shape-based histograms”. Multimed Tools Appl. 2016.

[2] Zhiping Shi & Xi Liu & Qingyong Li & Qing He & Zhongzhi Shi. “Extracting discriminative features for CBIR”. Multimed Tools Appl, pp. 263-279, 2011.

[3] Deepak S. Shete, M.S. Chavan. “Content Based Image Retrieval: Review”. International Journal of Emerging Technology and Advanced Engineering, vol. 2. 2012.

[4] John M. Zachary, Jr. and Sitharama S. Iyengar. “Content Based Image Retrieval systems”. 1999.

[5] Rice, J. R. “The algorithm selection problem”. Advances in Computers, vol. 15, 1976.

[6] Yu-Chi Ho, David L.Pepyne. “Simple explanation of the no free lunch theorem of optimization”. Conference on Decision and Control. Orlando, Florida, USA: IEEE. pp. 4409-4414, 2001.

[7] Kotthof, L.“Algorithm selection for combinatorial search problems: A Survey". AI MAGAZINE, pp. 48-60, 2014.

[8] L. Hu, F. Hutter, H. H. Hoos, K. Leyton-Brown. (2008). "SATzilla: Portfolio-based Algorithm Selection for SAT". Journal of Artificial Intelligence Research, pp. 565-606, 2008.

[9] Mario A. Muñoz, Michael Kirley, and Saman K. Halgamuge. "The Algorithm selection problem on the continuous optimization domain". Computational Intelligence in Intelligent Data Analysis, pp. 75-89, 2012.

[10] Tommy Messelis, Patrick De Causmaecker. "An automatic algorithm selection approach for the multi-mode resource-constrained project scheduling problem". European Journal of Operational Research. 2013.

[11] Varun Tulsian, Aditya Kanade, Rahul Kumar, Akash Lal, Aditya V. Nori. "MUX: Algorithm selection for software model checkers". ACM. 2014.