MTAP manuscript No.
(will be inserted by the editor)

A Genetic Algorithm Approach for Image Representation Learning through Color Quantization

Erico M. Pereira[1] · **Ricardo da S. Torres**[2,3] · **Jefersson A. dos Santos**[1*]

Received: date / Accepted: date

Abstract Over the last decades, hand-crafted feature extractors have been used to encode image visual properties into feature vectors. Recently, data-driven feature learning approaches have been successfully explored as alternatives for producing more representative visual features. In this work, we combine both research venues, focusing on the color quantization problem. We propose two data-driven approaches to learn image representations through the search for optimized quantization schemes, which lead to more effective feature extraction algorithms and compact representations. Our strategy employs Genetic Algorithm, a soft-computing apparatus successfully utilized in Information-retrieval-related optimization problems. We hypothesize that changing the quantization affects the quality of image description approaches, leading to effective and efficient representations. We evaluate our approaches in content-based image retrieval tasks, considering eight well-known datasets with different visual properties. Results indicate that the approach focused on representation effectiveness outperformed baselines in all tested scenarios. The other approach, which also considers the size of created representations, produced competitive results keeping or even reducing the dimensionality of feature vectors up to 25%.

Keywords Color Quantization · Representation Learning · Feature Extraction · Genetic Algorithm · Content-Based Image Retrieval.

[1]Department of Computer Science, Universidade Federal de Minas Gerais – Av. Antônio Carlos, 6627, Belo Horizonte, MG - Brazil – 31270-010
E-mail: {emarco.pereira, jefersson}@dcc.ufmg.br
[2]Institute of Computing, University of Campinas – Av. Albert Einstein, 1251, Campinas, SP - Brazil
[3]Department of ICT and Natural Sciences, NTNU – Norwegian University of Science and Technology – Ålesund – Norway
E-mail: ricardo.torres@ntnu.no
*Corresponding Author: jefersson@dcc.ufmg.br

1 Introduction

It is known that the form in which multimedia data, especially images, are represented can highly impact the performance of machine learning methods typically used in visual pattern recognition tasks, such as content-based image retrieval (CBIR) [54], object detection [64], remote sensing image analysis [11], and image classification [28]. In the last years, representation learning [2], which consists in the process of using pattern recognition algorithms to find representations optimized for a given data domain and/or task at focus, has become a tendency. In fact, the current state-of-the-art methods for representation learning, which are based on deep learning [22] techniques, in many cases present considerable gains in terms of the image content description quality.

However, the use of these methods presents serious drawbacks, such as the broad range of hyper-parameters and possible architectures, the huge computational workload spent to train existing models, the big amount of labeled data required to produce effective models, and the need of specific expertise or training for properly designing, optimizing, and evaluating promising solutions.

Representation learning methods usually employ one of two main approaches: those that learn representations from a feature set provided by a hand-crafted extractor and those that completely compose new ones without any prior feature extraction (from scratch). The latter approach often leads to the usage of more complex and consequently costly methodologies, such as deep learning. Such complexity, however, used to be avoided in the generation of representative features. A few years ago, before the arising of deep neural networks, hand-crafted feature extractors were used to encode image visual properties (e.g., color, texture, or shape) into effective representations [39, 40, 51]. In general, those solutions rely on less costly algorithms and do not depend on previously annotated datasets or time-consuming learning steps. On the other hand, these feature extractors are application-dependent, being less generalizable.

In this paper, we propose a hybrid scheme, focused on color quantization, which aims to take advantage of both research venues. We propose data-driven color quantization schemes, which improve the effectiveness of hand-crafted feature extractors, as it allows for the identification of discriminative visual features. Our representation learning scheme exploits a particular characteristic of the current image context, its color distribution, a simple but yet suitable visual cue in several applications [17, 25, 32]. Our hypothesis is that data-driven quantization optimizations are able to positively impact the quality of image content description approaches, leading to effective and efficient representations. In this paper, we investigate how these optimizations can be performed effectively and efficiently and to what extent.

Our color quantization optimization relies on a soft computing framework, implemented using genetic algorithms (GA). GA is an evolutionary algorithm widely used to solve optimization problems. According to its formulation, a population of individuals, representing possible solutions of a problem, evolves over generations, subjected to genetic operations. The goal is to find the best individuals, i.e., the best solutions for the problem. In our color quantization problem, a GA individual encodes how color channels should be divided in order to improve the effectiveness of feature extractors. To the best of our knowledge, this is the first work to use GA to model the representation learning problem.

In summary, the main contributions of this work are:

1. We show that different color quantizations impact the effectiveness performance of feature extractors;
2. We model the search of suitable color quantization using a soft computing apparatus based on the genetic algorithm;
3. We introduce two approaches for supervised representation learning capable of providing compact and more effective representations through color quantization optimization.

In summary, the main novelty of our work relies on the presentation of an integrative framework for the implementation of effective image search systems that combines several concepts, approaches and techniques, such as, Genetic Algorithm optimization, Color Quantization, Representation Learning, Feature Extraction, and Content-Based Image Retrieval.

We conducted a series of experiments in order to evaluate the robustness of the proposed approaches in content-based image retrieval tasks, considering eight well-known datasets containing images with different visual properties. Experimental results indicate that the approach focused on the representation effectiveness outperformed the baselines in all tested scenarios. The other approach, which focuses not only on the effectiveness, but also on the size of the generated feature vectors, was able to produce competitive results by keeping or even reducing the final feature vector dimensionality up to 25%.

The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 provides a background upon base methods used in our work. Section 4 describes the proposed color quantization schemes and its use in CBIR tasks. Section 5 details the experiments performed to assess the effectiveness and efficiency of the proposed methods. Section 6, in turn, presents and discusses achieved results. Finally, Section 7 presents our main conclusions and outlines possible future research directions.

2 Related Work

Image representation learning (a.k.a. feature learning) consists in automatically discovering the representations needed for object detection or classification from raw images. It is a set of approaches that aim at making it easier to extract useful information when building classifiers or other predictors [3]. In other words, feature learning allows to find the most suitable or discriminative representation from the raw data according to some constraint imposed by the target application. Thus, it is also commonly known as data-driven features because of its contraposition to engineered or hand-crafted features.

Although feature learning has been an active research area for a long time, the development of effective techniques (mainly based on deep learning) has been boosted in the last decade mainly due to the spread use of powerful computational resources, which were motivated by the development of graphical processing units (GPUs). Many successful recent feature representation approaches are based on deep belief nets [16], denoising auto-encoders [58], deep Boltzmann machines [50], K-Means-based feature learning [8], hierarchical matching pursuit [6],

and sparse coding [65]. Regarding image representation learning, the most successful approaches are based on the Convolutional Neural Networks (CNNs) [21].

Although, by definition, a large number of techniques perform feature learning, the term is most commonly employed by the community that develops methods based on deep learning or probabilistic graphical models. These methods are the basis for most of the state-of-the-art approaches for pattern recognition and computer vision. Despite the recent great success of these approaches, they still have several limitations, such as a large number of parameters for optimization and the difficulty in designing network architectures.

Evolutionary algorithms are meta-heuristic optimization techniques that use mechanisms inspired by biological evolution (e.g., reproduction, mutation, recombination, and selection). They have been widely employed in a myriad of frameworks developed for image analysis and retrieval usually for feature fusion [49] or selection [33, 37]. In the last few years, evolutionary algorithms have also been successfully employed for neural networks architecture search [56, 62]. Nonetheless, we did not find other works that directly model feature learning as an evolutionary algorithm-based problem from the raw data.

In this work, we propose to learn image features from images via genetic algorithms by color quantization optimization. Some works developed quantization learning using evolutive heuristics for image segmentation [29]. Scheunders [52] handles the quantization problem as global image segmentation and proposes an optimal mean squared quantizer and a hybrid technique combining optimal quantization with a Genetic Algorithm modelling [14]. Further, the same author [52] presents a genetic c-means clustering algorithm (GCMA), which is a hybrid technique combining the c-means clustering algorithm (CMA) with Genetic Algorithm. Lastly, Omran et al. [38] developed colour image quantization algorithm based on a combination of Particle Swarm Optimization (PSO) and K-means clustering.

Regarding the effects of colour quantization on image representations, Ponti et al. [42] approached the colour quantization procedure as a pre-processing step of feature extraction. They applied four fixed quantization methods – Gleam, Intensity, Luminance, and a concatenation of the Most Significant Bits (MSB) – over the images of three datasets and then used four feature extractors – ACC, BIC, CCV, and Haralick-6 – to compute representations intended to solve the tasks of Image Classification and Image Retrieval. Their conclusions show that it is possible to obtain compact and effective feature vectors by extracting features from images with a reduced pixel depth and how the feature extraction and dimensionality reduction are affected by different quantization methods.

New approaches based on deep learning developed in the last ten years have revolutionized the learning of representations from data. Regarding the learning of representations for images, convolutional networks have established themselves as the most effective solution. However, its use still has some limitations, such as: (1) they require a large amount of data for training from scratch; (2) traditional networks have a large number of parameters. Therefore, some works have been proposed in order to mitigate these limitations and produce more compact networks [31, 68]. In this context, approaches based on nature-inspired/evolutionary algorithms have emerged as an alternative to optimize network architectures in various ways [4, 46]. Although color quantization approaches are less used nowadays than in the past for image representation, they are still an alternative to

obtain compact and effective representation for some applications, such as color-based image retrieval [5, 41, 53, 66].

To the best of our knowledge, our work is the unique that provides an application-driven way to learn compact representation from color quantization. Note that it is not comparable to modern neural network-based compact approaches because it does not take advantage of transfer learning strategies.

3 Background

This section presents background concepts on feature extraction algorithms (Section 3.1) and genetic algorithms (Section 3.2). The feature extraction algorithms described here refer to methods that are combined with the quantization scheme defined by GA in the performed experiments.

3.1 Color Quantization-based Feature Extraction Algorithms

Border/Interior Classification (BIC). Stehling et al. [55] proposed BIC, a simple and fast approach for feature extraction which presented prominent results in web image retrieval [39] and remote sensing image classification [36, 51]. This approach relies on an RGB color-space uniformly quantized in $4 \times 4 \times 4 = 64$ colors. After the quantization, the authors propose to apply a segmentation procedure, which classifies the image pixels according to a neighborhood criterion: a pixel is classified as interior if its 4-neighbours (right, left, top, and bottom) have the same quantized color; otherwise, it is classified as border. Then, two color histograms, one for border pixels and other for interior pixels, are computed and concatenated composing a 128-bin representation. In the end, the histograms undergo two normalizations: division by the maximum value, for image dimension invariance, and a transformation according to a discrete logarithmic function ($dLog$), aiming to smooth major discrepancies. When comparing BIC via L1 distance, it was observed that the dLog function is able to increase substantially the effectiveness of histogram-based CBIR approaches and also reduces by 50% the space required to represent a histogram.

Global Color Histogram (GCH). GCH [57] is a widely used feature extractor that presents one of the simplest forms of encoding image information in a representation, a color histogram, which is basically the computation of the pixel frequencies of each color. It relies on the same uniformly quantized RGB color-space such as BIC and, consequently, produces a feature vector of 64 bins. After the histogram computation, it undergoes a normalization by the max value in order to avoid scaling bias. Additionally, for the same reasons as for BIC, dLog normalization is also applied to the final histogram.

3.2 Genetic Algorithm

GA is a bio-inspired optimization heuristic that mimics natural genetic evolution to search the optimal in a solution space [14]. It models potential solutions for

the problem as individuals of a population and subjects them to an iterative process of combinations and transformations towards an improved population, i.e., a population with better solutions for the target problem.

At each step, GA randomly selects individuals from the current population, called parents, in an operation called tournament, in which individuals are grouped and only the best ones are selected. From this selection, GA exchanges genetic material of the individuals in order to produce new individuals of the next generation. This operation is known as cross-over. Some individuals are also selected to undergo a mutation operation, which consists in randomly changing small pieces of the individual representation. This new individual is also integrated into the new generation [10]. Typically a few of the best individuals of the population also compose the new one, a practice known as elitism. When a new generation is formed, its individuals are evaluated by means of a fitness function, which assesses the individual (solution) performance on the target problem. According to this function score, the algorithm selects the parent individuals that will generate the next population, simulating a natural selection process. At the end of the process, when the stopping condition is satisfied, the expected result is the best-performing individual, i.e., the one that best solves the target problem.

3.3 Winner-Take-All (WTA) Autoencoders

An autoencoder [15] is a framework that employs representation learning by optimizing an encoding that reconstructs as well as possible the entry data. It is specified by a explicitly defined feature-extraction function f_θ, called encoder, which allows the computation of a representation $z = f_\theta(x)$ from a given input x, and a parametrised function g_θ, that maps the representation from feature space back to input space producing a reconstruction $r = g_\theta(x)$. The set of parameters θ of the encoder and decoder are learned simultaneously by reconstructing the original input x with the lowest possible discrepancy $L(x, r)$ between x and r, employing a optimization process that minimizes:

$$\Gamma_{AE}(\theta) = \sum_t L(x^{(t)}, g_\theta(f_\theta(x^{(t)}))) \tag{1}$$

where $x^{(t)}$ is a training sample.

It is crucial that an autoencoder presents good generalization, i.e., that the produced representations yield low reconstruction error for both train and test samples. For this purpose, it is important that the training criterion or the parametrisation prevents the auto-encoder from learning the identity function to the training samples, which presents zero reconstruction error. This is achieved by imposing different forms of regularisation in different versions of autoencoders. Regularized Autoencoders limit the representational capacity of z provoking a bottleneck effect that does not allow the autoencoder to reconstruct the whole input and forces it to learn more meaningful features. As a consequence, it is trained to reconstruct well the training samples and also present small reconstruction error on test samples, implying generalization.

The most common types of regularised autoencoders include: Sparse autoencoders [30, 35, 43], which limit capacity by imposing a sparsity constraint on the learnt representation of the data; Denoising autoencoders [58, 59], which has the

objective of removing noise of an artificially corrupted input, i.e. learning to reconstruct the clean version from a corrupted data; Contractive Autoencoders [44], which penalize the sensitivity of learned features to input variations producing more robust features; and Variational Autoencoders [19], which learn probabilistic latent spaces in order to generate artificial samples.

Winner-takes-all Autoencoders (WTA-AE) [31] are sparse autoencoders that employ two types of sparsity constraints:

- A spatial sparsity constraint, which, rather than reconstructing the input from all of the representational hidden units, selects the single largest value within each feature map, and set the rest to zero. This results in a sparse representation whose sparsity level is the number of feature maps and in a reconstruction which uses only the active hidden units in the feature maps;
- A *winner-take-all* lifetime sparsity constraint, which maintains only the $k\%$ largest values of each feature map, and set others to zero, considering the values selected spatial sparsity within an entire mini-batch.

We choose WTA as baseline because it is one of the most robust and efficient Sparse Enconders – the most effective class of (non-generative) methods based on deep learning that are dedicated for feature extraction/representation learning. WTA autoencoders were capable of aiming at any target sparsity rate, training very fast compared to other sparse autoencoders, and efficiently training all hidden units even under very aggressive sparsity rates (e.g., 1%). Furthermore, the usage of its sparsity properties allows the train of non-symmetrical architectures (different sizes for encoder and decoder) reducing computation and data resource consumption.

4 GA-based Color Quantization

In this paper, we introduce the use of Genetic Algorithm to learn an optimized color quantization for a given image domain. Figure 1 provides an overview of the entire process, which is composed of two main steps: (A) quantization search, and (B) feature extraction. These steps are described next.

4.1 Quantization Search

We propose the use of Genetic Algorithm [14] to learn the best color quantization for a given collection. GA has been a widely used approach for finding near-optimal solutions for optimization problems. One remarkable property of this optimization apparatus relies on its ability in performing parallel searches starting from multiple random initial search points and considering several candidate solutions simultaneously. Consequently, it represents a fair alternative to an exhaustive search strategy, which would be unfeasible given the number of possible solutions.

According to this optimization algorithm, an individual corresponds to a representation of a potential solution to the problem that is being analyzed. In our modeling, each individual represents a possible color quantization, as detailed in

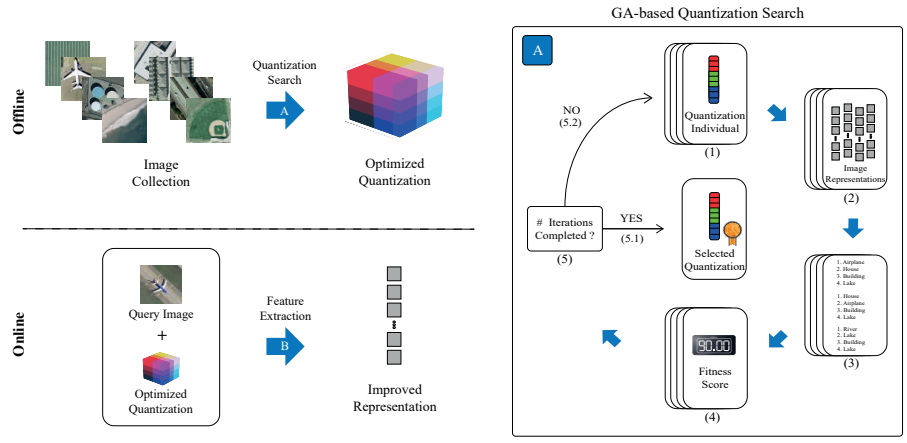


Fig. 1: Overview of the proposed approach. First, (A) we use Genetic Algorithm to search for an optimized color quantization. Later, (B) the resulting quantization is incorporated into the feature extractor to generate improved image representations. The GA-based quantization search proceeds as follows: first, (1) a population of encoded color quantizations is randomly produced; second, (2) sets of image representations of the whole collection are produced being each one according to one quantization color space; third, (3) similarity rankings for all to all images are computed within each representation set; and fourth, (4) a fitness score is computed to measure each retrieval effectiveness. Finally, if the stopping condition is met or the total number of iterations is achieved, (5.1) the quantization of the highest fitness of the last population is selected as the optimized colour space, otherwise, (5.2) a new population is created, via crossover and mutation operations over the current population, initiating the next iteration.

Section 4.1.1. During the evolution process, described in Section 4.1.2, these individuals are gradually evolved. At the end of the evolutionary process, the best-performing individual, which encodes a quantization that leads to an improved representation, is selected.

4.1.1 Quantization Encoding

In our modeling, a quantization is represented in a GA individual as follows: Let $\mathbb{M}$ be a color model composed of three channels. Without loss of generality, we will assume the RGB color model from now on. Assume that each channel is divided into 256 discrete levels, i.e., eight bits can be used to define the number of colors in each channel. In the case of the traditional 24-bit RGB model, there are almost 17 million ($256 \times 256 \times 256$) different colors.

In our formulation, a 24-bit long GA individual encodes the number of partitions of the different channels. Figure 2 (top) presents the typical 24-bit RGB channel partitioning. Figure 2 (middle) illustrates a possible GA individual encoding how each channel should be divided. Figure 2 (bottom), in turn, illustrates the resulting color quantization after using the GA individual encoding.

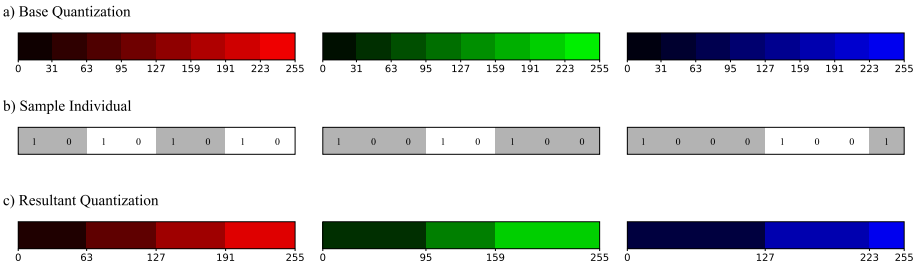
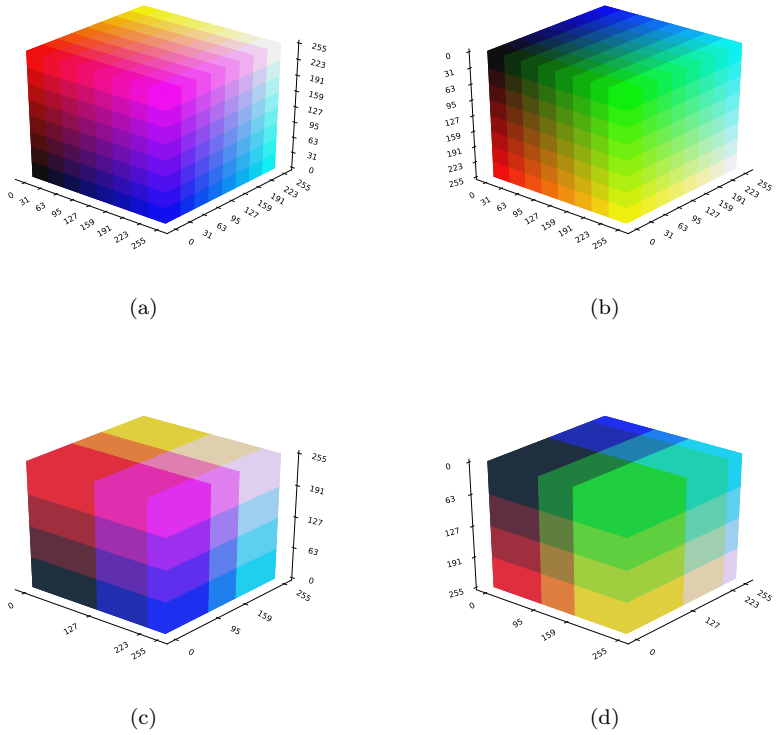


Fig. 2: Our modeling takes reference from a base quantization (a) representing each interval of color tonalities as a bit in individuals implemented as binary arrays (b). These bits dictate the union of intervals producing a new quantization (c): if a bit is set, its respective interval has its own position, otherwise, it is aggregated to the immediate previous interval. The first bit of each color axis is forced to always be set.

Figure 3 presents the RGB color space before (a-b) and after (c-d) using the GA-based encoding defined in Figure 2(middle). Figures 3(b) and 3(d) present different views of the same color space presented in Figures 3(a) and 3(c), respectively.



(a) (b)

(c) (d)

Fig. 3: (a) RGB color space using the traditional 8-bit quantization per channel. (b) The same color space presented in (a), but rotated in 180° over the Z axis. (c) Color space after applying the GA individual illustrated in Figure 2 (middle). (d) The same color space presented in (c), but rotated in 180°over the Z axis.

4.1.2 GA-based Quantization Search

Algorithm 1 illustrates the proposed GA-based quantization. The population starts with individuals created randomly (line 3). The population evolves generation by generation through genetic operations (line 4). A function (described in Section 4.3) is used to assign the fitness value for each individual (lines 5-7), i.e., to assess how well an individual solves the target problem. According to the elitism operation, the top k best individuals of the current generation are recorded (line 8). Then, individuals from P are selected according to a tournament operation of n_t-sized groupings (line 9). After that, the next generation is formed from the union of the resulting individuals from the operations of mutation and cross-over over the tournament selection and those selected in elitism (line 10). If the stopping condition (discussed on Section 5.3) were met, the iterations stop (lines 11-13). The last step is concerned with the selection of the best individual q^* of all generations (line 15). The individual q^* is used later to define the quantization used in the

feature representation process. For details regarding genetic operators (cross-over, mutation, tournament and elitism), refer to Section 5.3.

Algorithm 1 GA-based quantization

Let T be a training set
Let P, S_e e S_t be sets of pairs $(q, fitness_q)$, where q and $fitness_q$ are an individual and its fitness, respectively
$P \leftarrow$ Initial random population of individuals
For each generation g of N_g generations **do**
    For each individual $q \in P$ **do**
        $fitness_q \leftarrow fitness(q, T)$
    End For
    $S_e \leftarrow elitism(k, P)$
    $S_t \leftarrow tournament(n_t, P)$
    $P \leftarrow S_e \cup mutation(S_t) \cup crossover(S_t)$
    If stopping condition is met
        Break outer loop
    End If
End For
Select the best individual $q^* = \arg\max_{q \in P}(fitness_q)$

4.2 Feature Extraction

In the second phase, the best individual, i.e., the one which leads to the best quantization q^* is used with the feature extractor algorithm to produce a color image representation. In order to do that, it was necessary to implement a slightly modified version of the feature extractor, that incorporates the capacity of generating representations according to a specified color quantization. Equations 1, 2, and 3, where M_c is the maximum color axis size and q^* is the quantization individual, define how to calculate the new R, G, and B (referred to as R_{new}, G_{new}, and B_{new}, respectively) values for each pixel. In this work, according to empirical observations, M_c was chosen as 8.

$$R_{new} = \left(\sum_{i=0}^{r} q^*[i] \right) \times \frac{|R_{axis}|}{256}, \tag{2}$$

$$where \quad r = R \times \frac{M_c}{256}; \quad |R_{axis}| = \sum_{l=0}^{M_c} q^*[l]$$

$$G_{new} = \left(\sum_{j=N}^{g+M_c} q^*[i] \right) \times \frac{|G_{axis}|}{256}, \tag{3}$$

$$where \quad g = G \times \frac{M_c}{256}; \quad |G_{axis}| = \sum_{m=M_c}^{2M_c} q^*[m]$$

Fig. 4: This figure[1] shows the visual effect of different color quantizations on different sample images. The first column shows the RGB color spaces defined according to the specified quantizations: the original space in which the image is captured, a widely-used hand-crafted quantization scheme using 64 colors, and an example of optimized quantization defined by our method. The remaining columns show sample images after using each quantization scheme.

$$B_{new} = \left(\sum_{k=2M_c}^{b+2M_c} q^*[i] \right) \times \frac{|B_{axis}|}{256}, \tag{4}$$

$$where \quad b = B \times \frac{M_c}{256}; \quad |B_{axis}| = \sum_{n=2M_c}^{3M_c} q^*[n]$$

Figure 4 shows the visual effect of different color quantizations on different sample images. The first column shows the RGB color spaces defined according to the specified quantizations: the original space in which the image is captured, a widely-used hand-crafted quantization scheme using 64 colors, and an example of optimized quantization defined by our method. The remaining columns show sample images after using each quantization scheme. Original images are shown in the top line. Above each quantized image, we present the color spectrum and its respective histogram.

4.3 Individual Fitness Computation

The use of the proposed GA-based quantization leads to discriminative features, which may be useful in different applications, such as Image Classification [51], Image Retrieval [39], and Object Recognition. In this paper, we opted for evaluating the method in the context of Content-Based Image Retrieval (CBIR) [54] tasks.

[1] We recommend colourful printing for adequate visualization.

The goal of this task is to retrieve the most relevant images from a collection, given their similarity to a given query image. The similarity computation relies on the use of a distance (or similarity) function applied to feature vectors, which encode their content (in our case, their color properties).

We first extract feature vectors from all images within a collection, by taking into account feature extractors that benefit from the learned color quantization. Collection images are later ranked according to the distance of their feature vectors to the feature vector of a query using the Manhattan Distance (L1). Two images belonging to the same class are assumed to be relevant to each other. Given a query image, our goal is to produce a ranked list with collection images of the same class of the query on top positions. The more relevant images on top positions, the more effective is the ranked list, i.e., the more effective is the description approach.

More formally, an image img is firstly encoded through a feature extraction procedure, which allows quantifying the similarity between images. Let $C = \{img_1, img_2, \ldots, img_n\}$ be a collection with n images. Let $\mathcal{D}$ be a descriptor, which can be defined as a tuple (ϵ, δ) [47], where:

- ϵ: $img_i \rightarrow \mathbb{R}^d$ is a function, which extracts a feature vector v_i from an image img_i;
- δ: $\mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$ is a function that computes the distance between two images according to the distance between their corresponding feature vectors.

The distance between two images img_i, img_j is computed as $\delta(\epsilon(o_i), \epsilon(o_j))$. The Euclidean distance is commonly used to compute δ, although the proposed ranking method is independent of distance measures. A similarity measure $\rho(img_i, img_j)$ can be computed based on distance function δ and used for ranking tasks. We will use $\rho(i, j)$ from now on to simplify the notation.

The target task refers to retrieving multimedia objects (e.g., images, videos) from C based on their content. Let img_q be a query image. A ranked list τ_q can be computed in response to img_q based on the similarity function ρ. The ranked list τ_q=(img_1, img_2, ..., img_n) can be defined as a permutation of the collection C. A permutation τ_q is a bijection from the set C onto the set $[N] = \{1, 2, \ldots, n\}$. For a permutation τ_q, we interpret $\tau_q(i)$ as the position (or rank) of the image img_i in the ranked list τ_q. If img_i is ranked before img_j in the ranked list of img_q, i.e., $\tau_q(i) < \tau_q(j)$, then $\rho(q, i) \geq \rho(q, j)$.

Given a training set composed of a set of queries and their respective list of relevant objects, the fitness of an individual is measured as a function of the quality (effectiveness) of ranked lists produced for each query, considering the use of a feature extractor implemented using the GA-based quantization. The more relevant images found at top positions, the better the GA individual is.

4.4 Computational Complexity of GA-based Quantization Search

The GA training procedure takes $\mathcal{O}(N_g \times N_i \times F)$, where N_g is the number of generations considered in the evolution process, N_i is the number of individuals in the population, and F is the cost for evaluating the fitness function.

The costs for computing F depends on the number of training samples N_s and the size of pre-computed histograms S_h. The later, in the worst case, is k^3, where k is the number of bins in a color axis. As overlying detailed base color

Table 1: Image datasets and statistics

Dataset	# of samples	# of classes	Images content
Coil-100 [34]	7,200	100	objects
Corel-1566 [60]	1,566	43	mixed (objects, landscapes etc)
Corel-3906 [60]	3,906	85	mixed (objects, landscapes etc)
ETH-80 [23]	3,280	80	objects
MSRCORID [9]	4,320	20	mixed (scenes and objects)
Groundtruth [26, 27]	1,285	21	landscapes
Supermarket Produce [45]	2,633	15	fruits
UC Merced Land-use [63]	2,100	21	aerial scenes

spaces does not improve the results, k is typically small, making S_h also small ($k = 8$ and $S_h = k^3 = 512$ in our experiments). As a consequence, F takes $\mathcal{O}(N_s \times S_h)$ for feature extraction and $\mathcal{O}({N_s}^2 \times logN_s)$ for computing rankings, then $\mathcal{O}(F) = \mathcal{O}({N_s}^2 \times logN_s)$.

Finally, the whole procedure takes $\mathcal{O}(N_g \times N_i \times {N_s}^2 \times logN_s)$ to find the final quantization. Recall that the training process is performed offline.

4.5 Quantization Approaches

In this paper, we propose two formulations of the GA-based quantization method. The first, named Unconstrained Approach (UA), is intended to provide a quantization focused on generating representations that have the best possible effectiveness performance. The second, named Size-Constrained Approach (SCA), focuses not only on effectiveness aspects, but also on the size of the representation. The goal is to find the best-performing individual, which leads to feature vectors with a pre-defined size, i.e., the target feature vector size is defined a priori. From the implementation point of view, the GA-based quantization approach assigns a negative fitness score for the individuals that present dimensions higher than the pre-defined feature vector size. As a consequence, this latter formulation tends to produce more compact representations.

5 Experimental Setup

In this section, we present the adopted experimental setup, which concerns the image datasets considered (Section 5.1), the configuration of parameters of the method (Section 5.3), the baselines used for comparative analysis (Section 5.2), the metrics used to evaluate the effectiveness and compactness of the produced feature vectors (Section 5.4), and the employed experimental protocol (Section 5.5).

5.1 Datasets

In order to assess the effectiveness of the employed quantization approach, we conducted experiments using eight different image datasets, which are described next. For convenience, Table 1 summarizes some important information about them.

- ***Coil-100***: This dataset [34] comprises images of 100 everyday objects, being each one used to define a different class. Pictures of each object were taken in 72 different poses composing a total set of 7,200 images. Some samples of this dataset are shown in Figure 6.
- ***Corel-1566 and Corel-3906***: These datasets [60] correspond to two sets from a collection with 200,000 images from the Corel Gallery Magic–Stock Photo Library 2. The first (Fig. 7) contains 1,566 samples distributed among 43 classes, while the second (Fig. 8) contains 3,906 samples among 85 classes. Besides the images quantity, the main difference between them is that the latter presents more intra-class variability.
- ***ETH-80***: This dataset [23] was originally tailored to the task of object categorization. It includes images of 80 objects from 8 basic-level categories. Each object is represented by 41 views over the upper viewing hemisphere, performing a total of 2,384 images. Some samples of this dataset are shown in Figure 9.
- ***Groundtruth***: This dataset [26, 27] contains a variety of 1,285 scenes and objects grouped among 21 high-level concepts, such as: Arbor Greens, Australia, Barcelona, Cambridge, Campus In Fall, Cannon Beach, Cherries, Columbia George, Football, Geneva, Green Lake, Greenland, Indonesia, Iran, Italy, Japan, Leafless Trees, San Juans, Spring Flowers, Swiss Mountains, Yellow Stone. Figure 12 depicts some of its classes.
- ***Microsoft Research Cambridge Object Recognition Image Database (MSRCORID)***: This collection [9] contains a set of 4,320 images of scenes, objects and landscapes. Its images are grouped into 20 categories: Aeroplanes, Cows, Sheep, Benches and Chairs, Bicycles, Birds, Buildings, Cars, Chimneys, Clouds, Doors, Flowers, Kitchen Utensils, Leaves, Scenes Countryside, Scenes Office, Scenes Urban, Signs, Trees, Windows. Some samples of this dataset are shown in Figure 10.
- ***Supermarket Produce***: This dataset [45] contains images of fruits and vegetables collected from a local distribution center. It comprises 2,633 images distributed into 15 different categories: Plum, Agata Potato, Asterix Potato, Cashew, Onion, Orange, Tahiti Lime, Kiwi, Fuji Apple, Granny-Smith Apple, Watermelon, Honeydew Melon, Nectarine, Williams Pear, and Diamond Peach. Figure 11 depicts some samples of its categories.
- ***UC Merced Land-use***: This dataset [63] is composed of 2,100 aerial scene images divided into 21 classes selected from the United States Geological Survey (USGS) National Map. Its 21 categories are Agricultural, Airplane, Baseball Diamond, Beach, Buildings, Chaparral, Dense Residential, Forest, Freeway, Golf Course, Harbor, Intersection, Medium Density Residential, Mobile Home Park, Overpass, Parking Lot, River, Runway, Sparse Residential, Storage Tanks, and Tennis Courts. Some samples of this dataset are shown in Figure 5.

(a) Forest (b) Beach (c) Tennis Court

(d) Dense Residential (e) Medium Residential (f) Sparse Residential

Fig. 5: Examples of the *UC Merced Land-use* dataset.

(a) Object 13 (b) Object 14

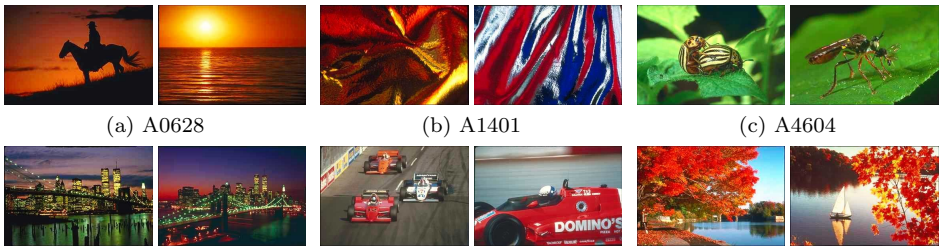
(c) Object 87 (d) Object 81

Fig. 6: Examples of the *COIL-100* dataset.

(a) A6140 (b) A14935 (c) A2231

(d) A0908 (e) A7840 (f) A0004 (g) A12147

Fig. 7: Examples of the *COREL-1566* dataset.



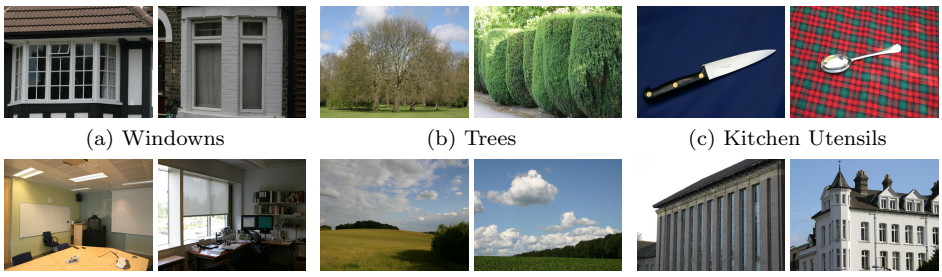
(a) A0628 (b) A1401 (c) A4604

(d) A4932 (e) A7601 (f) A14208

Fig. 8: Examples of the *COREL-3906* dataset.

. (a) Car (b) Cow (c) Cup

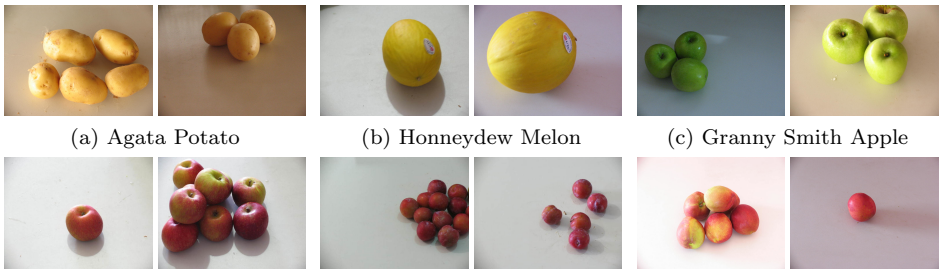
Fig. 9: Examples of the *ETH-80* dataset.



(a) Windowns (b) Trees (c) Kitchen Utensils

(d) Scenes Office (e) Scenes Countryside (f) Buildings

Fig. 10: Examples of the *MSRCORID* dataset.



(a) Agata Potato (b) Honneydew Melon (c) Granny Smith Apple

(d) Fuji Apple (e) Plum (f) Nectarine

Fig. 11: Examples of the *Supermarket Produces* dataset.

(a) Cambridge (b) Australia (c) Columbia Gorge

(d) Barcelona (e) Indonesia (f) Geneva (g) Japan

Fig. 12: Examples of the *Groundtruth* dataset.

5.2 Baselines

5.2.1 Feature Extraction Algorithms

In order to demonstrate the impact of using the learned quantizations in the generation of more effective image representations, we compare GA-based feature extractors with similar formulations *without* any quantization procedure. We use the BIC and the GCH original formulations (see Section 3.1) as baselines.

5.2.2 Winner-Take-All Autoencoder

We also perform comparisons with autoencoders (see Section 3.3), a class of methods based on Deep Learning – state-of-the-Art framework for computer vision – dedicated to perform representation learning and, consequently. They are, therefore, suitable recent approaches for comparison purposes.

According to Makhzani et al. [30], Sparse Autoencoders (SAE) yield the best performance than other types, such as Denoising Autoencoders, for feature extraction in tasks such as image classification. Among SAEs, we selected Winner-Take-All Autoencoders WTA-AE (see Section 3.3) which hold some advantages in comparison with other SAEs including the capability of aiming any sparsity rate, efficient training and resource consumption besides allowing the use of reduced architectures.

Among the WTA-AE proposed configurations, we selected the CONV-WTA autoencoder, which is a non-symmetric architecture where the encoder consists of a stack of three 256-units ReLU convolutional layers (5×5 filters) and the decoder is a 256-units linear deconvolutional layer of larger size (11×11 filters): 256conv3-256conv3-256conv3-256deconv7. It also maintains N_u hidden representation units between the encoder e decoder. This is the same architecture used by Makhzani et al. [31] in experiments for the CIFAR-10 dataset [20], an image dataset of domain similar to the ones used in our experiments.

Following the instructions of Makhzani et al., with the purpose of composing representations adequate to being used on an image classification/retrieval setting, we employed, after training, max-pooling on the last N_u feature maps of the encoder, over 6×6 regions at strides of 4 pixels to obtain the final representation of $N_u \times 8 \times 8 = N_u \times 64$ total size. In order to allow a fair performance comparison between the different-sized representations of WTA-AE and SCA, we employed Principal Components Analysis [61] – a well-known data projection algorithm – on the WTA-AE representation as dimensionality reduction procedure where the number of dimensions corresponded to the imposed representation size limits.

5.3 Parameters

Table 2 presents the values adopted for the GA-based quantization learning process. The values chosen for population size, cross-over, mutation, elitism, and tournament parameters were defined empirically, but all of them represent typical values employed in GA-based optimization solutions. Initially, it was applied a parameter search according to a 2^k Fractional Factorial Design (please refer to

Table 2: Genetic algorithm parameters. The indicated variables refer to Algorithm 1.

Two-point Cross-over Probability	60%
One-point Mutation Probability	40%
Number of Generations (N_g)	200
Population Size	200
Tournament (n_t)	5
Elitism (k)	1%

item 16.3.3 of [7]) over a portion of the dataset. For the parameters which presented major sensitivities, a binary search was employed for exploration of different values.

The total number of generations was defined aiming to ensure convergence of the evolutionary algorithm. However, we empirically observed that typically the best fitness value is not significantly improved after remaining unchanged for more than 50 iterations. Thus, one might impose a stopping condition regarding the fitness value as an option to avoid unnecessary iterations.

We assess the quality of ranked lists defined by image representations obtained by means of a GA individual through the FFP4 function [12]. This score is defined for a given query image q as:

$$FFP4_q = \sum_{i=1}^{|D|} r_q(d_i) \times k_8 \times k_9^i \tag{5}$$

where D is the image dataset; $r_q(d) \in [0, 1]$ is the relevance score for the image d_i associated to the query, it being 1 if relevant and 0 otherwise; and k_8 and k_9 are two scaling factors adjusted to 7 e 0.982 respectively. The final fitness score is computed as the mean $FFP4$ for all images $q \in D$.

As Fan et al. [12] explain, FFP4 is a utility function based on the idea that the utility of a relevant document decreases with its ranking order. More formally, we need a utility function $U(x)$ which satisfies the condition $U(x_1) > U(x_2)$ for two ranks x_1 and x_2 which $x_1 < x_2$. Although there are many possible functions $U(x)$, we decided to use FFP4 as it presents good results in previous works [48] applying this measure on similar evolutionary approaches that address rank-based tasks. Fan et al. report that this function and its associated parameters were chosen after exploratory data analysis.

For the baseline WTA-AE, we set the parameters as: number of hidden representation units $N_u = 1024$ and *winner-take-all* lifetime sparsity $k = 40\%$ empirically selecting them within the ranges {64, 128, 256, 512, 1024} and {5%, 10%, 20%, 30%, 40%, 50%, 80%}, respectively.

5.4 Evaluation Metrics

5.4.1 Precision-Recall Curves

The most traditional measures to evaluate retrieval effectiveness over a set of queries are Precision and Recall [1]. Precision measures the proportion of relevant

images regarding the answer set, while Recall measures the proportion of relevant images retrieved in the answer set regarding all relevant images existing in the database.

A perfect system would provide a Precision equal to 1 (all the retrieved images are relevant) and a Recall also equal to 1 (all the relevant images were retrieved). In practice, there is an inverse relationship between them: the more items the system returns, the higher the likelihood that relevant documents will be retrieved (increasing recall). However, this comes at the cost of also retrieving many irrelevant documents (decreasing precision). Therefore, in general, it is necessary to define a compromise between them.

In our case, we chose a measurement that considers Precision and Recall as functions of each other, generating interpolated Precision-Recall curves (11 points) whose the precision points P given by

$$P(r_i) = \max_{\forall j | r_i \leq r_j} P(r_j) \tag{6}$$

where $i, j \in 0, 1, ..., 10$ represent recall levels.

In order to evaluate the retrieval effectiveness over a set of query images Q, an averaged Precision-Recall curve is computed according to

$$\overline{P}(r_i) = \sum_{q=1}^{|Q|} \frac{1}{|Q|} P_q(r_i) \tag{7}$$

where P_q corresponds to the precision of the q-th query image.

5.4.2 MAP: Mean Average Precision

In some cases, the Precision-Recall curves appear occluded or inter-crossed, restraining a proper visual comparison. Because of the compromise between Precision and Recall, it is possible to employ a combination of the two measures as a single metric. This is the case of Mean Average Precision [1] which provides a convenient measure to quantitatively compare Precision-Recall curves and is defined as

$$MAP = \frac{1}{|Q|} \sum_{q=1}^{|Q|} AP_q \tag{8}$$

$$AP_q = \frac{1}{|R_q|} \sum_{k=1}^{|R_q|} P(R_q[k]) \tag{9}$$

where R_q is the set of relevant images in the dataset Q for each image q.

5.4.3 P@10

As observed on real-world applications of CBIR, the user gives prior attention for a small group of the top answers, corresponding to the first page of results, usually preferring to reformulate the query instead of checking the next pages. The *Precision-Recall* curves and *MAP* do not provide an adequate measurement for the effectiveness of these top results as they generally consider longer portions of the ranking. In order to address this issue, we also measured the precision at the top-10 results (P@10) [1].

Due to the proximity of some measures and aiming to provide accurate comparisons between the methods and its baselines, we used the Student's Paired t-Test [18] (p-value < 0.05) to statistically verify the results of Precision-Recall, MAP, and P@10.

5.4.4 Representation Size

In order to evaluate the descriptions dimensionality and possibly detect occurrence compactness regarding the previous methods, we measured the representation size, defined as the total number of bins that compose the histogram representations.

5.5 Experimental Protocol

In order to evaluate the proposed method, we conducted a k-fold cross-validation. According to this protocol, the dataset is randomly split into k mutually exclusive samples subset (folds) of approximated size. Then, the $k-1$ subsets are chosen as training set, and the remaining one as test set. The execution is repeated k times, and for each time, a different subset (without replacement) is chosen as the current test set and the remaining compose the training set.

We carried out all experiments considering $k = 5$ folds. As a consequence, for each experiment, the method was executed 5 times using 80% of the dataset as training set and 20% as test set.

6 Results and Discussion

This section compares the results of the proposed methods and baselines according to the evaluation measures.

First, we present the results of the UA methods with regard to Precision-Recall (Figs. 13 and 15), P@10 (Figs. 14a and 16a), MAP (Figs. 14b and 16b), and representation size (Fig. 17) for all datasets and feature extractors. Next, we present charts comparing the SCA results for Precision-Recall (Figs. 19-22), P@10 (Figs. 23 and 24), MAP (Figs. 25 and 26), and representation size (Fig. 18). In the figures, the symbols above each pair of measures indicate whether the proposed method yields statistically better ⊕, worse ⊖, or similar ⊜ results to those observed for the baselines (the minimum between BIC/GCH and WTA-AE), considering rejection of the null hypothesis when p-value < 0.05.

The following sections present and discuss the experimental results and provides comparisons between these two proposed approaches and baselines.

6.1 Unconstrained Approach

Observing the Precision-Recall curves for the BIC feature extractor (Fig. 13), the UA outperforms its baselines for all datasets. According to the P@10 measurements (Fig. 14a), the method also presents, on average, more relevant results in the first positions of the ranking for all datasets. The superior MAP results (Fig. 14b) confirm the superiority of UA, as this measure takes into account the performance of the evaluated methods for the whole Precision-Recall curve.

Similar results were observed when the GCH feature extractor is considered. Figs. 15, 16a, and 16b provide the effectiveness results in terms of Precision-Recall, MAP, and P@10, respectively. For all datasets, but for the *Supermarket Produce*, the proposed UA approach yielded better results than those of the baseline. For the *Supermarket Produce* dataset, no statistical difference was observed.

With regard to the representation sizes (Fig. 17), the differences between the proposed methods and the baselines are very high. Comparing to the feature extractors, the representations produced by our method approach were, on average, around 521% larger for BIC (Fig. 17a) and around 328% larger for GCH (Fig. 17b). A possible reason relies on the fact that the fitness function used for evaluating the genetic algorithm individuals prioritizes the representation effectiveness performance on the retrieval task, i.e., the optimization process is not guided to guarantee compact representations. In this scenario, the proposed method quantized more regions in the color space, leading to representation with higher dimensions. The Size-Constrained Approach (SCA), whose results are discussed next, addresses this issue.

6.2 Size-Constrained Approach

In the evaluation of the SCA approach, we varied the number of bins in the ranges $\{16, 32, 64, 96, 128, 256, 384\}$ and $\{8, 16, 32, 48, 64, 128, 192\}$ for BIC and GCH approaches, respectively. These ranges were defined based on a logarithmic sequence of proportions (12,5%, 25%, 50%, 100%, 200%) of the baselines vector sizes and some additional points among them (75% and 300%) to provide a clearer view of the performances behaviour. Figs. 19 and 20 present the Precision-Recall curves for the BIC-based approaches and WTA-AE for all datasets, considering these different feature vector sizes. We can observe that the proposed method yielded comparable or better results than those observed for the baselines for feature vectors whose size is higher than 96 for the majority of the datasets. In fact, the smaller the feature vector size, the worse the results of SCA when compared to the baselines. Similar results were observed for the GCH-based approaches at Figs. 21 and 22.

Figs. 23 and 24 provide the P@10 results for the SCA method when compared with baselines for both BIC and GCH description approaches, respectively. Figs. 25 and 26, in turn, provide the MAP results for both BIC and GCH description approaches, respectively. Results related to MAP and P@10 demonstrated that, regardless the feature extraction method considered, the use of the SCA approach

is able to create quite effective description approaches, without a high cost in terms of storage requirements, i.e., in terms of the feature vector size.

Figure 18 shows the sizes of the produced representations given the respective size upper-bounds. For the BIC approach (Fig. 18a) representations whose size reached or were very close to the imposed upper-bound were produced, showing a tendency for generating quantizations with strong tonality detailing. In contrast, the results for the GCH approach (Fig. 18b) are quite different. For example, for the upper limits 128 and 192, the produced representations were considerably smaller than the maximum size. In these cases, the more effective representations are not necessarily the ones with the highest possible dimensionality. This finding means that increasing the number of tonalities does not necessarily lead to performance improvements. In other words, the proposed methods are able to generate representations that are significantly smaller than the predefined upper-bound but with high effectiveness.

7 Conclusions

We proposed two approaches of a representation learning method, which intends to provide more effective and compact image representations by optimizing the colour quantization for the image domain. We performed experiments on eight different image datasets comparing the results with a pre-defined quantization approach and a Sparse Autoencoder in terms of effectiveness performance on content-based image retrieval tasks. Methods are also evaluated in terms of the representation dimensionality.

The first approach, named Unconstrained approach, produced representations that outperformed the hand-crafted baselines in terms of effectiveness but presented feature vectors with several times higher dimensionality. It also outperformed the effectiveness of the autoencoder representation, but presenting intensively lower sizes. The second approach, which imposes a limitation on the representation dimension (Size-Constrained approach), presented, in general, better effectiveness results for the same dimensionality (e.g., 128 bins). In other situations, this approach reduced the representation size up to 50%, maintaining statistically comparable performance to the hand-crafted baselines. Finally, the SCA approach also produced results that imposed a reduction of more than 75% of the storage requirements, but presented poor effectiveness performance, showing the existence of a trade-off between compactness and effectiveness.

Since the representations are based on color histograms, the over- and the subsampling of determined color space regions allows for the identification of more effective representations and, consequently, improvements on the search performance. Furthermore, a domain-oriented quantization allows for discarding the less contributing tonalities resulting in a possible reduction of the representation size.

In the end, the results confirm our hypothesis, for the tested scenarios, that it was possible to produce more effective and compact fitness by exploring a colour quantization optimized for the image domain. Moreover, our method is capable of improving already existent feature extraction methods by providing descriptions more effective in terms of representation quality and more compact according to a parametric upper bound. This research, therefore, opens novel opportunities for future investigation. We plan to assess the effects of the proposed quantization

approches to other image processing applications such as image classification [24], image segmentation [13] and image dehazing [67]. We also plan to investigate the impact of the resulting quantization when combined with deep-learning-based feature extractors.

Acknowledgments

This study was financed in part by: the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* - Brasil (CAPES) - Finance Code 001; the Brazilian National Council for Scientific and Technological Development (CNPq) - grants #424700/2018-2 and #311395/2018-0; and the Minas Gerais Research Foundation (FAPEMIG) - grant APQ-00449-17. Authors are also grateful to CAPES (grant #88881.145912/2017-01), São Paulo Research Foundation – FAPESP (grants #2014/12236-1, #2015/24494-8, #2016/50250-1, and #2017/20945-0) and the FAPESP - Microsoft Virtual Institute (grants #2013/50155-0, #2013/50169-1, and #2014/50715-9).

References

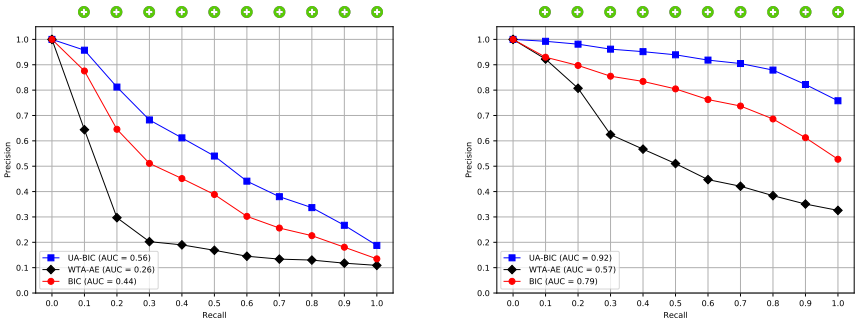
1. Baeza-Yates RA, Ribeiro-Neto B (1999) Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA
2. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828
3. Bengio Y, Courville A, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35(8):1798–1828
4. Bharti V, Biswas B, Shukla KK (2020) Recent trends in nature inspired computation with applications to deep learning. In: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, pp 294–299
5. Bhunia AK, Bhattacharyya A, Banerjee P, Roy PP, Murala S (2019) A novel feature descriptor for image retrieval by combining modified color histogram and diagonally symmetric co-occurrence texture pattern. Pattern Analysis and Applications pp 1–21
6. Bo L, Ren X, Fox D (2011) Hierarchical matching pursuit for image classification: Architecture and fast algorithms. Advances in neural information processing systems pp 2115–2123
7. Bukh PND (1992) The art of computer systems performance analysis, techniques for experimental design, measurement, simulation and modeling
8. Coates A, Ng AY (2011) The importance of encoding versus training with sparse coding and vector quantization. Proceedings of the 28th International Conference on Machine Learning (ICML-11) pp 921–928
9. Criminisi A (2004) Microsoft research cambridge object recognition image database. available online: https:
10. Davis L (1991) Handbook of genetic algorithms

11. Davis SM, Landgrebe DA, Phillips TL, Swain PH, Hoffer RM, Lindenlaub JC, Silva LF (1978) Remote sensing: the quantitative approach. New York, McGraw-Hill International Book Co, 1978 405 p
12. Fan W, Fox EA, Pathak P, Wu H (2004) The effects of fitness functions on genetic programming-based ranking discovery for web search. Journal of the American Society for Information Science and Technology 55(7):628–636
13. García-Lamont F, Cervantes J, López-Chau A, Ruiz-Castilla S (2020) Color image segmentation using saturated rgb colors and decoupling the intensity from the hue. Multimedia Tools and Applications 79(1-2):1555–1584
14. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning, 1989. Reading: Addison-Wesley
15. Hinton GE, Zemel RS (1994) Autoencoders, minimum description length and helmholtz free energy. In: Advances in neural information processing systems, pp 3–10
16. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural computation 18(7):1527–1554
17. Khaldi B, Aiadi O, Kherfi ML (2019) Combining colour and grey-level co-occurrence matrix features: a comparative study. IET Image Processing 13(9):1401–1410
18. Kim TK (2015) T test as a parametric statistic. Korean journal of anesthesiology 68(6):540–546
19. Kingma DP, Welling M (2013) Auto-encoding variational bayes. arXiv preprint arXiv:13126114
20. Krizhevsky A, Hinton G, et al (2009) Learning multiple layers of features from tiny images
21. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
22. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
23. Leibe B, Schiele B (2003) Analyzing appearance and contour based methods for object categorization. Computer Vision and Pattern Recognition, 2003 Proceedings 2003 IEEE Computer Society Conference on 2:II–409
24. Li T, Leng J, Kong L, Guo S, Bai G, Wang K (2019) Dcnr: deep cube cnn with random forest for hyperspectral image classification. Multimedia Tools and Applications 78(3):3411–3433
25. Li X, Li D, Peng L, Zhou H, Chen D, Zhang Y, Xie L (2019) Color and depth image registration algorithm based on multi-vector-fields constraints. Multimedia Tools Appl 78(17):24,301–24,319, DOI 10.1007/s11042-018-7048-4, URL `https://doi.org/10.1007/s11042-018-7048-4`
26. Li Y (2005) Object and concept recognition for content-based image retrieval. PhD thesis, University of Washington, Seattle, WA, USA
27. Li Y, Shapiro LG (2002) Consistent line clusters for building recognition in cbir. Proceedings of the International Conference on Pattern Recognition
28. Lu D, Weng Q (2007) A survey of image classification methods and techniques for improving classification performance. International journal of Remote sensing 28(5):823–870
29. Luccheseyz L, Mitray S (2001) Color image segmentation: A state-of-the-art survey. Proceedings of the Indian National Science Academy (INSA-A)

67(2):207–221
30. Makhzani A, Frey B (2013) K-sparse autoencoders. arXiv preprint arXiv:13125663
31. Makhzani A, Frey BJ (2015) Winner-take-all autoencoders. In: Advances in neural information processing systems, pp 2791–2799
32. Mohseni SA, Wu HR, Thom JA, Bab-Hadiashar A (2020) Recognizing induced emotions with only one feature: A novel color histogram-based system. IEEE Access 8:37,173–37,190
33. Nakamura R, Fonseca L, dos Santos JA, Torres RdS, Yang XS, Papa JP (2014) Nature-inspired framework for hyperspectral band selection. IEEE Transactions on Geoscience and Remote Sensing 52(4):2126–2137
34. Nayar SK, Nene SA, Murase H (1996) Real-time 100 object recognition system. Proceedings of IEEE International Conference on Robotics and Automation 3:2321–2325 vol.3
35. Ng A, et al (2011) Sparse autoencoder. CS294A Lecture notes 72(2011):1–19
36. Nogueira K, Penatti OA, dos Santos JA (2017) Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognition 61:539 – 556
37. Oh IS, Lee JS, Moon BR (2004) Hybrid genetic algorithms for feature selection. IEEE Transactions on Pattern Analysis & Machine Intelligence (11):1424–1437
38. Omran MG, Engelbrecht AP, Salman A (2005) A color image quantization algorithm based on particle swarm optimization. Informatica 29(3)
39. Penatti OA, Valle E, Torres RdS (2012) Comparative study of global color and texture descriptors for web image retrieval. Journal of Visual Communication and Image Representation
40. Penatti OAB, d S Torres R (2008) Color descriptors for web image retrieval: A comparative study. 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing pp 163–170
41. Pérez-Delgado ML (2019) The color quantization problem solved by swarm-based operations. Applied Intelligence 49(7):2482–2514
42. Ponti M, Nazaré TS, Thumé GS (2016) Image quantization as a dimensionality reduction procedure in color and texture feature extraction. Neurocomputing 173:385–396
43. Ranzato M, Poultney C, Chopra S, Cun YL (2007) Efficient learning of sparse representations with an energy-based model. In: Advances in neural information processing systems, pp 1137–1144
44. Rifai S, Vincent P, Muller X, Glorot X, Bengio Y (2011) Contractive auto-encoders: Explicit invariance during feature extraction
45. Rocha A, Hauagge DC, Wainer J, Goldenstein S (2010) Automatic fruit and vegetable classification from images. Computers and Electronics in Agriculture 70(1):96–104
46. Rodriguez-Coayahuitl L, Morales-Reyes A, Escalante HJ (2019) Evolving autoencoding structures through genetic programming. Genetic Programming and Evolvable Machines 20(3):413–440
47. da S Torres R, Falcão AX (2006) Content-based image retrieval: Theory and applications. Revista de Informática Teórica e Aplicada (RITA) 13(2):161–185
48. da S Torres R, Falcão AX, Gonçalves MA, Papa JP, Zhang B, Fan W, Fox EA (2009) A genetic programming framework for content-based image retrieval. Pattern Recognition 42(2):283 – 292

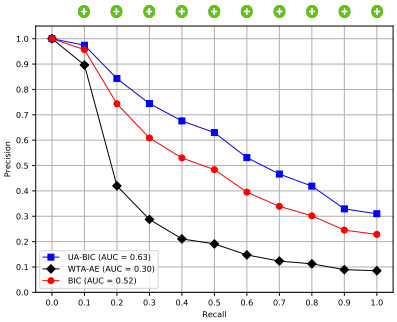
49. da S Torres R, Falcão AX, Gonçalves MA, Papa JP, Zhang B, Fan W, Fox WA (2009) A genetic programming framework for content-based image retrieval. Pattern Recognition 42(2):283 – 292
50. Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. Artificial Intelligence and Statistics pp 448–455
51. dos Santos JA, Penatti OAB, da Silva Torres R (2010) Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. VISAPP (2) pp 203–208
52. Scheunders P (1996) A genetic lloyd-max image quantization algorithm. Pattern Recognition Letters 17(5):547–556
53. Sheng T, Feng C, Zhuo S, Zhang X, Shen L, Aleksic M (2018) A quantization-friendly separable convolution for mobilenets. In: 2018 1st Workshop on Energy Efficient Machine Learning and Cognitive Computing for Embedded Applications (EMC2), IEEE, pp 14–18
54. Smeulders AW, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Transactions on pattern analysis and machine intelligence 22(12):1349–1380
55. Stehling RO, Nascimento MA, Falcão AX (2002) A compact and efficient image retrieval approach based on border/interior pixel classification. International Conference on Information and Knowledge Management pp 102–109
56. Suganuma M, Shirakawa S, Nagao T (2017) A genetic programming approach to designing convolutional neural network architectures. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp 497–504
57. Swain MJ, Ballard DH (1991) Color indexing. International journal of computer vision 7(1):11–32
58. Vincent P, Larochelle H, Bengio Y, Manzagol PA (2008) Extracting and composing robust features with denoising autoencoders. Proceedings of the 25th international conference on Machine learning pp 1096–1103
59. Vincent P, Larochelle H, Lajoie I, Bengio Y, Manzagol PA (2010) Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research 11(Dec):3371–3408
60. Wang JZ, Li J, Wiederhold G (2001) Simplicity: Semantics-sensitive integrated matching for picture libraries. IEEE Transactions on pattern analysis and machine intelligence 23(9):947–963
61. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemometrics and intelligent laboratory systems 2(1-3):37–52
62. Xie L, Yuille A (2017) Genetic cnn. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 1388–1397
63. Yang Y, Newsam S (2010) Bag-of-visual-words and spatial extensions for land-use classification. Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems pp 270–279
64. Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. Acm computing surveys (CSUR) 38(4):13
65. Yu K, Lin Y, Lafferty J (2011) Learning image representations from the pixel level via hierarchical sparse coding. Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on pp 1713–1720
66. Zeng S, Huang R, Wang H, Kang Z (2016) Image retrieval using spatiograms of colors quantized by gaussian mixture models. Neurocomputing 171:673–684

67. Zhang S, He F (2019) Drcdn: learning deep residual convolutional dehazing networks. The Visual Computer pp 1–12
68. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Computer Vision and Pattern Recognition, pp 6848–6856

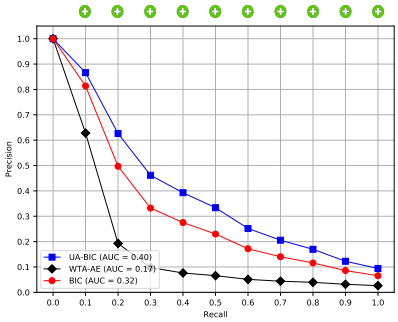


(a) Groundtruth

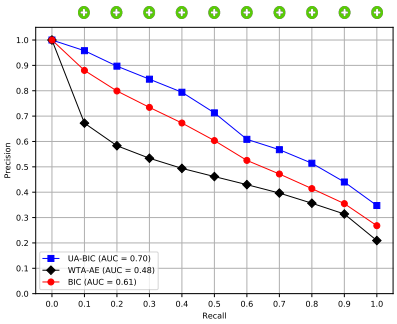
(b) Coil-100



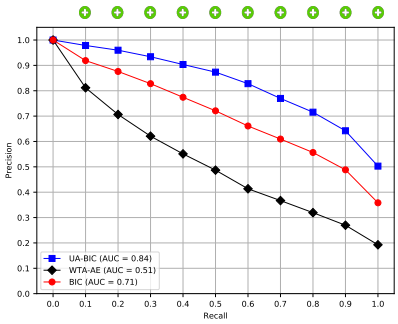
(c) Corel-1566



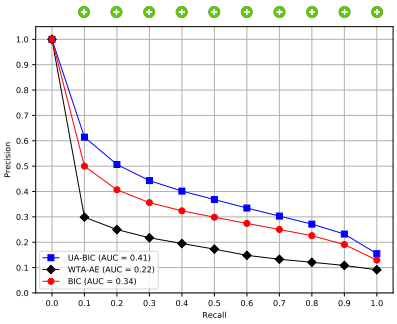
(d) Corel-3906



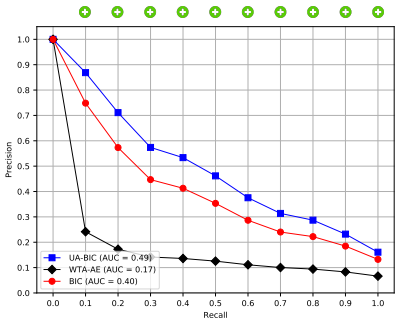
(e) ETH-80



(f) Supermarket P.

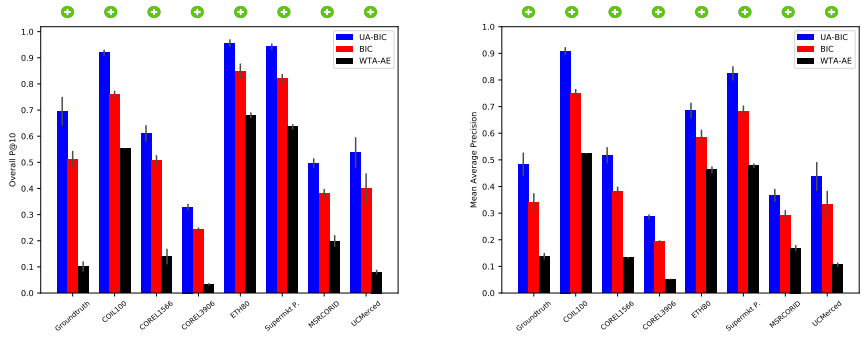


(g) MSRCORID



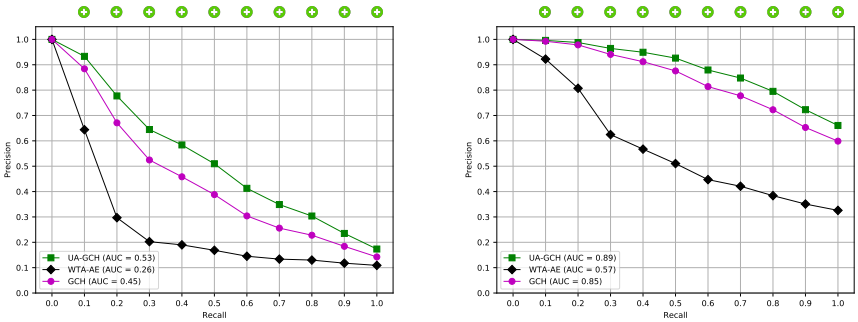
(h) UCMerced Land-use

Fig. 13: Comparison between the Precision-Recall Curves of the UA method, WTA Autoencoder and the BIC feature extractor.



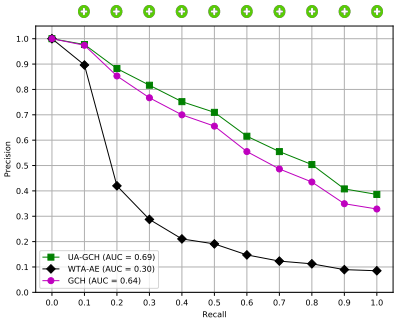
(a) P@10 (b) MAP

Fig. 14: Comparison between the (a) P@10 and (b) MAP results of UA, WTA Autoencoder and the **BIC** feature extractor.

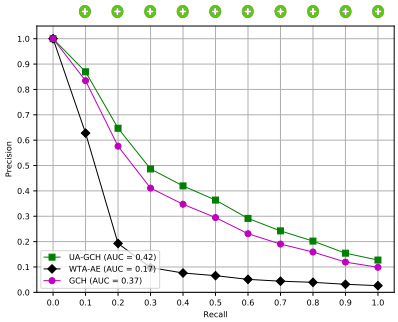


(a) Groundtruth

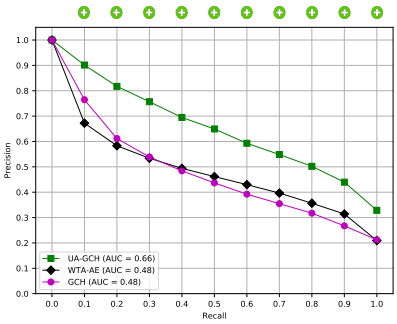
(b) Coil-100



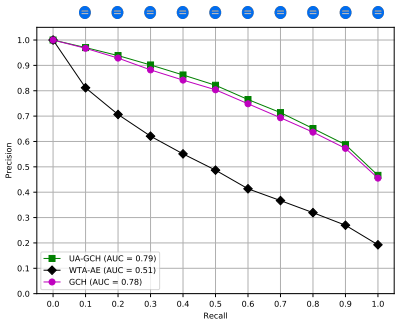
(c) Corel-1566



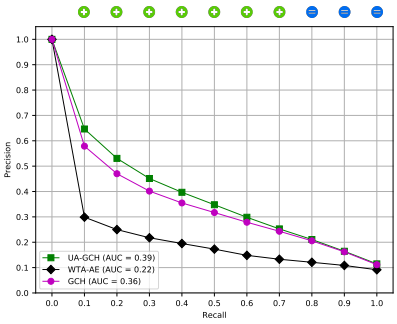
(d) Corel-3906



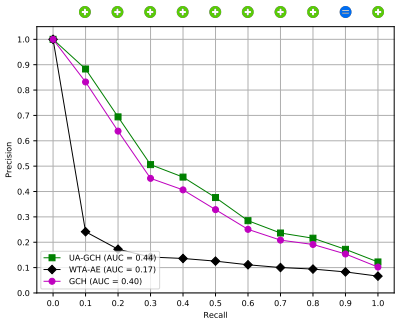
(e) ETH-80



(f) Supermarket P.

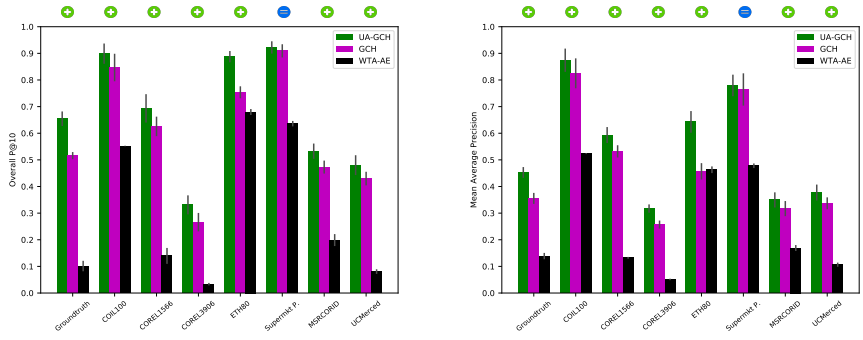


(g) MSRCORID



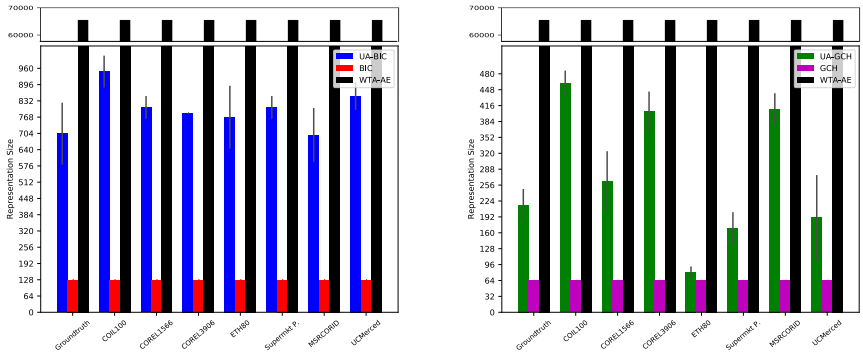
(h) UCMerced Land-use

Fig. 15: Comparison between the Precision-Recall curves of UA, WTA Autoencoder and GCH feature extractor.



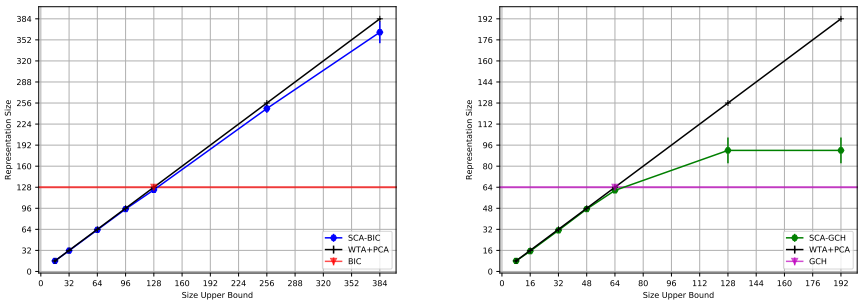
(a) P@10 (b) MAP

Fig. 16: Comparison between the (a) P@10 and (b) MAP results of UA, WTA Autoencoder and the **GCH** feature extractor.



(a) BIC (b) GCH

Fig. 17: Comparison between the representation size results of UA, WTA Autoencoder and the feature extractors: (a) BIC and (b) GCH. The upper windows show a cut of the highest columns while the lower windows show a view of the bottom.



(a) BIC (b) GCH

Fig. 18: Comparison between the representation size results of SCA, WTA Autoencoder and the feature extractors: (a) BIC and (b) GCH, for the ETH-80 dataset. The supplementary material of this article contains the same comparison for the remaining datasets.

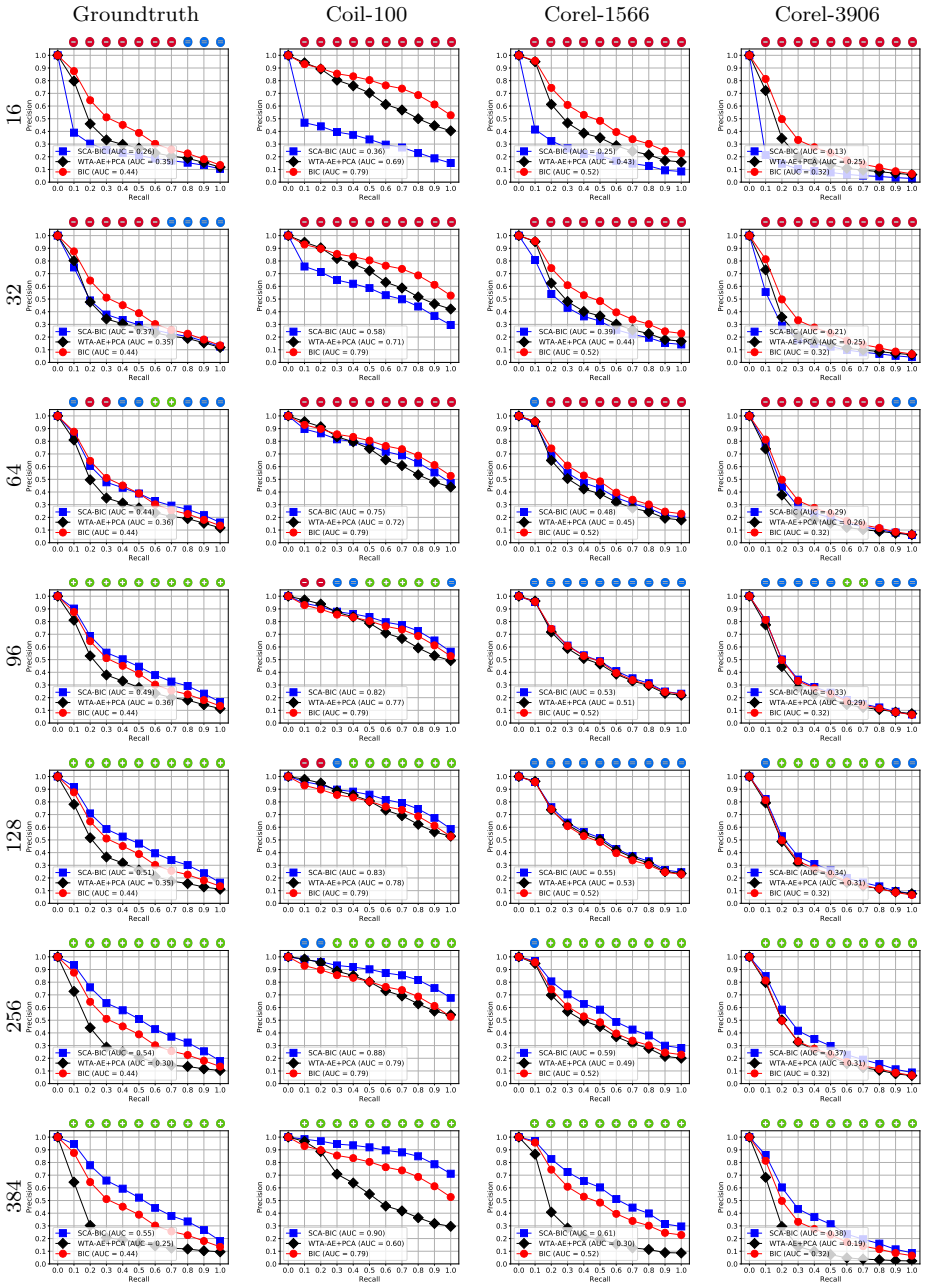


Fig. 19: Comparison between the Precision-Recall curves of SCA, WTA Autoencoder and BIC feature extractor considering all representation size limits for the datasets *Groundtruth*, *Coil-100*, *Corel-1566*, and *Corel-3906*. We recommend colourful printing for adequate visualization.

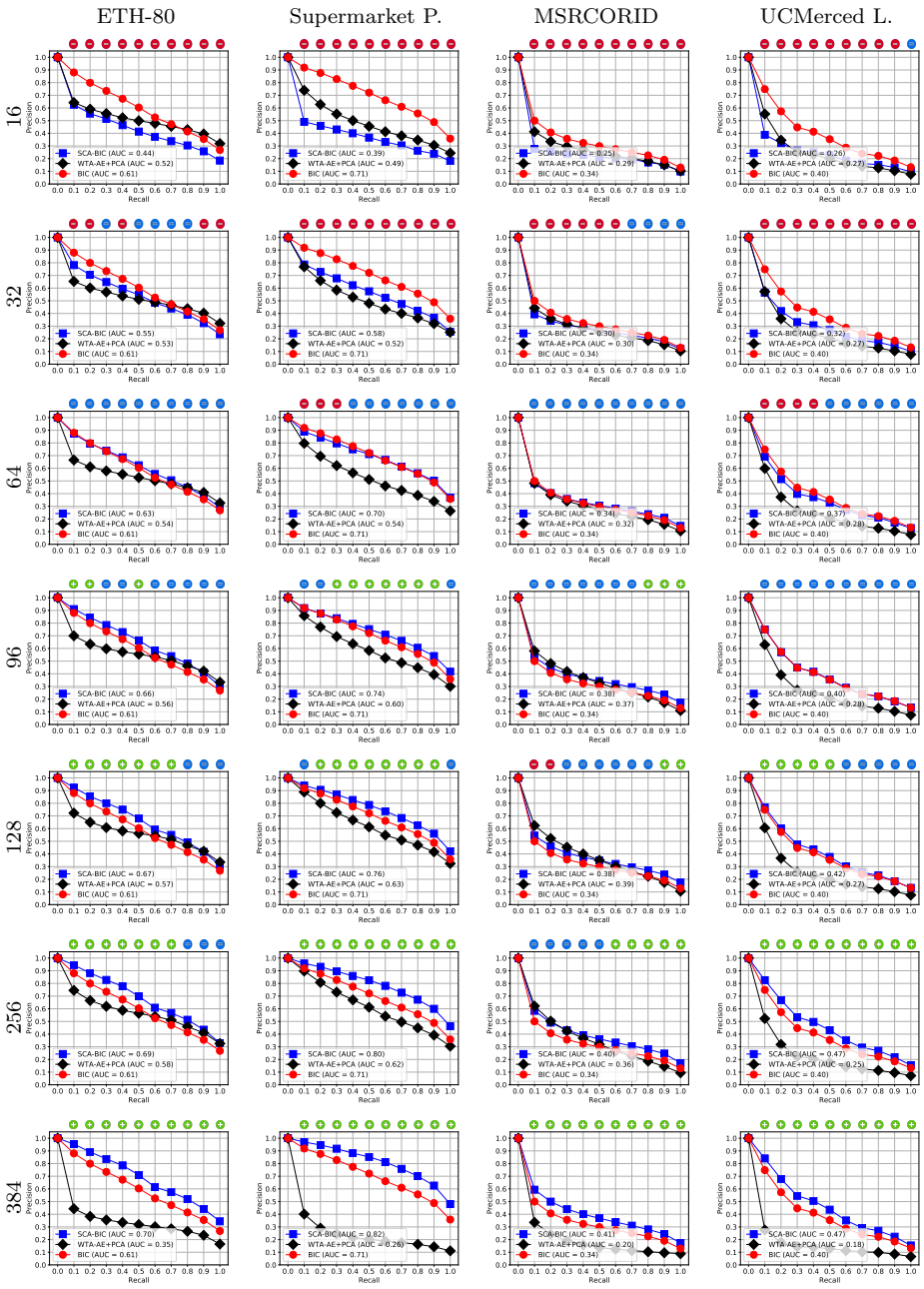


Fig. 20: Comparison between the Precision-Recall curves of SCA, WTA Autoencoder and BIC feature extractor considering all representation size limits for the datasets *ETH-80*, *Supermarket Produce*, *MSRCORID*, and *UCMerced Landuse*. We recommend colourful printing for adequate visualization.

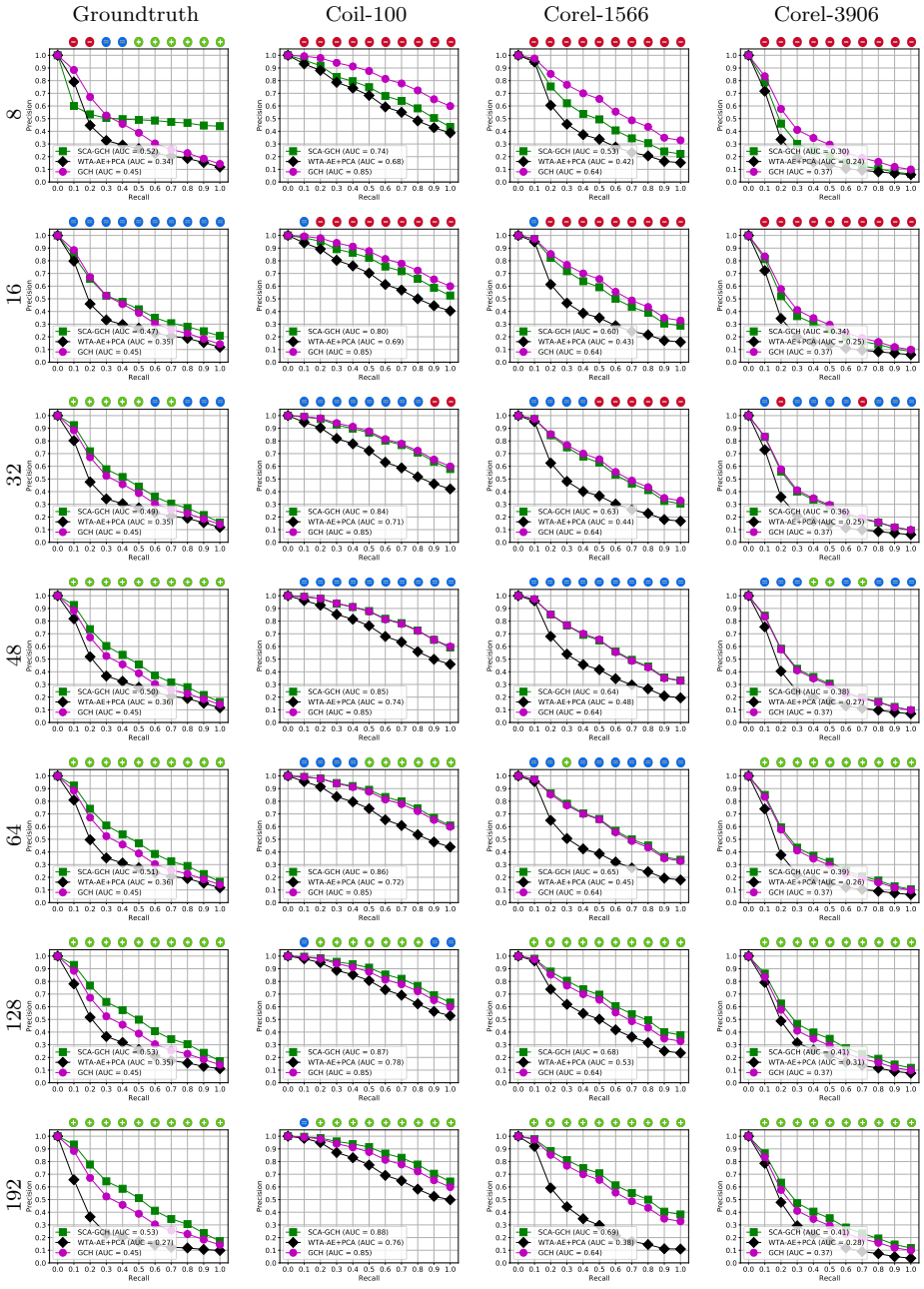


Fig. 21: Comparison between the Precision-Recall curves of SCA, WTA Autoencoder and GCH feature extractor considering all representation size limits for the datasets *Groundtruth*, *Coil-100*, *Corel-1566*, and *Corel-3906*. We recommend colourful printing for adequate visualization.

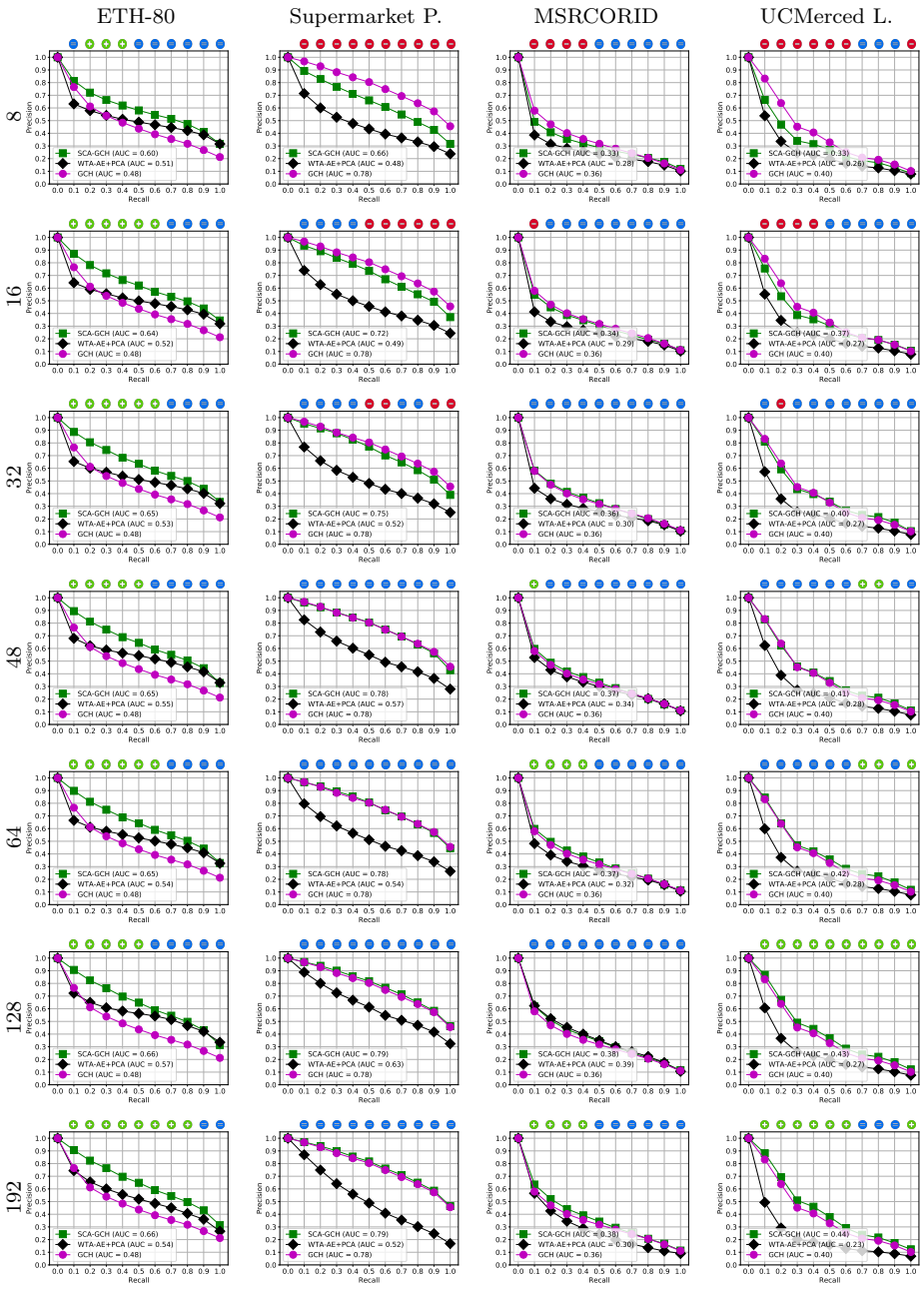
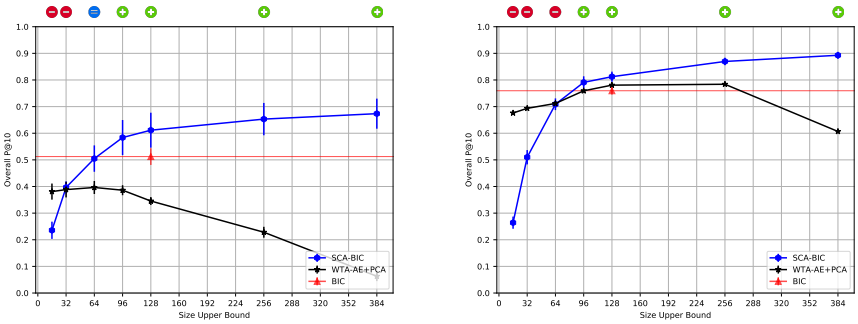
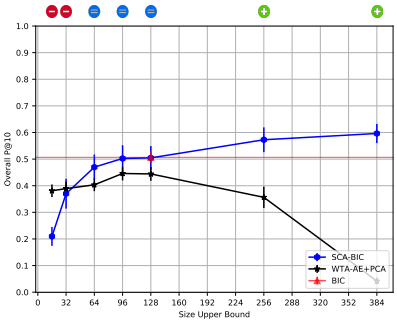


Fig. 22: Comparison between the Precision-Recall curves of SCA, WTA Autoencoder and GCH feature extractor considering all representation size limits for the datasets *ETH-80*, *Supermarket Produce*, *MSRCORID*, and *UCMerced Landuse*. We recommend colourful printing for adequate visualization.

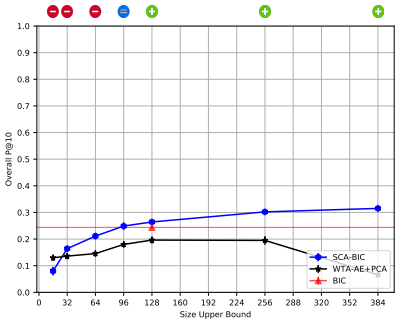


(a) Groundtruth

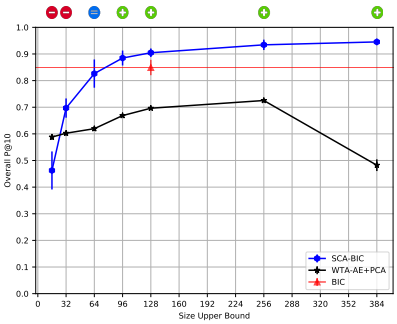
(b) Coil-100



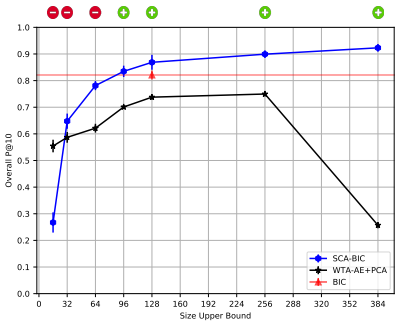
(c) Corel-1566



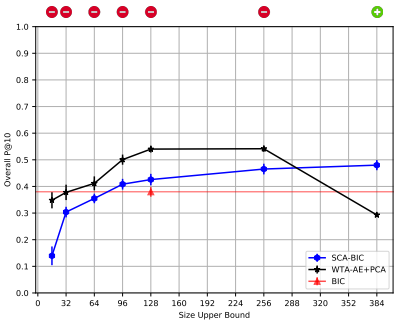
(d) Corel-3906



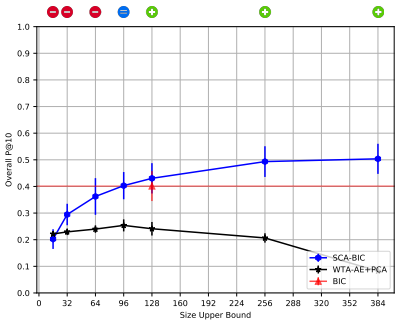
(e) ETH-80



(f) Supermarket Produce

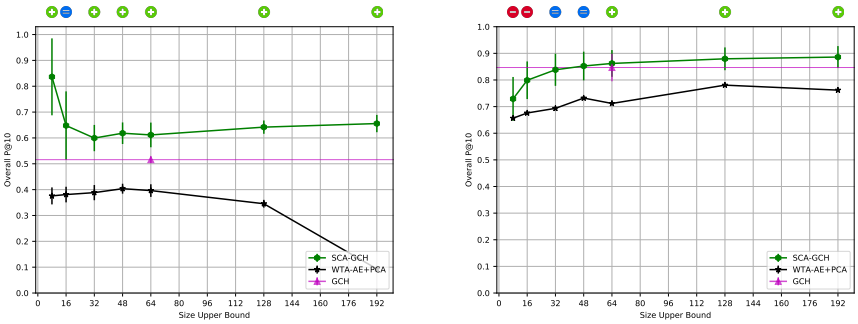


(g) MSRCORID



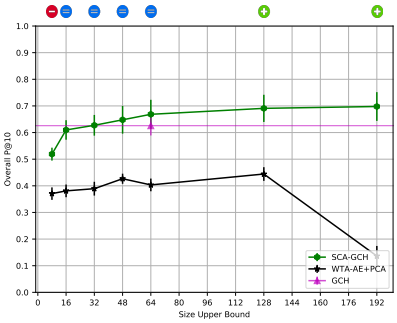
(h) UCMerced Land-use

Fig. 23: Comparison between the P@10 results of SCA, WTA Autoencoder and BIC feature extractor

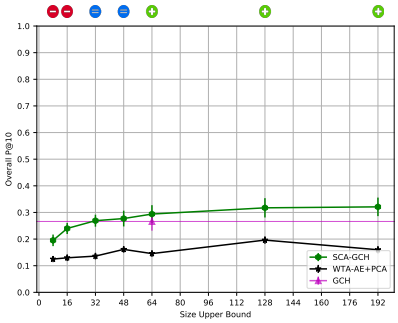


(a) Groundtruth

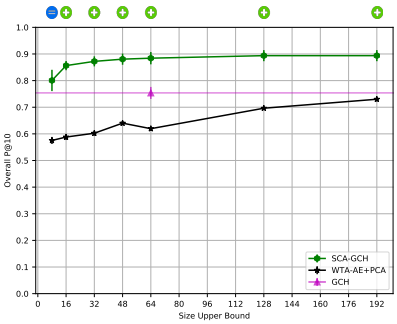
(b) Coil-100



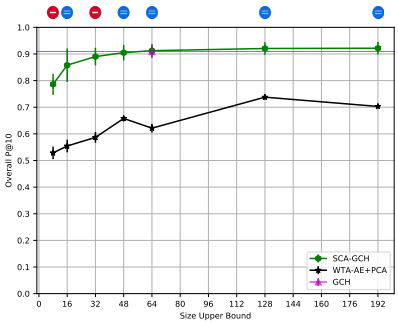
(c) Corel-1566



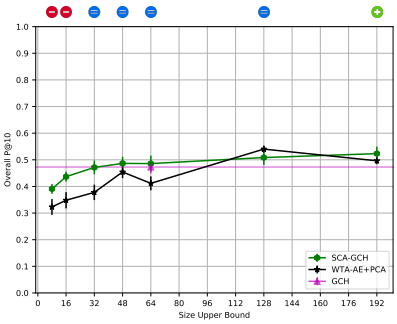
(d) Corel-3906



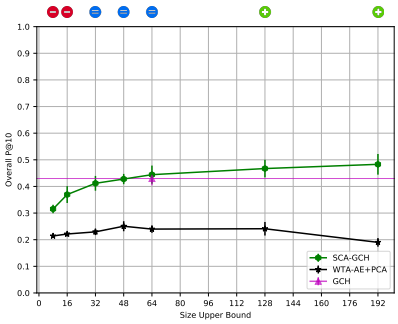
(e) ETH-80



(f) Supermarket Produce

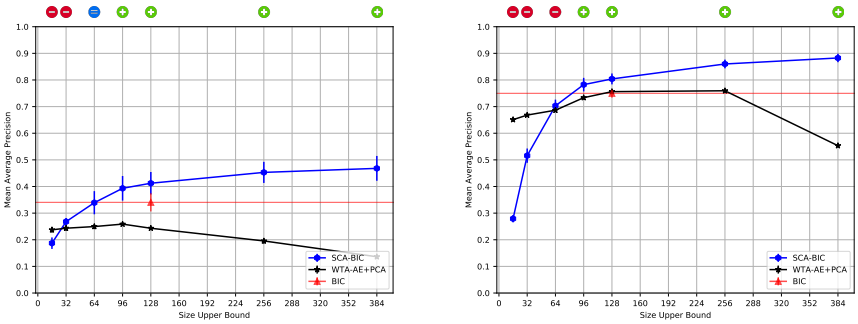


(g) MSRCORID



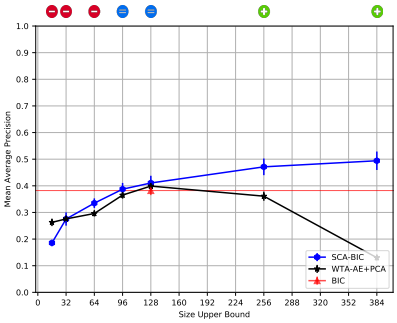
(h) UCMerced Land-use

Fig. 24: Comparison between the P@10 results of SCA, WTA Autoencoder and GCH feature extractor

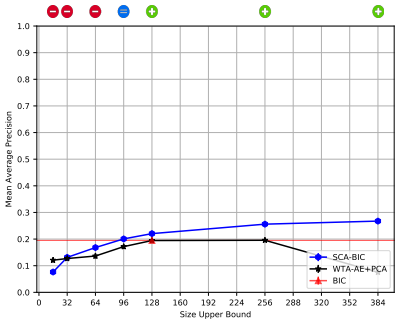


(a) Groundtruth

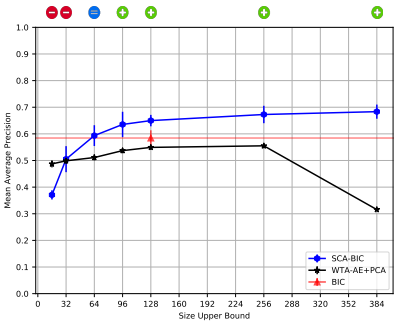
(b) Coil-100



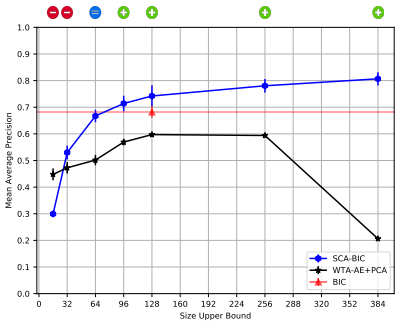
(c) Corel-1566



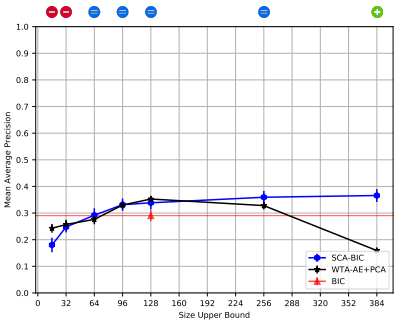
(d) Corel-3906



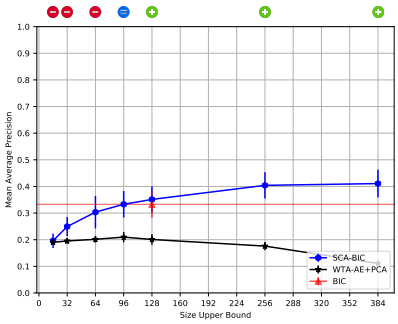
(e) ETH-80



(f) Supermarket Produce

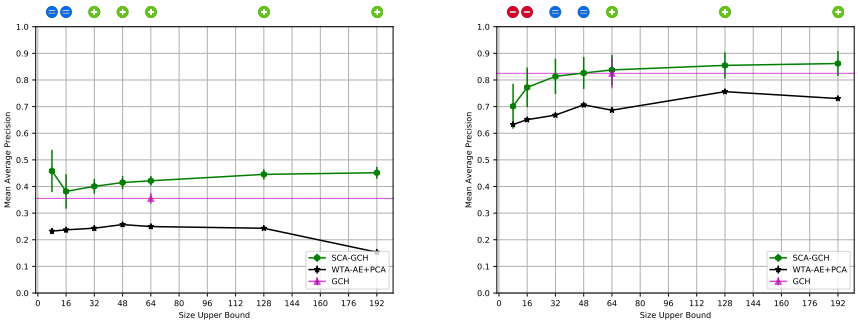


(g) MSRCORID

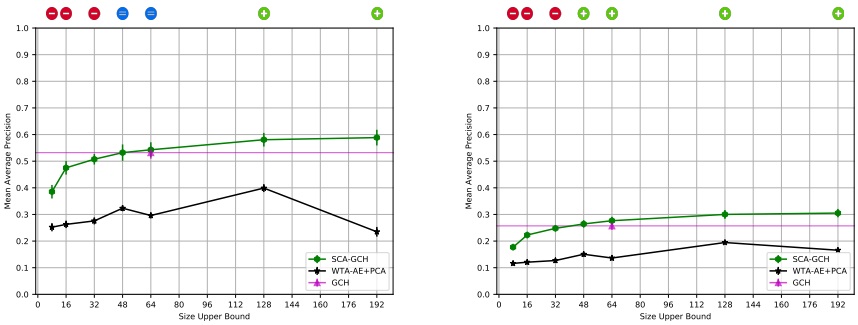


(h) UCMerced Land-use

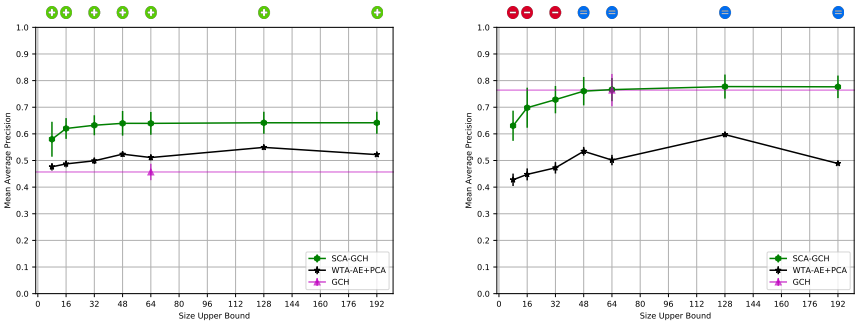
Fig. 25: Comparison between the MAP results of SCA, WTA Autoencoder and BIC feature extractor



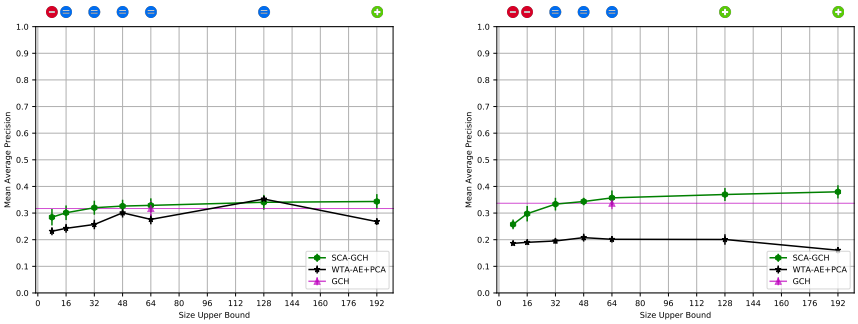
(a) Groundtruth (b) Coil-100



(c) Corel-1566 (d) Corel-3906



(e) ETH-80 (f) Supermarket Produce



(g) MSRCORID (h) UCMerced Land-use

Fig. 26: Comparison between the MAP results of SCA, WTA Autoencoder and GCH feature extractor