2019

Composite description based on Salient Contours and Color information for CBIR tasks

Iakovidou, C.

Institute of Electrical and Electronics Engineers (IEEE)

Composite description based on Salient Contours and Color information for CBIR tasks

C. Iakovidou, N. Anagnostopoulos, M. Lux, K. Christodoulou, Y. Boutalis and S. A. Chatzichristofis

***Abstract*—This paper introduces a novel image descriptor for content based image retrieval tasks that integrates contour and color information into a compact vector. Loosely inspired by the human visual system and its mechanisms in efficiently identifying visual saliency, operations are performed on a fixed lattice of discrete positions by a set of edge detecting kernels that calculate region derivatives at different scales and orientation. The description method utilizes a weighted edge histogram where bins are populated on the premise of whether the regions contain edges belonging to the salient contours, while the discriminative power is further enhanced by integrating regional quantized color information. The proposed technique is both efficient and adaptive to the specifics of each depiction, while it does not need any training data to adjust parameters.**

Experimental evaluation conducted on seven benchmarking datasets against 13 well known global descriptors along with SIFT, SURF implementations (both in VLAD and BOVW), highlight the effectiveness and efficiency of the proposed descriptor.

***Index Terms*—Image Retrieval, Multimedia Retrieval, Global Features.**

I. Introduction

CONTENT -based image retrieval (CBIR) is one of the fundamental research challenges extensively studied by the multimedia community for decades, due to its wide range of applications in information retrieval and computer/robotic vision systems. The term "content-based" refers to the fact that the indexing and searching mechanisms depend on information derived from features extracted by the image itself like, texture, color, shape, etc., rather than assigned text annotations. Thus, the goal of any CBIR system is to vectorize an image based on extracted features in a way that grasps its unique characteristics, and visual content.

A long debate has been going on concerning the most effective way to treat an image for indexing and retrieval [1], [2]. Several approaches for CBIR have been proposed and evaluated employing a wide spectrum of strategies from global to local features representations. Whether hand-crafted and tuned based on domain knowledge or learned directly from raw data, all attempts try to address a main challenge; i.e., narrowing the "semantic gap" that exists between low-level image pixels captured by devices, and high-level semantic concepts perceived by humans. Research conducted so far on retrieval methods suggests that each of the proposed approaches has its own benefits and certain limitations [3], [4] and despite the promising results reported, the underlying theoretical foundation does not yet clearly argue the conditions that characterise which strategy performs better or outperforms other approaches, or how to determine the optimal structure for a certain task [5].

In contrast, human vision has a remarkable proficiency and an unmatched adaptability when dealing with various computer vision tasks. Concerning image representation and understanding, humans' successful perception in categorizing images and spotting similarities between depictions begins by intuitively understanding the intention of the composition. Thus, from an abstract high-level understanding of the image, a human observer begins the process of interpretation by extracting rich information from images, effortlessly judging the saliency of image regions, with an attention to the important parts. More specifically, theories of human attention hypothesize that the human vision system only processes (salient) parts of an image in detail, while leaving others nearly unprocessed [6], [7]. Saliency originates from visual stimuli that instigates visual uniqueness, rarity or surprise, and is often attributed to variations in image attributes like color, gradient, edges, contours and boundaries [8].

Visual saliency has been successfully explored as an efficient pre-processing step for image segmentation, classification, and object recognition [9], [10], [11], [12]. In contrast to the image classification or object detection problems that consist of labeling input images with a probability of the presence of a particular visual object class, and an estimation of the object's position, CBIR is not always about direct object matching between query and dataset. A wide range of image retrieval tasks also aim to rank similar images to a given query by evaluating the scene. While research in scene understanding often treats objects as the atoms of scene recognition, behavioral experiments on fast scene perception suggest that most real-world scenes can be inferred from the arrangement of the basic geometrical forms, the spatial relationship between regions and blobs of a particular size, and aspect ration [13]. Moreover, other research [14], [15] suggests that even coarse color information in the representation significantly boosts fast scene recognition, while, in addition to colors, the configuration of contours can help predict presence or absence of objects in images and determine with a high probability basic-level classes of environmental scenes (e.g. forest, building, street), as well as global properties

Chryssanthi Iakovidou: This work was completed while affiliated with Democritus University of Thrace, Greece. During the review and at the time of acceptance the author is with the Information Technologies Institute, Centre for Research and Technology Hellas (CERTH), Greece.
Nektarios Anagnostopoulos: Microsoft, Prague, Czech Republic
Mathias Lux: Alpen-Adria-Universität Klagenfurt, Austria
Klitos Christodoulou: University of Nicosia, Cyprus
Yiannis Boutalis: Democritus University of Thrace, Greece
Savvas A. Chatzichristofis: Neapolis University Pafos, Cyprus

of the three-dimensional space (e.g. perspective, clutter) [13], [16], [17]. To that extent, global features that capture the diagnostic structure of an image based on visual saliency, can significantly improve the performance of retrieval tasks of natural-scenes with advanced computational efficiency.

To sum up, extracting rich information from images is a multifold problem thus, making sense of a depiction can potentially include many computations related to extracting, evaluating and forming feature representations of the contents. Even though significant advancements have been made possible by the broader image processing scientific community, combining different features and techniques remains a challenging process. A major challenge derives from estimating the best trade-off between a successful description and computational costs. Another challenge relates to the process of combining different image features into a single signature so that the similarity between vectors/images is measurable. In addition, as image collections are expanding, a CBIR system should be capable of undertaking and processing a massive load of already accumulated data that are dynamically populated by images that vary in their respective properties, such as size and resolution, color depth and encoding.

Considering the aforementioned raised topics and challenges, our design strategy is grounded on the following key properties:

(1) *produce a descriptor that looks for meaningful visual information globally;* the descriptor should be capable of differentiating between the salient parts from non-salient but account for both,
(2) *make the descriptor adaptive to the specific contents of the depictions in hand;* the manual tuning of free parameters is an impractical strategy, thus, an unsupervised adaptive descriptor is more advantageous when it comes to real life retrieval tasks, and
(3) *create an efficient descriptor;* specifically, by efficient we mean: (i) to reduce the computations required and the overall complexity to the minimum, (ii) to design an implementation that is parallel-execution friendly, and (iii) to consider the storage requirements and search time by keeping the descriptor compact.

In that sense, this work introduces a novel global image descriptor for image retrieval tasks that incorporates all the aforementioned desired properties ensuring, that the proposed method is both effective and efficient. Feature extraction is based on a weighted, quantized edge histogram where bins are populated on the premise of whether the image regions contain edges belonging to the salient contours of the depiction or not, loosely inspired by the human visual system and its operations in identifying visual saliency. To achieve a good level of discrimination the proposed method utilizes spatial information (i.e., image composition) in a multiscale search. Further, in order to complete our description the method employs quantized color information. The result of the description process is a composite and compact `120-bin` descriptor.

During the design of the methodology, a top priority was set to eliminate as many free parameters as possible, focusing instead on developing an adaptive, fully unsupervised algorithm that allows extraction of features and image description for a variety of images. In particular, through this work we contribute the following: (i) a fully unsupervised, parameter-free adaptive image descriptor; (ii) a novel feature extraction strategy which builds on weighted quantized edge histograms; and (iii) a large scale experimental evaluation on seven benchmarking datasets against 13 well known global descriptors.

II. Related Work

CBIR is a long studied field with a significant amount of contributions over the last two decades. In what follows we briefly revisit the related work organized by the type of features employed for CBIR, pointing out their scope, advantages and limitations.

Global features representations vectorize image contents by extracting features, such as, color, texture and shapes over the whole image. In order for the representation to have some degree of invariance in terms of rotation, scale, lighting conditions, viewpoint change etc., quantized histogram representations are preferred. Often, such representations are normalized either locally based on the histogram, or globally based on the corpus [18], [19]. Examples of global descriptors from the literature include texture based descriptors, such as, the Edge Histogram Descriptor - EHD [20], Local Binary Patterns - LBP [21], and Rotation Invariant LBP [21] that represent the spatial distribution of edges and local texture patterns. Alternatively, shape descriptors, such as, Pyramid Histogram of Oriented Gradients - PHOG [22] that builds upon Histogram Oriented Gardients (HOG) [23] to represent the spatial layout of local image shape, and (CENsus TRansform hISTogram - CENTRIST [24] and its variant Spatial CENTRIST [24], that encode the structural properties within an image suppressing detailed textural information for scene recognition. Other methods focus on color information for building global image descriptions. To name a few, the RGB Histogram descriptor [2] approximates the distribution of colors subdividing the RGB color space, the Opponent Histogram descriptor [25] uses a combination of three 1-D histograms of the three channels of the opponent color space presenting shift-invariance with respect to light intensity, the Auto-Color Correlograms descriptor [26] measures and encodes how often color x occurs in the immediate vicinity of color x, while from the `MPEG-7` family of descriptors the Scalable Color Descriptor - SCD [27] builds a color histogram in a fixed HSV space through a uniform quantization of the space, while the Color Layout Descriptor - CLD [27] represents the spatial distribution of the colors in images. Over the last few years, several attempts focused on combining multiple types of features to improve retrieval accuracy.

However, feature fusion, which is repeatedly reported to perform superiorly [28], relies on early or late fusion techniques that increase both the complexity and the overall execution time for indexing and retrieval tasks. To surpass these drawbacks composite descriptors were introduced. Representative examples of such are the Color and Edge Directivity Descriptor - CEDD [18] that utilizes both color and edge information in a compact, quantized manner, and the Fuzzy Color and Texture Histogram descriptor - FCTH [19] that encodes texture

using the high frequency bands of the Haar wavelet transform together with color information.

The foremost advantages of extracting global features is the low cost of the single-feature space computations. An additional advantage of such features is the fact that indexing one image is independent of the type or the total number of images in the collection. However, annotating an image solely by a global feature vector often leads to a rather generalized outline of its visual information. Therefore, retrieved results to a given query usually manage to capture visual similarity but often lack in extracting semantic similarity.

Local features representations, on the other hand, manage to include correct results even for verbose images or images where objects appear with partial occlusions. This property has been especially useful for image classification and object/face recognition tasks [29], [30], [31], [32]. For CBIR many techniques have been proposed [33], [34], [35], [36], [3], [37] that utilize extracted local features and then aggregate them so as to form a single vector representation, i.e., an image descriptor for index and retrieval. Methods that employ local image features search for salient image patches which are local extrema of some function on the image -like edges, corners and blobs-, detecting and then describing what is commonly referred to as points-of-interest (POI). Among the most popular POI and blob detectors are SIFT [38] and SURF [39]. Typically, after feature extraction, samples are forwarded to a classifier to form codebooks or vocabularies. Later, an aggregation step takes place employing models such as Bag-of-Visual-Words or Vector of Locally Aggregated Descriptors [40], so as to result to a single descriptor.

The enhanced retrieval robustness, however, comes with a higher computational cost and limited scalability. The process of extracting the local features, evaluating and classifying them takes place in a high dimensional feature space, and thus, a considerable complexity is introduced. Another considerable drawback of such approaches is related to the formation of the codebooks. Apart from the many free parameters introduced when employing codebooks; such as, deciding the optimum clustering size and technique, weighting its terms according to the specific statistics of the collection, and deciding whether the terms found in the images will be hard or soft assigned to the codebook terms, probably, the most limiting aspect of such approaches lies in the fact that a new codebook needs to be computed each time a significant amount of images is added to an indexed collection. This introduces the need of re-indexing the whole dataset in order for the retrieval to remain robust.

Deep learning methods, are recently gaining traction and outperform the traditional low level features in many vision tasks [41], [42]. Despite recent research attention on applying deep learning techniques for image classification and recognition in computer vision, there is still a limited amount of attention focusing on applications for CBIR [43], [4]. Finding the link between pixels and semantic understanding with deep learning is approached by training large neural networks on datasets that either define pairwise image similarity or categories for images like the popular ImageNet dataset [44]; which includes $14,197,122$ images in $21,841$ synsets. In a neural network the weights are learned by using large training sets and the output of the last layers can then be used as a global image feature. While the use of a neural network, in an optimal case, provides the best possible global feature according to the training data, common drawbacks are concerned with: (i) the significant processing time required to calculate the weights for deep networks, i.e., model creation; (ii) finding enough data for a domain to train a model to achieve optimal retrieval performance [45]; (iii) finding an optimal model architecture and learning strategy for the domain and scenario, and (iv) the vectors generated by the last layers of a neural network are typically rather large and not quantized, therefore hard to handle on limited resources [46]. For the first two drawbacks, with regards to machine learning, typically transfer learning techniques are applied [47], [45]. However, limitations for transfer learning still apply if there are no or only ill-fitting source models for transfer learning. Moreover, approaches based on transfer learning are often not robust to scaling, cropping and cluttered images [41].

Overall, deep learning based techniques require a significant amount of data for training computations, and for generating accurate results. In addition, such techniques require training the model to a specific domain that requires the identification of the appropriate dataset. Such a process is only made feasible with the existence of experts in data science or machine learning. Notwithstanding the significant improvements introduced by such approaches, in some use cases and in special domains, large repositories of images are not readily available and one cannot employ external experts into the project. Examples include: commercial image search engines; where images are not transferred unless payment is done, classification of explicit material; where transmission of the data defeats the purpose of a filter, or large scale investigations of criminal material; for instance, cases involving child abuse. These scenarios indicate examples where training data are not always sufficient or easily accessible, especially in controlled environments; where access to sensitive image data is not always permitted. Moreover, the probably best-known disadvantage of deep learning approaches is their "black box" nature, meaning that we don't know *how* and *why* a Neural Network came up with a certain output.

III. SaCoCo: Method description

The Salient Contours and Color information Descriptor (SaCoCo), comprises of two main unsupervised processing units: (i) the *Contour Unit*; for extracting the contour information, and (ii) the *Color Unit*; for integrating color information through a two-staged fuzzy-linking scheme.

Inspired by the effectiveness of the Human Visual System (HVS) in being able to achieve the designed considerations mentioned above, our method is loosely based on the way neurons of the primary visual cortex promote the integration of salient contours. Thus, before proceeding with laying out the details of the proposed methodology along with its implementation details, we briefly introduce the fundamentals with regards to the HVS.

The Human Visual System: The primary visual area, also known as *Area 17*, forms the first link in the chain of

cerebral analysis of a visual image [48]. The neurons forming the primary visual cortex have a hierarchic organization, a functional specialization while the spatial precision of the connections within the system allows retinotopic representation in the visual cortex, i.e., each point of the retina is projected into a specific area of the cortex. The cortex comprises of autonomous sets of neurons, organized into basic functional units known as hyper-columns that are responsible for analyzing independently a precise zone of the visual field. Each hyper-column is made up of multiple orientation columns, two ocular dominance columns, and *blob regions* which are sensitive to orientation/spatial frequency, color and depth, respectively.

The popular model used for depicting hyper-columns is the pinwheel model [49], [50]. According to this model, the center of a hyper-column is occupied by cells that respond and identify the wavelength of light incident to the retinal part to which they are attached. In a radial pattern around those cells are cells responsive to edge orientation and motion in particular directions. Moving from the top of a hyper-column to its bottom the cells are scaled allowing thus the detection of edges of various orientations in different scales for every point of the visual field.

One of the main models for early vision in humans, as proposed by Neisser [51] suggests that it consists of pre-attentive and attentive stages. In the pre-attentive stages, context is not taken under consideration and only image-driven data are utilized to detect pop-out (salient) features. Next, during the attentive stages, the detected features are grouped and classified based on their distinct characteristics.

On another note, Marr [52] focused on the computational analysis of vision. Marr modeled the human visual system into three associated processing stages that transform a structure-less two-dimensional representation of the visual scene into a three-dimensional representation of the visual environment, that can serve as input to recognition and classification processes.

During the first stage, the image information is processed so as to obtain what is referred to as the primary sketch; an image of salient contours similar to the one obtained when squinting the eyes. The primary sketch emphasizes on visual stimulation landing on the retina, particularly on changes in intensity values along with their geometry and overall organization. The process of exporting the primary sketch of a visual scene is based on the calculation of zero-crossings (changes from light to dark). Characteristics represented in the primary sketch are exported consecutively for different scales. This allows the separation of the main characteristics from details.

The next higher processing stage produces the 2.5 dimensional sketch that displays orientation of visible surfaces in viewer-centered coordinates, while the third and final stage is a three-dimensional object-centered representation where the scene is visualized in a continuous, 3-Dmap, hierarchically organized in terms of surface and volumetric primitives.

The proposed approach aims in simulating the pre-attentive stages of vision as defined by Neisser to achieve a light-weighted implementation. The pop-up features we are extracting (contours) refer to the first stage of Marr's computational analysis of vision and are the building blocks of our scheme. Thus, even though our method is not tuned for precise salient contour extraction we still emphasize the extraction process, and provide visualized output results of the extracted contours in subsequent sections.

A. Contour Unit

Scaling: The input image is re-sampled to form three different layers of a spatial pyramid, using bilinear interpolation (Fig. 1 – 1.1 Scaling). The pyramid is a data structure consisting of the same image represented several times, at a decreasing spatial resolution each time. Each level of the pyramid contains the image at a particular resolution. For easy reference the scales are named `Fine`, `Middle` and `Small`, with the `Fine` scale being 600×600 pixels, the `Middle` 300×300, and the `Small` 150×150 pixels.

The benefits of employing the spatial pyramid approach for our implementation is two-fold: (i) *scalability*; due to the large number of images involved in CBIR, the spatial pyramid allows us to initialize the process by normalizing the images' resolutions so that feature extraction, vector building, and search can be done reliably on diverse image collections, and (ii) *standardized computational time and resources*; the controlled resolution of the images is related to computational time and resources. Both can be foreseen and standardized, which is a significant attribute for a variety of applications that are either time-critical or run on limited resources.

The resolution of the three layers was heuristically chosen after extensive experimental exploration and testing with image collections varying in size, resolution, theme and their levels of semantics in query to result relevance interpretation [1]. The chosen resolution allows for easy partitioning of all scales during the *tiling* process and easy segmentation of the images during the *spatial information* process (both described in the following paragraphs). We further note that the resolutions employed for the spatial pyramid align with the primary sketch of Marr's computational model [52], since they serve as a low-cost pre-processing step to eliminate fine textures and degrade noise.

Tiling: All three scales are partitioned into non-overlapping 10×10 pixels blocks (Fig. 1 – 1.2 Tiling). Every block is independently processed by a set of 2-dimensional oriented kernels. The inspiration behind this process is to imitate operations of the primary visual cortex that processes image-driven data to evaluate the parts of a scene that are pre-attentively distinctive and present some kind of immediate visual arousal. As mentioned earlier, each point of the retina is projected into a specific area of the cortex where autonomous sets of neurons, organized into hyper-columns, analyze independently a precise zone of the visual field.

Simple oriented cells are usually modeled as 2D Gabor filters or elongated 2D Gaussians of specific orientations that are used as kernels convolving with an image.

[1] Additional, experimentation was performed on the Holidays dataset [53] (original image resolutions are 2448×3264) by rescaling the whole collection prior to the retrieval to: (a) 4896×6528, (b) 1224×1632, (c) 612×816, (d) 306×408, and (e) 153×204. We observed that the retrieval performance remained stable for all test cases. The highest drop was reported for (e) where MAP had a -0.006 difference compared to the original resolution. In (c) the MAP score actually raised by $+0.003$

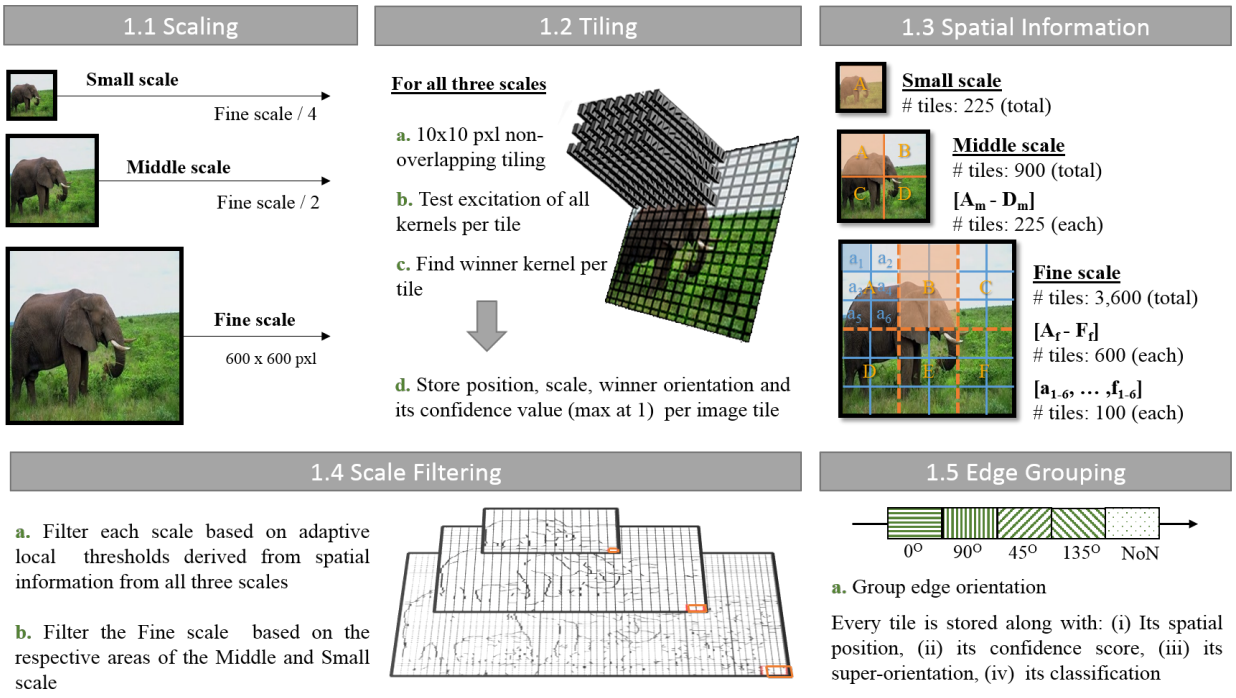

Fig. 1. Graphical abstract of the Contour Unit

Gabor filters target changes in lighting and texture to analyse the image and highlight the prominent features. They have been employed in a variety of computer vision tasks due to their excellent performance on orientation and spatial frequency selectivity. Additionally, Gabor filters are considered a more robust alternative to wavelets for joint space-frequency representations of images. Unfortunately, analyzing images by convoluting Gabor filters at pixel level is not in-line with the discrete cortical architecture of the Human Visual System. Moreover, the large number of convolutions executed in these approaches coupled with the need for predefining or tuning the parameters involved in these filters, introduce a computational cost that is prohibitive for tasks that deal with many images, like CBIR tasks.

An alternative strategy to reduce computational costs is to extract local texture information employing Haar-like features which are usually a predefined set of filters based on the Haar wavelets. Haar-like features are described by a template which includes connected black and white areas, their relative coordinates to the origin of the search window and the size of the feature, and represent a fast and simple way to calculate region derivatives at different scales by means of computing the average intensities of concrete sub-regions. Image representation based on Haar-like features succeeds at capturing local texture, however, lack orientation information.

In the proposed method we utilize a set of edge-detecting kernels, that are an adaptation of the respective kernel-masks presented in [54]. In contrast to the masks introduced in [54] that represent oriented line-segments (positive) over a background (negative), the *binary masks* utilized in our proposed methodology consist of a dark and a light region defining straight oriented edges. The proposed implementation is coupled with the employed Excitation Rule (Eq. 1), that essentially implements a normalized Haar transform to evaluate the relationship between average intensities so as to detect edges. At the same time eliminating the need to pre-process the input image with an edge-extraction mask, and a zero-crossing detector.

Furthermore, in order to extract information with regards to the orientation of the edges (during the detection stage), and inspired by the strategies in [54], [55] the kernels are designed to detect twelve orientations with 15^o increments, which are further grouped into four super-orientations (Horizontal, 45^o, Vertical, and 135^o). In order to be in-line with the discrete cortical architecture, computations on the image are taking place at a fixed lattice of discrete positions and orientations. Thus, instead of using one kernel per orientation and relying on sliding window operations over the whole image to locate edges, all the possible positions of the oriented segment, are templated as sifted instances of the same orientation. With regards to the size of the gradient orientation detection window (which in our case also matches the size of the kernels) is set at 10×10 pixels. Since the image is tiled in non-overlapping regions and each kernel directly fits on the tile's size, it makes sense to choose a kernel size that facilitates easy image tiling, given that the majority of default image sizes are multiples of 10. A kernel size of 10×10 pixels is of the same order of magnitude as the cells defined in HOG, and furthermore allows us to build a sufficient set of kernels with sifted oriented segments. Fig. 2 presents the complete set of 58 kernels, employed to form the model of the hyper-column; for each of the 12 orientations an appropriate number of instances (kernels) represents all possible positions (2-pixel shifts) of the edge within the region of the kernel.

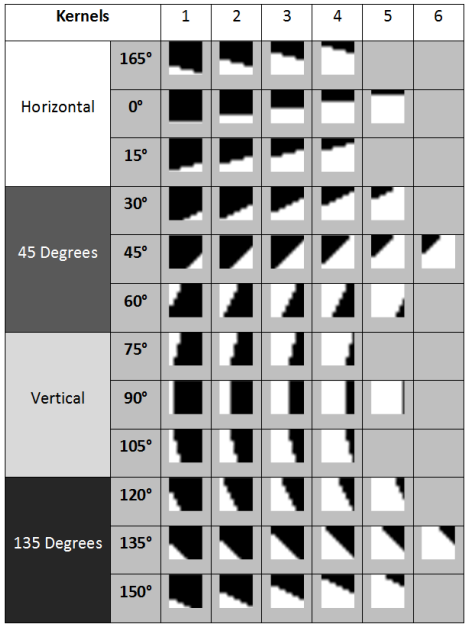

Fig. 2. Modelling the hyper-column: Edge detecting kernels

The pinwheel modeling of the hyper-column (as discussed earlier) also suggests that the multiscale search of the visual field is achieved by differently scaled cell combinations along the hyper-column. However, trying to scale the kernels introduces uncertainties related to the appropriate number of instances (edge shifts) per orientation, over all scales. Thus, in our modeling, multiscale search is performed by scaling the input image.

Similarly to [54], even though we employ a greater number of kernels to process each tile with multiple instances of an edge orientation, the total number of operations required is much smaller, when compared to sliding window operations with one kernel per orientation. Convolving a $n \times m$ image with 12 orientation filters demands $12 \times n \times m$ convolutions, while the tiling method demands only $58 \times (n/10) \times (m/10)$, to process the same number of orientations.

Every tile of all three scaled images is processed by the hyper-column structure to define the best fitting edge among the 58 kernels, independently of the others. To do so, the images are transformed to the YIQ space. The Excitation rule that defines the confidence score c_i in the interval [0,1], of every kernel k_i of the hyper-column with i in $[1, 58] \in \mathbb{Z}$, is implemented as follows:

For every tile t of the image,

$$c_i = \left| \sum_{x=1}^{10} \sum_{y=1}^{10} \left[\frac{t_{(x,y)} \times k_{i(x,y)}}{|L_i|} - \frac{t_{(x,y)} \times (1 - k_i(x,y))}{|T| - |L_i|} \right] \right| \tag{1}$$

where, $t_{(x,y)}$ is the luminance value Y of tile pixel at position (x, y),

$$t_{(x,y)} : (x, y) \rightarrow [0, 1]$$

$$k_{i(x,y)} = \begin{cases} 1, & \forall T_{(x,y)} \in L_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where T denotes the set of pixels belonging to tile t; L_i is the subset of pixels in tile t that fall under the light region of a given kernel k_i, and; D_i is the subset pixels in tile t that fall under the dark region of a given kernel k_i such that,

$$L_i \cup D_i = T \tag{3}$$

Thus, $|T|$ denotes the cardinality of T, i.e., number of elements (pixels) in tile t; $|L_i|$ denotes the cardinality of L_i, i.e., the number of elements (pixels) in the L_i subset, and; $|T| - |L_i|$ equals the cardinality of D_i.

When all tiles of the three images have been processed, the method stores the position of the tile of each image along with the respective best calculated confidence score (max at 1), and the index i of the winner kernel in the hyper-column (i.e., the detected orientation and super-orientation).

***Spatial Information**:* Through the Tiling procedure our method calculates the winner kernel along with its confidence score for each tile-block of the images. As a later step, the method calculates statistical data of the confidence scores based on their spatial distribution. These data are then used as input for filtering the kernels in an adaptive, automated manner allowing us to differentiate between tiles containing parts of prominent edges, and contours from tiles that do not (background/texture).

For our model, it is essential that the input image is treated in a fashion that enables the estimation and extraction of the most useful object/region boundaries and contours, over the whole depiction. However, as previously discussed, the computation of the absolute edge-confidence value is not indicative of the saliency of the tile. A universal hard threshold cannot be set successfully for the segmentation process neither per image nor over the whole collection.

Our solution considers the above limitation towards implementing an adaptive, unsupervised algorithm to set thresholds. The first step of this process is to extract information on a local space over all three scales. Towards this direction, our method introduces a partitioning strategy designed to be both lightweight and effective over a variety of depictions. Segmenting without any prior knowledge of the characteristics of the visual composition demands complex implementations that are not fit to be part of a CBIR system. Thus, we propose a *grid-based partitioning process* that takes into account the most commonly applied composition strategies in an attempt to make assumptions concerning the various components that appear in the image.

The grid-based partitioning process uses as input the `Fine` scale representation of an image to partition it into smaller regions by adapting on the classic *rule-of-thirds* image composition principles [56]. This rule, apart from being widely used in professional photo-shooting, is also being embedded as a default assistance software for both mobile, and standalone commercial cameras. The basic principle behind the rule-of-thirds is to break a given image down into thirds (both

horizontally and vertically) resulting with nine parts. The produced grid identifies lines and intersections to place important parts of the image. Fig. 3 depicts the partitioning of an input image based on the rule-of-thirds, and our adaptation. More specifically, the depicted grid is created using the horizontal lines 1h and 2h, along with the vertical lines 1v and 2v. The proposed partitioning uses the same lines for the vertical sectioning but replaces the horizontal lines (1h and 2h) with only one line located in the middle of the image.

This is essential for the partitioning process to ensure that the *horizon line* (an important contour according to the rule-of-third) is placed either along line 1h or 2h and is not lost due to the partitioning. A subsequent section on Filtering discusses thoroughly, how the image segments are evaluated separately from one another; statistical data of the confidence values that exist in each image part help to adaptively define a unique threshold to filter edges belonging to texture and background, from prominent edges and contours.

Furthermore, it is common practice to place the main theme of the composition inside the conceivable circle as shown in the center of the image with a radius equal to the $1/4$ of the diagonals (this is depicted in Fig. 3 as green points). If this is the case, by employing the proposed partitioning strategy we ensure that all six segments contain both parts of the main depiction, as well as, background. Again, the prominent edges of the main objects in the image are used to separate background from foreground.

Each first-level segment of the proposed partitioning process of the `Fine` scale represents the $1/6$ of the image. Since the `Fine` scale holds substantial information compared to the other scales, we further partition each first-level segment into smaller segments to gain a second-level of even more localized aspects of the depiction.

To sum up, as shown in Fig. 1 (see 1.3 – Spatial Information) the `Fine` scale is partitioned into six equally-sized subregions $[A_f - F_f]$, where each one of them is further partitioned into six smaller regions $[a_{1-6}, ..., f_{1-6}]$. The `Middle` scale is partitioned into four equally-sized regions $[A_m - D_m]$, whereas the `Small` scale is not further partitioned.

The sizes of the regions and subregions over all scales i.e., the number of tiles comprising each segment (as shown in (Fig. 1 – 1.3 Spatial Information) are defined as follows:

- `Fine` scale: a_f is to A_f as A_f is to $Image_{Fine}$
- `Middle` scale: $A_m$is to $Image_{Middle}$ as $Image_{Middle}$ is to $Image_{Fine}$
- `Small` scale: $Image_{Small}$ is to $Image_{Middle}$ as A_m is to $Image_{Middle}$.

Filtering*:* This step enables the proposed method to adaptively differentiate between image tiles that are characterized as contours (salient edge information), and tiles that are classified as background/texture information. Two filtering stages are proposed; `stage-1`: the derived localized thresholds are used to filter separately (but not independently) each of the three scales, and `stage-2`: the filtered results per scale are combined to further filter the final contour representation of the `Fine` scale.

Filtering – Stage-1: At this stage we compute a number of statistical measures for each of the three scales that will allow us to filter each scale with thresholds that adapt to different levels of spatial correlation. Specifically, we measure the *mean confidence score* over all tiles per scale, denoted as ($\bar{c_f}$, $\bar{c_m}$ and $\bar{c_s}$), and the *respective coefficient of variation*, denoted as (CV_f, CV_m and CV_s). These measures serve as an initial rough indicator of the global characteristics of the depiction.

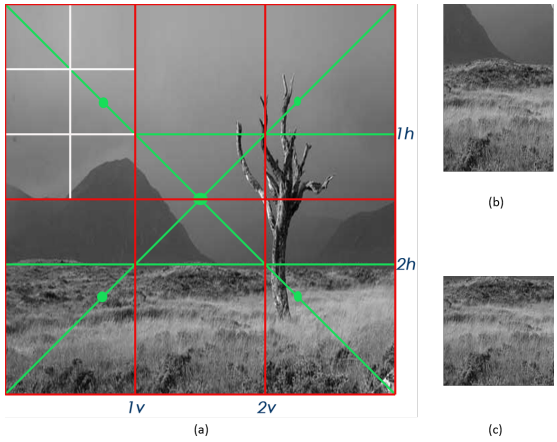

Fig. 3. (a) The proposed partitioning (red and white) against the rule-of-thirds partitioning; (b) middle lower part segment of the proposed partitioning; (c) middle lower part segment based on the rule-of-thirds grid. By observing the two example segments in b and c, it is evident that if we require to calculate for an adaptive threshold to extract only the most salient edges, based completely on the particular segment, processing the upper segment yields a more strict (higher) threshold due to the strong edges of the horizon that downgrade the edge information of the grass texture.

A] The `Fine` scale: The objective for this scale is to automate the computation of a total of 36 adaptive thresholds, one for each of the second-level regions ($a_{1-6} - f_{1-6}$). The process of setting these thresholds (T_{R2}) involves a cascade of information deriving from spatial statistic extracted at different levels and scales. More specifically, information is transferred from calculations (i) over the first-level regions (T_{R1}); (ii) over the whole image (T_I); and (iii) at all employed scales.

Initially, and consulting the information from the computed CV_f for the `Fine` scale, we concluded the following. A $CV_f > 1$ suggests that the data (i.e, confidence scores over the whole image) present a high variance. The higher the CV_f, the safer it is to set an Image Threshold (T_I) at the calculated mean confidence value of the image or, for a stricter classification, even higher than that. Thus,

$$T_I = \bar{c}_f * CV_f. \tag{4}$$

In contrast, when $CV_f < 1$ the data present a low variance, which is related to a number of reasons. For instance, low variance is reported for: (i) images where there are no clear winners (i.e, a lack of salient edges from prominent objects); (ii) images of overall low contrast (in that case $\bar{c}$ is also relatively low), and (iii) images of cluttered scenes or repetitive texture patterns (in these cases $\bar{c}$ is relatively high). Therefore, for the $CV_f < 1$ case, the definition of the (T_I) takes into account information from the other two scales so that,

$$T_I = \bar{c}_f + \bar{c}_f * \bar{CV}_{f,m,s} \tag{5}$$

where,

$$\bar{CV}_{f,m,s} = \frac{CV_f + CV_m + CV_s}{3}. \quad (6)$$

In cases of *problematic* images the method yields different $\bar{CV}_{f,m,s}$. As an example consider the case where the `Fine` scale presents a low variance due to fine texture over an extended part of the image (e.g., grass). In this case, the CV of the smaller scales, that degrade this type of textures, will be significantly higher. On the other hand, when the low variance presented is a result of low contrast, CV will be low for all the three scales, and T_I will not be much higher than the computed $\bar{c}_f$ (Eq. 5)

Next, the computed T_I is compared with the mean confidence scores ($\bar{c}_{R1}$) of the six first-level regions ($A_f - F_f$). The six thresholds of these regions (T_{R1}) are computed as follows:

$$T_{R1} = \begin{cases} 2\bar{c}_{R1} - \frac{\bar{c}_{R1}^2}{T_I}, & T_I \geq \bar{c}_{R1} \\ \\ 2T_I - \frac{{T_I}^2}{\bar{c}_{R1}}, & \text{otherwise} \end{cases} \quad (7)$$

To define T_{R1} a conservative strategy is followed. The smallest value between the previously calculated T_I of the image and the mean confidence value of the region at hand, $\bar{c}_{R1}$ is discovered. Then, the threshold for the region is defined as the smallest value increased in relation to the percentage they differ.

This strategy prohibits the over-filtering of parts of the image with localized strong edges among an overall low contrast image, or the under-filtering of local parts of the image when overall there are strong edges in other areas of the depiction. For instance, consider an image that depicts grass and has a prominent object only in the upper left part of the image. The confidence values will be low for almost the whole depiction with the only exception being the region that contains the object. Even though, based on this region's mean confidence, the threshold is expected to be high, it is unfair, compared to the rest of the image to over-filter it. Instead, the actual region's threshold will be set at max, that is equal to $2 * T_I$. In the opposite case, where strong edges exist and are extracted over the whole depiction but one region has limited to no edges (e.g., a part depicting the sky with soft clouds), the process defined by Eq. 7 will consider the overall scores embedded in T_I and increase the region's threshold up to double, i.e., its mean confidence value.

Finally, for every second-level region, the method calculates it's mean confidence value $\bar{c}_{R2}$, compare it to the respective T_{R1} of the first-level region that it belongs to, and defines the threshold T_{R2} as follows:

$$T_{R2} = \begin{cases} \bar{c}_{R2}, & \bar{c}_{R2} \geq T_{R1} \\ \\ T_{R1}, & \text{otherwise} \end{cases} \quad (8)$$

The 36 T_{R2} thresholds computed are then used to guide the classification process of the tiles as carrying salient contours or not, each at its respective set of tiles on the image.

B] The `Middle` scale filtering follows the exact same procedure as described for the `Fine` scale, up to threshold T_{R1}. Initially, the overall Image Threshold T_I is computed by applying the Eq. 4 and 5, and then the method proceeds by deriving the four thresholds for the regions $A_m - D_m$ as depicted in Fig. 1 – 1.3 Spatial Information) using Eq. 7.

C] Due to the heavy scaling that has already degraded the fine texture and weak edges, all tiles from the `Small` scale are filtered by a single threshold, the $\bar{c}_s$ lowered by 10%. Any tile scoring a confidence value lower than $\bar{c}_s * 0.9$, is characterized as non-contour.

Filtering – Stage-2: This stage enables the method to further filter the previously characterized contour-carrying tiles of the `Fine` scale. The image area that *a single* tile of the `Small` scale occupies corresponds to a respective 2×2 tiles' area in the `Middle` scale, and a 4×4 tiles' area in the `Fine` scale. Thus, if a single tile of the `Small` scale along with the respective 2×2 tiles of the `Middle` scale are classified as non-contour, then all tiles from the respective 4×4 tiles' area of the `Fine` scale are getting filtered.

Fig. 4 visualizes example outcomes derived from the *Contour Unit*. In order to assist the visualization, for images shown at positions $a2, a4, a6, b1, b2, b3, b4, b6$ and $c1, c3, c6$ the winner kernels for all the tiles are represented by a one-pixel-wide edge along the line where the light and the dark regions of the respective binary kernel meet. Note that this substitution is not part of the proposed method and is just used here to facilitate the visualization, as well as, the comprehension of the produced outcomes. On left part of Fig. 4, i.e., columns $1, 2, 3$ and 4, `a1` and `a3` are input images, `a2 and a4` are the respective unfiltered `Fine` scales (where every tile has a winner), `b1,b3 and b2,b4`, depict the results from the *Stage-1* filtering process over the `Middle-Small` scales, and the `Fine` scales, respectively. The final salient contours representations for input images `a1 and a3` -produced after the *Stage-2* filtering- are depicted in `c1 and c3`. To give a sense of the composite information that the descriptor integrates, images at positions `c2 and c4` show the average colors per tile at the locations of the salient contours. For illustration reasons colors of the non-salient parts are not displayed. On the right part of Fig. 4 we provide additional results of the located salient contours for input images of varying depictions in terms of clutter, perspective, overlapping texture, dynamic range etc. At this point it is worth mentioning that each tile of the final Contour Unit output, despite of being classified as contour or non-contour, is accompanied by its confidence score and is participating in the final representation accordingly. Thus, even though the strength of the confidence score of the located salient contours is not visible in the visualizations of Fig. 4, stronger edges have a bigger impact on the formation of the final image descriptor, compared to the contours of week edges such as those produced by the textured areas or non-salient parts.

In order to showcase the adaptability of the thresholding method to the specifics of a depiction, we additionally prepared visualizations with manipulated images that highlight this property.

Fig. 5 depicts the results of the final contour representation starting with a low-contrast image `a1` depicting soft clouds. The method assesses the overall information and produces the contours shown in `a2`. Next, we place a prominent object

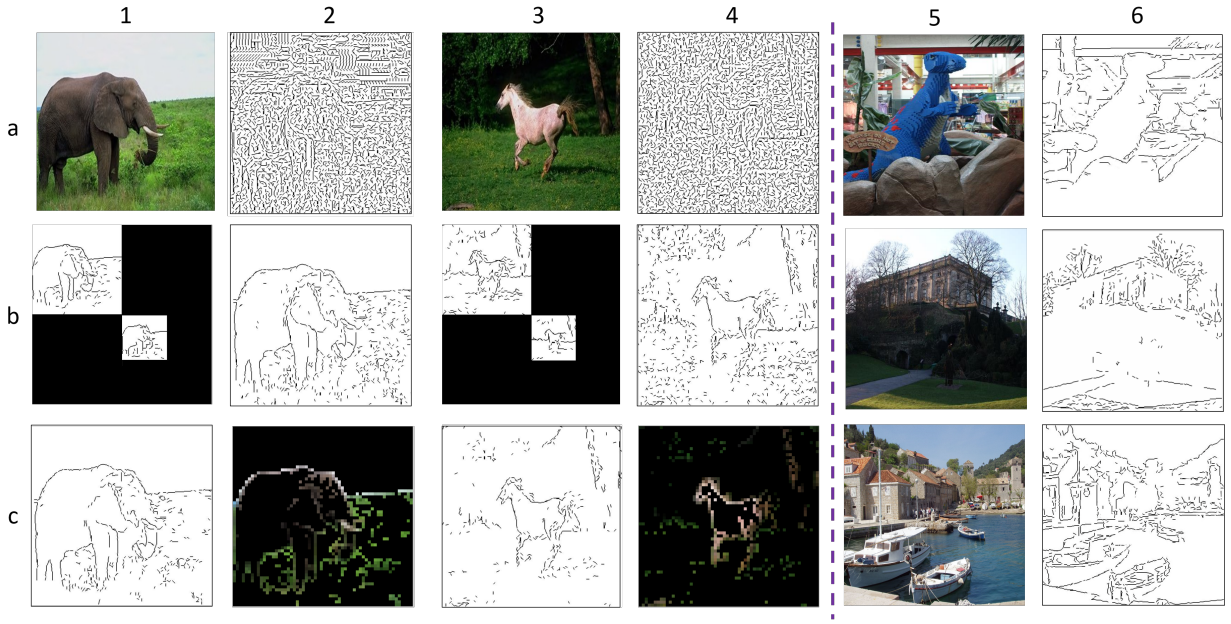

Fig. 4. Visualized results of the different stages of the Contour Unit. Left side; a1, a3 are input (images taken from COREL-1K), a2, a4 are unfiltered `Fine` scales, b1, b2 and c1, c2 are stage-1 filtered results of the `Small/Middle`, t and the `Fine` scales, c1 and c2 are the final contour representation and c2, c3 are visualized results of extracted colors at contours' location. Right Side; additional input images (a5, b5, c5) and their respective contours representations (a6, b6, c6).

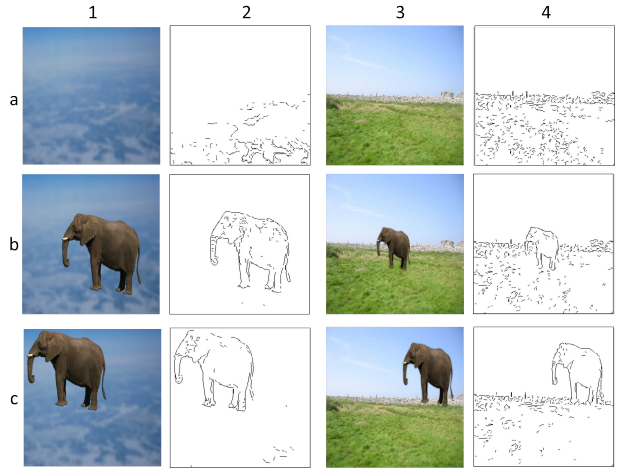

Fig. 5. Showcasing the adaptability of contours' extraction: placing a prominent object (b1, c1, b3, c3) over a low-contrast image (a1) and over a textured image (a3). Columns 2 and 4 depict the respective Contour Unit outputs

.

over the initial image. The method re-evaluates the saliency and eliminates all the cloud contours, classifying them as background to the main theme, which is clearly the flying elephant. In the third case-study `c1`, we moved the elephant away from the parts of the image that were contours in the original depiction. As shown in `c2`, the method adapts to the changed spatial distribution of contours and as a result of this some originally extracted contours re-appear.

Additionally, depiction and results in Fig. 5 columns 3 and 4 highlight the adaptability of the method over textured images. In the empty landscape of image `a3`, the contours represent the textured grass and fence areas, since nothing else is happening. As soon as we place a small-sized prominent object e.g., `b3`, much of the texture is filtered out, `b4`. When we scale the object up, `c3`, and despite the fact that we place the object at a distance from the most textured part of the depiction, the saliency is re-evaluated and the textured parts are significantly filtered out.

Edge Grouping: This is the final step of the Contour Unit (Fig. 1 – 1.5 Edge Grouping) process. In this step, the method aims to further quantize the edge orientation, and group the edges according to their respective super-orientations (refer to Fig. 2). Thus, by the end of the contour extraction process every tile of the filtered `Fine` scale is annotated with the following properties:

(i) its spatial position in the image: $[1, 3600] \in \mathbb{Z}$;
(ii) its confidence value: $[0, 1] \in \mathbb{R}$;
(iii) the super-orientation of the winner kernel: [0^o, 45^o, 90^o, 135^o], and
(iv) an index value [1 or 0] to indicate whether it has been classified as contour or not.

B. Color Unit

Color information is incorporated in the proposed image descriptor utilizing the color extraction unit as proposed by [18]. The graphical abstract of the Color Unit is shown in Fig. 6.

Average Color per Tile: For each tile of the `Fine` scale we compute the mean RGB color of its pixels and transform it to the HSV color space.

Fuzzy Color Histogram: At the *first stage*, the employed fuzzy system generates a fuzzy-linking histogram that uses the three HSV channels of a given tile as inputs to form a 10-bin

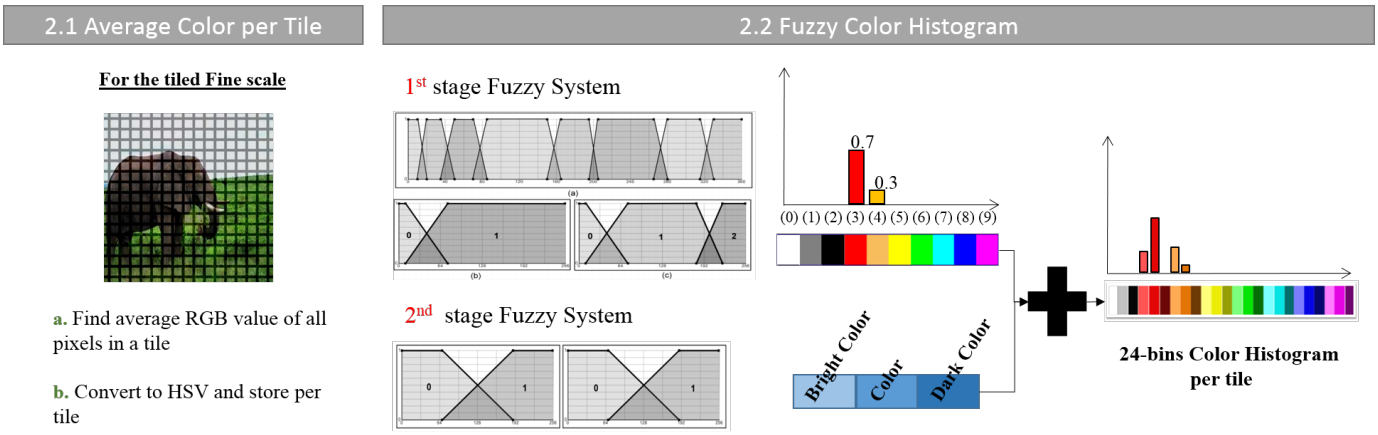

Fig. 6. Graphical abstract of the Color Unit.

histogram as output. Each bin represents a preset color: (0) White, (1) Gray, (2) Black, (3) Red, (4) Orange, (5) Yellow, (6) Green, (7) Cyan, (8) Blue and (9) Magenta.

Channel H is divided into eight fuzzy areas (Fig. 6 2.2 – 1st stage Fuzzy System) defined as follows: (0) Red to Orange, (1) Orange, (2) Yellow, (3) Green, (4) Cyan, (5) Blue, (6) Magenta and (7) Magenta to Red. For more details concerning the boundaries and the shaping of the membership functions we refer the reader to [18].

Channel S is divided into two fuzzy areas (0, 1) and *Channel V*, is divided into three areas (0, 1, 2). S values that fall in the first area link to a non-color output, black, grey or white, depending on the activation happening for channel V (0, 1, and 2 respectively). S values that fall in the second area link to varying color outputs depending on the activation of H, as long as V is not in its first area. If V falls in the first fuzzy area the output in this case is black, independently from the other input values. For more details regarding the fuzzy rules used to produce the crisp outputs, the reader is refered to [57].

At the *second stage* of the fuzzy-linking system, the method produces a 24-bin histogram as output. Each bin represents a preset color as follows: (0) White, (1) Grey, (2) Black, (3) Light Red, (4) Red, (5) Dark Red, (6) Light Orange, (7) Orange, (8) Dark Orange, (9) Light Yellow, (10) Yellow, (11) Dark Yellow, (12) Light Green, (13) Green, (14) Dark Green, (15) Light Cyan, (16) Cyan, (17) Dark Cyan, (18) Light Blue, (19) Blue, (20) Dark Blue, (21) Light Magenta, (22) Magenta, and (23) Dark Magenta.

The second stage essentially extends the output produced during the first-stage; by assigning three different shades to each original color from the 10-bin pallet. To define the different shades (Light Color, Color, and Dark Color) a second fuzzy system is employed that uses the values of S and V as inputs.

Both Channels S and V are divided into two fuzzy regions (Fig. 6 2.2 – 2nd stage Fuzzy System). For values of V that fall in the first fuzzy area, and independently of S, the original color (from the 10-bin histogram) is assigned to the respective Dark Color (in the 24-bins histogram). The S values from the other hand suggests whether the assignment of the original color is assigned to Light Color or remains the same in the final 24-bins palette. Note that since the first three bins are already shades of Grey (White, Grey, and Black) their values are transferred directly to the final 24-bins histogram.

C. Forming the Descriptor

The final descriptor is formed by combining the information extracted both from the Contour and the Color Unit. The output from the Color Unit is a 24-bins color for each tile. This histogram is multiplied (weighted) by its associated confidence score computed by the Contour Unit and added to the respective super-orientation bin (0^o, 45^o, 90^o or 135^o) of the final descriptor if it was classified as contour, or to the fifth NoN bin if it was classified as non-contour. Fig. 7 illustrates the $5 \times 24 = 120$-bins SaCoCo descriptor.

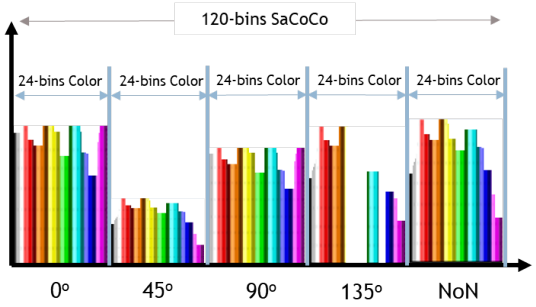

Fig. 7. The Salient Contours and Color Information Descriptor.

IV. Experimental Evaluation

A. Experimental Setup

For the evaluation of the retrieval performance of the proposed SaCoCo descriptor, experiments were conducted on seven different benchmarking datasets. Table I summarizes the main attributes from each dataset along with the query mode employed for experimentation purposes.

The proposed descriptor was compared to 13 global-feature descriptors from the literature. The choice of the descriptors used for experimentation was primarily based on their reported performance and overall popularity. Additionally, the

TABLE I
DATABASES USED FOR THE EVALUATION WITH, NUMBER OF IMAGES IN THE DATABASE, NUMBER OF QUERY IMAGES USED, AVERAGE NUMBER OF RELEVANT IMAGES PER QUERY, AND HOW THE QUERIES ARE EVALUATED.

Name	Size\Queries	Ground Truth	Query mode
UCID [58]	1338\262	3.45	query-in-groundTruth
Holidays [53]	1491\500	2.98	query-in-groundTruth
UkBench [59]	10200\250	4	query-in-groundTruth
COREL-1K [60]	1000\1000	100	query-in-groundTruth
ZuBuD [61]	1005\115	5	queries & dataset are disjoint
COREL-10K [62]	10000\10000	100	query-in-groundTruth
GHIM-10K [63]	10000\10000	500	query-in-groundTruth

chosen descriptors form a versatile collection of descriptors that allowed us to compare our methodology against many different dimensions, such as, the type of information they extract (color, texture, and spatial distribution) compactness, and efficiency. Specifically, for the purposes of our experimentation we compared against: CEDD [18], JCD [64], SCD [27], CLD [27], RGB Hist. [2], Opponent Histogram, Auto-Color Correlograms, CENTRIST [24], Spatial CENTRIST [24], PHOG [22], EHD [20], LBP [21] and RILBP [21]. All of which are explained in detail in Section II.

Additionally, experiments were conducted using the local-features descriptors SURF and SIFT with two aggregation methods VLAD (16 and 64) and BOVW + WS (512, 2048). BOVW stands for Bag of Visual Words, WS for Weighting Scheme, and VLAD stands for Vector of Locally Aggregated Descriptors [40]. The codebooks, in both cases, were computed by randomly forwarding 10% of the extracted features to the k-means classifier.

All descriptors (including the proposed one) were re-implemented as part of the LIRE library [65] and can be used under the GNU GPL license.

B. Experimental Results

a) Results on Benchmarking Datasets: Table II presents the Mean Average Precision (MAP) [66] scores for each descriptor per collection, while the last column shows the average MAP scores over all collections. The NS score [59] is also reported on the `UkBench` dataset, as it is the performance measure usually reported for this dataset. In general, we observed that the composite descriptors (i.e., descriptors based on both color and texture image properties) perform steadily better than the rest, achieving in all cases high MAP scores.

With regards to the color descriptors we observed that these performed reasonably well. In contrast, the texture-based descriptors reported a low performance in all cases with the only exception the ones that make use of spacial information. Especially when evaluated on the `ZuBuD` collection. After observing the collection we noted that these images are all depicting buildings whose main distinctive feature is the color they are painted. Moreover, the fact that the queries in this collection are of lower resolution forces descriptors, that construct their vector as a histogram of all pixels in the image, to completely fail without normalization.

We observed that our `SaCoCo` descriptor performs robustly over all collections reporting the highest average MAP score of 0.5636. `SaCoCo` is the best performing descriptor in 5 out of 7 datasets (`UCID, Holidays, COREL-1K, COREL-10K and GHIM-10K`) and the second best descriptor, with a slight non-significant difference from the first, in the rest of the collections (`UkBench` and `ZuBuD`).

Table III reports the experimental performance of the SURF and SIFT descriptors on 5 image collections. Note that this table focuses on the best performing setups for each case (VLAD 16/64 and BOVW 512/2048 + best performing weighting scheme). SaCoCo outperforms simple SURF and SIFT implementations for all collections.

Recent literature contains several sophisticated methods and algorithms that outperform the retrieval accuracy of the proposed descriptor. Table IV presents a comparison of different image retrieval proposals that adopt large scale visual vocabularies and deep learning-based methods on the Holidays database. More extensive evaluation results of the state-of-the-art methods on this database are reported in [42]. As one can easily observe, even when compared with these methods, the performance of SaCoCo is competitive comparable. Moreover, the proposed descriptor is a *plug-n-play* method, which can be adopted for description and retrieval without any prior initialization or training. The foremost advantages of SaCoCo is the low cost of the single-feature space computations along with the fact that indexing one image is independent of the type or the total number of images in a collection.

b) Large-scale Experiments: In order to test the scalability of our proposed method we incorporated a large scale image database as distractors in the retrieval database, similarly to common practice [71], [28], [72], [73]. This practice allowed us to evaluate the scalability of our method, overcoming the lack of a publicly available large dataset with an assigned ground truth for CBIR. Thus, we gradually populated the original datasets with fractions of randomly selected images (i.e., distractors) from the `MIRFLICKR-1M` collection [74].

The evaluation of a descriptor is based on the retrieved ranked list of images per query, compared to the initial collection's ground truth. This means that retrieved images that are part of the distractors are considered false results by default.

Table V reports the MAP evaluations of SaCoCo, ACC, CEDD, and RGB Histogram for $10,000$, $100,000$ and $1,000,000$ distractors and the percentage of performance degradation compared to the ones observed without distractors (Table II) according to the same metric. Experiments were conducted on 5 image collections.

The results indicate that SaCoCo manages to perform satisfactorily in these large scale setups compared against to the rest of the descriptors used for evaluation. Percentage-wise only ACC shows a smaller degradation than SaCoCo, possibly

TABLE II
RETRIEVAL SCORE PER IMAGE COLLECTION AND AVERAGE SCORE OVER ALL COLLECTIONS. BOLD FONTS INDICATE THE TOP FOUR RESULTS AND UNDERLINED FONTS THE BEST PERFORMANCE OVER ALL DESCRIPTORS.

Descriptor	UCID	Holidays	UkBench		COREL-1K	ZuBuD	COREL-10K	GHIM-10k	Avg
	MAP	MAP	MAP	NS	MAP	MAP	MAP	MAP	MAP
SaCoCo	**0.7204**	**0.7616**	**0.8662**	**3.336**	**0.5414**	**0.7253**	**0.1760**	**0.1543**	**0.5636**
CEDD	**0.6740**	**0.7263**	**0.8055**	**3.068**	**0.5040**	**0.7226**	**0.1407**	**0.1439**	**0.5310**
JCD	**0.6945**	**0.7351**	**0.8480**	**3.248**	**0.5140**	**0.7263**	**0.1598**	**0.1434**	**0.5459**
SCD	0.4958	0.5369	0.4676	1.660	0.3186	0.3508	0.0793	0.0471	0.3280
CLD	0.5531	0.6395	0.6679	2.524	0.4480	0.5851	0.1121	0.0896	0.4422
RGB Hist	0.5871	0.6558	0.7385	2.760	0.4650	**0.5868**	0.1373	0.0903	0.4658
OppHist	0.5900	0.6583	0.7352	2.752	0.4614	0.5809	0.1173	0.0945	0.4625
ACC	**0.7083**	**0.7557**	**0.8866**	**3.408**	0.4706	0.5044	**0.1647**	0.0993	0.5128
CENTRIST	0.5595	0.6046	0.5640	2.076	0.4401	0.0293	0.1156	0.1046	0.3454
Sp. CENTRIST	0.6393	0.6738	0.7114	2.732	**0.4769**	0.1052	0.1558	**0.1303**	0.4132
PHOG	0.5369	0.6037	0.5077	1.924	0.3770	0.4416	0.1023	0.1154	0.3835
EHD	0.5019	0.5551	0.4832	1.804	0.3454	0.3819	0.1016	0.0793	0.3498
LBP	0.5325	0.5575	0.5302	1.964	0.3699	0.1321	0.0955	0.0810	0.3284
RILBP	0.4910	0.5067	0.4134	1.528	0.3502	0.0550	0.0639	0.0608	0.2773

TABLE III
MAP EVALUATIONS FOR SURF AND SIFT. THE LETTER V INDICATES VLAD FOLLOWED BY THE SIZE OF THE CODEBOOK, B INDICATES BOVW FOLLOWED BY THE SIZE OF THE CODEBOOK AND THE WEIGHTING SCHEME.

	UCID		Holidays		UkBench		COREL-1K		ZuBuD	
SURF	V.64	0.6441	V.16	0.7169	V.16	0.6681	V.64	0.437	V.64	0.6922
	B.2048 (nnc)	0.5852	B.512 (nnc)	0.6777	B.512 (nnc)	0.6711	B.2048 (nnc)	0.3801	B.2048 (nnc)	0.6131
SIFT	V.64	0.6433	V.64	0.7581	V.64	0.8047	V.64	0.4489	V.64	0.7582
	B.2048 (nnc)	0.6085	B.512 (nnc)	0.6914	B.512 (nnc)	0.6847	B.512 (lnn)	0.3756	B.2048 (nnc)	0.624

TABLE IV
HOLIDAYS IMAGE COLLECTION - PERFORMANCE COMPARISON TO STATE-OF-THE-ART METHODS.

Descriptor	MAP	Descriptor	MAP
Zheng Et al. (PPS) [28]	**0.852**	Local CoMo [3]	0.811
CNNaug-ss [67]	0.843	MOP-CNN [68]	0.802
LF - GoogLeNet [69]	0.840	Zheng Et al. [28]	0.796
LF - OxfordNet [69]	0.838	AlexNet-conv3 [70]	0.793
LF (VLAD) - GoogLeNet [69]	0.836	Mikulík Et al. [33]	0.742
Xinchao Li Et al. [36]	0.825	PhilippNet [70]	0.741
Tolias Et al. [34]	0.822	AlexNet-conv2 [70]	0.689
LF (VLAD) - OxfordNet [69]	0.816		

due to the fact that it has double the vector length, however, in terms of the average MAP score, SaCoCo always outperforms.

We note also that all descriptors suffer from a high percentage of degradation in the COREL−1K database which is caused by the fact that the employed distractors have similar images for some categories of the original collection.

C. Runtime and Memory Performance

To calculate the runtime and memory performance of the proposed descriptor, a set of experiments were conducted on an Intel Core i7−3770K CPU. The first $10,000$ images from the MIRFLICKR−1M dataset were employed as benchmarking database. In order to calculate an average extraction time per image, images need to be of same dimensions, since resolution significantly affects most of the descriptors extraction time. The size of 1024×768 pixels was chosen as it serves for a good benchmark resolution for most real-life scenarios. Table VI summarizes the main properties of the proposed method along with the global descriptors from the state-of-the-art that were re-implemented in order to ensure fair comparison.

Extraction of SaCoCo descriptor is fast and scales linearly so for extraction of n descriptors one needs $o(n)$ time. Experiments with the Java implementation of SaCoCo reported that in a single thread extraction of a single image the descriptor requires 68.43 ms on average. Using eight threads the time is reduced to an average of 18.31 ms per image. The extraction time of the proposed descriptor is of the same order of magnitude as, though smaller than, the calculation time of CEDD. At the same time, the average extraction time of all listed global descriptors is equal to 79.36 ms, higher than the calculation time of SaCoCo. It is worth noting that the proposed descriptor is the only global feature that incorporates, during the calculation procedure, color, texture and spatial information.

Memory consumption is constant for extraction and scales linearly for storage. For the single threaded extraction memory consumption peaks at 182 MB with a drop to below 16 MB after garbage collection. For the extraction with eight threads memory consumption peaks at 624 MB, but after garbage collection drops to 100 MB. This seems considerably higher, but derives from the approach of holding up to 200 uncompressed images in the in-memory queue to minimize disk access.

As one can easily observe, the proposed descriptor‘s searching time is slightly higher than the time of descriptors of similar or even smaller vector lengths. Although fluctuation is not significant, it can be justified. Most of the listed features adopt Euclidean or Tanimoto distances as divergence methods. On the other hand, the proposed approach uses Jensen-Shannon [75] divergence method. This approach is based on the Kullbac-Leibler divergence, with some notable (and useful) differences, including that it is symmetric and

TABLE V
LARGE-SCALE EXPERIMENTS: MAP EVALUATIONS AND % DEGRADATION COMPARED TO ORIGINAL SCORE, PER DESCRIPTOR.

		UCID	Holidays	UkBench	COREL-1K	ZuBuD	Avg
SaCoCo	10k	0.6792	0.7207	0.8597	0.2265	0.7102	**0.6393**
		5.72%	5.37%	0.75%	58.16%	2.08%	
ACC	10k	0.6789	0.7439	0.8825	0.2554	0.4948	0.6111
		4.15%	1.56%	0.46%	45.73%	1.90%	
CEDD	10k	0.6110	0.6723	0.7906	0.1825	0.7018	0.5916
		9.35%	7.43%	1.85%	63.79%	2.88%	
RGB Hist	10k	0.5326	0.6084	0.7269	0.1561	0.5607	0.5169
		9.28%	7.23%	1.57%	66.43%	4.45%	
SaCoCo	100k	0.6132	0.6741	0.8332	0.1028	0.6783	**0.5803**
		14.88%	11.49%	3.81%	81.01%	6.48%	
ACC	100k	0.6360	0.7198	0.8672	0.1334	0.4578	0.5628
		10.21%	4.75%	2.19%	71.65%	9.24%	
CEDD	100k	0.5327	0.6127	0.7495	0.0659	0.6635	0.5249
		20.96%	15.64%	6.95%	86.92%	8.18%	
RGB Hist	100k	0.4857	0.5519	0.6865	0.0613	0.5061	0.4583
		17.27%	15.84%	7.04%	86.82%	13.75%	
SaCoCo	1M	0.5493	0.6235	0.7879	0.0424	0.6294	**0.5265**
		24%	18%	9%	92%	13%	
ACC	1M	0.5763	0.6762	0.8336	0.0648	0.3706	0.5043
		19%	11%	6%	86%	27%	
CEDD	1M	0.4603	0.5528	0.6782	0.0262	0.5878	0.4611
		32%	24%	16%	95%	19%	
RGB Hist	1M	0.4524	0.4973	0.6135	0.0253	0.4023	0.3982
		23%	24%	17%	95%	31%	

TABLE VI
SUMMARY OF THE PROPERTIES OF THE DESCRIPTORS WITH, THE TYPE OF INFORMATION THEY SUPPORT (C:COLOR, T:TEXTURE, S:SPATIAL), THEIR VECTOR LENGTH, THE EXTRACTION TIME IN MS PER IMAGE AND SEARCH TIME IN MS, PER IMAGE (USING A SINGLE THREAD).

Descriptor	C	T	S	Vector length	Extraction time	Search time
SaCoCo	x	x	x	120-bin	68.43	15.96
CEDD	x	x		144-bin	68.51	12.54
JCD	x	x		168-bin	143.78	12.33
SCD	x			64-bin	23.70	12.80
CLD	x		x	192-bin	15.01	11.36
RGB Hist	x			64-bin	36.47	12.74
OppHist	x			64-bin	41.43	12.18
ACC	x		x	256-bin	255.70	26.78
CENTRIST		x		256-bin	50.43	14.03
Sp. CENTRIST		x	x	7,936-bin	188.58	129.34
PHOG		x	x	630-bin	181.44	12.51
EHD		x	x	80-bin	44.18	11.75
LBP		x		256-bin	26.29	11.95
RILBP		x		36-bin	65.35	11.01

it is always a finite value. Even with a different and a more computationally demanding divergence method, the searching time of the proposed descriptor is competitive comparable and significantly lower that the one of ACC and Sp. CENTRIST.

Finally, it is important to compare the runtime and memory performance of the proposed descriptor against deep learning-based state of the art methods. Using `Keras` and `Tensorflow` to extract features with the pre-trained `VGG-16` [76] model provided by Keras on the same computer took 354.17 ms per image and used up around 512 MB of memory throughout multithreaded, CPU-based extraction. Using the `InceptionResNetV2` model [77] provided by Keras, feature extraction requires, on the same computer and dataset, on average 270.15 ms and uses up around 900 MB of memory throughout multithreaded, CPU-based extraction.

TABLE VII
COMPARING THE PROPERTIES OF THE PROPOSED DESCRIPTOR AGAINST DEEP-LEARNING STATE OF THE ART METHODS (USING MULTITHREADING).

Desctiptor	Vector Length	Extraction Time	Size
SaCoCo	120	18.31	240B
VGG-16	25088	354.17	98KB
InceptionResNetV2	38400	270.15	150KB

SaCoCo features are also considerably smaller than the output of the above neural networks with 120 dimensional vectors quantized to 16-bit integers using up 240 bytes. In contrast to that, descriptors for the neural networks use 32-bit floating point numbers and have 38400 dimensions for InceptionResNetV2's layer before the last one (150 kilobytes) and 25088 dimensions for VGG-16 (98 kilobytes). Table VII summarizes the properties of the proposed descriptor against Deep-Learning based state of the art approaches.

V. CONCLUSIONS AND FUTURE EXTENSIONS

This paper presents a fully unsupervised, parameter-free, adaptive method, that extracts contour and color information to represent images. It segments the image into salient and non-salient parts, while all information is weighted and affects the final representation accordingly.

Both the contour information and the color information are effectively quantized in the final descriptor. The color information of each image part is softly assigned to a 24-hues palette, thus normalizing light hue variations and lighting conditions. The contours' orientation is initially searched for by 15 degrees increments and quantized into four super groups, normalizing and tolerating shape distortions caused from the initial image scaling and rotations up to 45 degrees.

Overall, the experimental evaluation showed that SaCoCo is well suited for CBIR. The descriptor managed to outperform all global descriptors from the literature, as well as, simple SURF and SIFT implementations when evaluated on a wide variety of diverse datasets that varied both in theme and query to database relevance assumption. Furthermore, SaCoCo showed enough evidence that can compactly and efficiently describe images with a vector representation of 120-bins that passed the test of scalability.

Experimental evaluation indicates that, although SaCoCo does not supersede all of the other approaches, it brings a new approach for domains, where global, unsupervised features are required. It is ready to use, no training or parameter fitting is needed, but still reports significantly good retrieval results. Runtime performance and memory footprint make it usable for embedded systems like smart cameras and robots. As outlined in Section IV-C SaCoCo descriptors are more than 400 times smaller than those built on VGG-16 and 640 times smaller than those extracted with the InceptionResNetV2 model. The compactness of the descriptor make it a good fit for scenarios with large amounts of data, like retrieval of video streams. Moreover, the compact descriptors can easily fit into memory for large amounts of data leading to fast online retrieval. As shown with the runtime experiment indexing is faster by an order of magnitude than the two investigated CNNs and – by design – SaCoCo does not need transfer learning, training or parameter fitting. This makes SaCoCo an interesting alternative for many small domains with limited amount of training data, expertise or computational power.

Future work mainly plans to address different aspect ratios for the spatial grid partitioning, and more domain specific datasets, such as, portraits or medical images. In addition, and since the approach is parallel-execution friendly by design, we plan to explore a GPU and/or FPGA implementation strategy to further accelerate the indexing procedure.

SaCoCo has been included in the LIRE library [65] and can be used under the GNU GPL license for testing and evaluation.

REFERENCES

[1] C.-R. Shyu, C. Brodley, A. Kak, A. Kosaka, A. Aisen, and L. Broderick, "Local versus global features for content-based image retrieval," in *Content-Based Access of Image and Video Libraries, 1998. Proceedings. IEEE Workshop on*. IEEE, 1998, pp. 30–34.

[2] T. Deselaers, D. Keysers, and H. Ney, "Features for image retrieval: An experimental comparison," *Information Retrieval*, vol. 11, no. 2, pp. 77–107, 2008.

[3] S. A. Vassou, N. Anagnostopoulos, A. Amanatiadis, K. Christodoulou, and S. A. Chatzichristofis, "Como: A compact composite moment-based descriptor for image retrieval," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, ser. CBMI '17, 2017, pp. 30:1–30:5.

[4] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, pp. 157–166.

[5] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[6] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of intelligence*. Springer, 1987, pp. 115–141.

[7] R. Desimone and J. Duncan, "Neural mechanisms of selective visual attention," *Annual review of neuroscience*, vol. 18, no. 1, pp. 193–222, 1995.

[8] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Salient object detection and segmentation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 1, pp. 1–1, 2014.

[9] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook, "Efficient salient region detection with soft image abstraction," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1529–1536.

[10] M. Donoser, M. Urschler, M. Hirzer, and H. Bischof, "Saliency driven total variation segmentation," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 817–824.

[11] U. Rutishauser, D. Walther, C. Koch, and P. Perona, "Is bottom-up attention useful for object recognition?" in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–II.

[12] N. Tong, H. Lu, Y. Zhang, and X. Ruan, "Salient object detection via global and local cues," *Pattern Recognition*, vol. 48, no. 10, pp. 3258–3267, 2015.

[13] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23–36, 2006.

[14] A. Oliva and P. G. Schyns, "Diagnostic colors mediate scene recognition," *Cognitive psychology*, vol. 41, no. 2, pp. 176–210, 2000.

[15] V. Goffaux, C. Jacques, A. Mouraux, A. Oliva, P. Schyns, and B. Rossion, "Diagnostic colours contribute to the early stages of scene categorization: Behavioural and neurophysiological evidence," *Visual Cognition*, vol. 12, no. 6, pp. 878–892, 2005.

[16] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 524–531.

[17] A. Torralba and A. Oliva, "Depth estimation from image structure," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 9, pp. 1226–1238, 2002.

[18] S. Chatzichristofis and Y. Boutalis, "Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval," *Computer Vision Systems*, vol. 5008, pp. 312–322, 2008.

[19] S. A. Chatzichristofis and Y. S. Boutalis, "Fcth: Fuzzy color and texture histogram a low level feature for accurate image retrieval," in *, IEEE Computer Society*. Klagenfurt: IEEE Computer Society, 2008, pp. 191–196.

[20] C. Won, D. Park, and S. Park, "Efficient use of mpeg-7 edge histogram descriptor," *Etri Journal*, vol. 24, no. 1, pp. 23–30, 2002.

[21] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 971–987, 2002.

[22] A. Bosch, A. Zisserman, and X. Munoz, "Representing shape with a spatial pyramid kernel," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 401–408.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[24] J. Wu and J. M. Rehg, "Centrist: A visual descriptor for scene categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1489–1501, 2011.

[25] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[26] J. Huang, S. Kumar, M. Mitra, and W. Zhu, "Image indexing using color correlograms," *US Patent 6,246,790*, vol. 12, pp. 1–16, Jun. 12 2001, uS Patent 6,246,790.

[27] B. S. Manjunath, P. Salembier, and T. Sikora, *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons, 2002, vol. 1.

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

[29] M. Gabryel and R. Damaševičius, "The image classification with different types of image features," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 497–506.

[30] S. A. A. Shah, M. Bennamoun, and F. Boussaid, "Keypoints-based surface representation for 3d modeling and 3d object recognition," *Pattern Recognition*, vol. 64, pp. 29–38, 2017.

[31] Z. Li, D. Gong, X. Li, and D. Tao, "Learning compact feature descriptor and adaptive matching framework for face recognition," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2736–2745, 2015.

[32] D. Gong, Z. Li, W. Huang, X. Li, and D. Tao, "Heterogeneous face recognition: A common encoding feature discriminant approach," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2079–2089, 2017.

[33] A. Mikulík, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part III*, 2010, pp. 1–14.

[34] G. Tolias, Y. S. Avrithis, and H. Jégou, "Image search with selective match kernels: Aggregation across single and multiple images," *International Journal of Computer Vision*, vol. 116, no. 3, pp. 247–261, 2016.

[35] G. Tolias and H. Jégou, "Visual query expansion with or without geometry: Refining local descriptors by feature aggregation," *Pattern Recognition*, vol. 47, no. 10, pp. 3466–3476, 2014.

[36] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 5153–5161.

[37] C. Iakovidou, N. Anagnostopoulos, A. Kapoutsis, Y. Boutalis, M. Lux, and S. A. Chatzichristofis, "Localizing global descriptors for content-based image retrieval," *EURASIP Journal on Advances in Signal Processing*, vol. 2015, no. 1, p. 80, 2015.

[38] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[39] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 404–417.

[40] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[41] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI*, 2016, pp. 241–257.

[42] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.

[43] R. R. Saritha, V. Paul, and P. G. Kumar, "Content based image retrieval using deep learning process," *Cluster Computing*, pp. 1–14, 2018.

[44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[45] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE, 2014, pp. 1717–1724.

[46] S. Petscharnig, M. Lux, and S. Chatzichristofis, "Dimensionality reduction for image features using deep learning and autoencoders," in *Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing*, ser. CBMI '17, 2017, pp. 23:1–23:6.

[47] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[48] K. E. Schmidt, R. Goebel, S. Löwel, and W. Singer, "The perceptual grouping criterion of colinearity is reflected by anisotropies of connections in the primary visual cortex," *European Journal of Neuroscience*, vol. 9, no. 5, pp. 1083–1089, 1997.

[49] D. McLaughlin, R. Shapley, M. Shelley, and D. Wielaard, "A neuronal network model of macaque primary visual cortex (v1): Orientation selectivity and dynamics in the input layer 4cα," *Proceedings of the National Academy of Sciences*, vol. 97, no. 14, pp. 8087–8092, 2000.

[50] M. Shelley, D. McLaughlin, R. Shapley, and J. Wielaard, "States of high conductance in a large-scale model of the visual cortex," *Journal of computational neuroscience*, vol. 13, no. 2, pp. 93–109, 2002.

[51] U. Neisser, "Visual search." *Scientific American*, 1964.

[52] D. Marr and A. Vision, "A computational investigation into the human representation and processing of visual information," *WH San Francisco: Freeman and Company*, 1982.

[53] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *European conference on computer vision*. Springer, 2008, pp. 304–317.

[54] V. Vonikakis, A. Gasteratos, and I. Andreadis, "Enhancement of perceptually salient contours using a parallel artificial cortical network," *Biological cybernetics*, vol. 94, no. 3, pp. 192–214, 2006.

[55] T. N. Mundhenk and L. Itti, "Cinnic, a new computational algorithm for the modeling of early visual contour integration in humans," *Neurocomputing*, vol. 52, pp. 599–604, 2003.

[56] S. Lok, S. Feiner, and G. Ngai, "Evaluation of visual balance for automated layout," in *Proceedings of the 9th international conference on Intelligent user interfaces*. ACM, 2004, pp. 101–108.

[57] S. A. Chatzichristofis, K Zagoris, Y. S. Boutalis and N. Papamarkos, "Accurate image retrieval based on compact composite descriptors and relevance feedback information," *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, vol. 2, pp. 207–244, 2010.

[58] G. Schaefer and M. Stich, "Ucid: An uncompressed color image database," *Storage and retrieval methods and applications for multimedia*, vol. 5307, pp. 472–480, 2004.

[59] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *CVPR, 17-22 June, New York, NY, USA*. IEEE, 2006, pp. 2161–2168.

[60] J. Z. Wang, J. Li, and G. Wiederhold, "Simplicity: Semantics-sensitive integrated matching for picture libraries," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 9, pp. 947–963, 2001.

[61] H. Shao, T. Svoboda, and L. Van Gool, "Zubud-zurich buildings database for image based recognition," *Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep*, vol. 260, 2003.

[62] G. Liu, J. Yang, and Z. Li, "Content-based image retrieval using computational visual attention model," *Pattern Recognition*, vol. 48, no. 8, pp. 2554–2566, 2015.

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

[64] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux, "Selection of the proper compact composite descriptor for improving content based image retrieval," in *SPPRA*, 2009, pp. 134–140.

[65] M. Lux and S. A. Chatzichristofis, "Lire: Lucene image retrieval - an extensible java cbir library," in *ACM International Conference on Multimedia 2008, (ACM MM), Open Source Application Competition*, Vancouver, BC, 2008, pp. 1085–1087.

[66] S. A. Chatzichristofis, C. Iakovidou, Y. S. Boutalis, and E. Angelopoulou, "Mean normalized retrieval order (MNRO): a new content-based image retrieval performance measure," *Multimedia Tools Appl.*, vol. 70, no. 3, pp. 1767–1798, 2014.

[67] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 512–519.

[68] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII*, 2014, pp. 392–407.

[69] J. Y. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, Boston, MA, USA, June 7-12, 2015*, 2015, pp. 53–61.

[70] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronnin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 2015, pp. 91–99.

[71] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 304–317.

[72] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, 2010.

[73] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building contextual visual vocabulary for large-scale image applications," in *ACM MM, 25-29 October, Firenze, Italy*. ACM, 2010, pp. 501–510.

[74] M. J. Huiskes, B. Thomee, and M. S. Lew, "New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative," in *ACM SIGMM International Conference on Multimedia Information Retrieval, MIR, 29-31 March, Philadelphia, Pennsylvania, USA*. ACM, 2010, pp. 527–536.

[75] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[76] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[77] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.