Image Replica Detection System Utilizing R-Trees and Linear Discriminant Analysis

S. Nikolopoulos S. Zafeiriou N. Nikolaidis I. Pitas

Aristotle University of Thessaloniki, Department of Informatics, Box 451, GR-54124 Thessaloniki, Greece

Abstract

This manuscript introduces a novel system for content-based identification of image replicas. The proposed approach utilizes image resemblance for deciding whether a test image has been replicated from a certain original or not. We formulate replica detection as a classification problem and show that we can optimize efficiency on a per query basis by dynamically solving a reduced mutliclass problem. For this purpose, we investigate the effective coupling of multidimensional indexing and machine learning approaches and we aim to achieve replica detection through the training of classifiers with distortions expected in a replica. Visual descriptors are indexed using an R-tree based multidimensional structure for fast image retrieval. Cases unsuccessfully handled by the R-tree are resolved by a multiclass classifier operating on the transformed feature space that results from the application of Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA). Experimental results shows that the proposed system can identify replicas with high accuracy and facilitate a wide range of applications such as copyright protection, content-based monitoring, content-aware multimedia management, etc.

Key words: Replica detection, copy image detection, perceptual hashing, robust hashing, fingerprinting, copyright protection, content-based monitoring, linear discriminant analysis (LDA), R-tree indexing.

1 Introduction

Recent technological advances in the area of multimedia content distribution have resulted in a major reorganization of this trade. Valuable digital artworks

Email addresses: nikolopo@iti.gr (S. Nikolopoulos), s.zafeiriou@imperial.ac.uk (S. Zafeiriou), nikolaid@aiia.auth.gr (N. Nikolaidis), pitas@aiia.auth.gr (I. Pitas).

can be reproduced and distributed arbitrarily, sometimes without any control by their owners. The identification of replicated data is considered an important issue for a number of applications such as copyright infringement, digital rights management, multimedia management using content-aware networks, monitoring and filtering broadcasted content (e.g., tracking of child pornography content), etc. Among the various types of multimedia content, images are a particularly valuable asset and will be the focus of this manuscript. The approaches that have been proposed for robust image identification are watermarking and, recently, image replica detection algorithms.

Watermarking is the technique of imperceptibly embedding information within the host image content [1]. Although watermarking has attracted considerable interest from both industry and academia, it bears certain deficiencies that pose limitations on its use. The requirement of embedding information in a digital image before it is made public, automatically excludes images that are already in the public domain and need to be copyright protected. Another inherent watermarking drawback is the fact that it is an active technique i.e., it modifies the content of the images to be protected. Although these modifications are in general invisible, they do exist and might create problems in certain content categories like medical images, where quality requirements are extremely high.

In order to overcome these inherent watermarking deficiencies, the scientific community started to investigate robust image identification from a content-based perspective. Replica detection, also referred as replica recognition, near-replica detection, perceptual or robust hashing [2], content-based copy detection [3], and multimedia fingerprinting aims at identifying all images that have been reproduced from a source original through the application of intentional or unintentional manipulations. It is based on image similarity and relies on the assumption that images shares plenty of information with their replicas and yet contains enough information to be discriminated from any other non-replica image. The type and severity of manipulations that should be successfully handled by a replica detection system depend on the target application.

The major benefit of such an approach stems from the fact that no additional information should be embedded within the image content, thus eliminating the invisibility constraint inherent to watermarking systems. On the other hand, the fact that the response speed and efficiency of a replica detection scheme is largely affected by the size of the original/reference image dataset, can be considered as the disadvantage of such an approach. All the above make replica detection an important alternative to watermarking that found applications on many types of multimedia data, such as video [4], [3] and audio [5]. Although the problem formulation as described above, bears many similarities with content based image retrieval (CBIR), certain differences do

exist, that are detailed in Section 2.

Image replica detection research is still in its early stages, thus only few works addressing tasks identical or slightly different to the one addressed in this manuscript can be found in the literature. In order to tackle the replica detection problem, existing works aim at a) optimizing the distance function quantifying the perceptual similarity between two images [6], [7], b) extracting highly representative and informative features for discriminating between replicas and non-replicas [8],[9], or c) using machine learning techniques and considering the problem as a classification task [10], [11], [12].

In the first case, Qamra et al [6] present an enhanced perceptual distance function (DPF) which adaptively chooses a different set of features according to their discriminative power. The benefit of this approach is that unlike other schemes that select that same features for all the images, DPF dynamically activate features (with minimum difference) in a pair-wise fashion. In the same direction Kim [7] use the ordinal measure of DCT coefficients as the feature to represent images and the ordinal measures of AC coefficients for measuring distance similarity. A scheme for the optimal selection of a similarity threshold, based on the maximum a posteriori (MAP) criterion, is used to enhance the efficiency of the employed distance function.

Concerning methods that focus on robust features extraction, Ke et al [9] use PCA-SIFT [13], a local descriptor that has been shown to be more discriminative and compact than the original SIFT [14], and features several characteristics that are ideal for solving the image replica detection problem. Roy and Chang [8] on the other hand, focus on finding a feature space where any two images in the database are well separated from each other. More precisely, the original images are slightly modified in order to increase their mutual separation within the feature space, while taking care that the perceptual difference between the original and the modified image is kept to a minimum.

Finally, in the group of methods that view the problem as a classification task, Maret et al [10] propose a method where binary classifiers based on Support Vector Machines are constructed for each original image and are independently applied to decide whether a query image is a replica or not. A variation of this system is described in [11] where indexing is used to perform a coarse and rapid selection of the most likely originals and reduce the number of classifiers that need to be applied. In a more recent work [12] the authors improve their method by trying to estimate and efficiently describe the partition of the image space that contains the replicas of a particular original image.

Even though the systems introduced in the aforementioned papers are trying to tackle the same replica detection problem, the proposed solutions, except from the ones proposed in [10], [11], [12], rely mainly on the discriminative

power of the extracted features and the effectiveness of the employed distance function. Thus, no particular attention is paid to the fact that having many similarities with a classification problem, image replica detection might benefit from the use of appropriately trained classifiers. In our work we try to take advantage of this fact by searching for an optimal space where the projection of visual features will enable the construction of more discriminant classifiers. The proposed system operates upon a database of stored originals. Its novelty stems from the fact that image similarity is dealt as a classification problem that employs a training scheme and a suitable feature space transformation in order to increase the system robustness. It generates training images based on the types of attacks that the system is designed to cope with, and during the classification process it uses class statistic information to achieve maximum separability between classes.

More specifically, each image is represented by a feature vector and a multidimensional indexing structure based on R-trees [15] is used for indexing these vectors. The "hyper-bounding boxes" employed by the R-tree are selected using an attack-oriented training strategy that aims at modeling all potential attacks that the system is designed to encounter. The structure returns a relatively small set of images (ideally one) that are candidates for being the original of the query image. In order to resolve cases where more than one candidates are returned by the R-tree we introduce the dynamic use of discriminant techniques. Each candidate original and its modified copies are assumed to form a class. Linear Discriminant Analysis (LDA) [16] is applied in order to yield more discriminant image representations taking into account class information. The resulting representations are expected to be more easily separable, since the reduced number of involved classes facilitates the estimation of a class-discriminant projection space. A classification function is subsequently applied on the projection space for selecting the image corresponding to the original version of the query, if such an image indeed exists. It must be noted that the manuscript is a largely extended and improved version of [17] where the proposed approach was initially presented.

The rest of the manuscript is organized as follows. Section 2 provides a solid definition of image replica detection and outlines its particularities with respect to image retrieval systems. The proposed image replica detection system is described in Section 3. Section 4 describes the experiments conducted and summarizes the performance evaluation results. Concluding remarks are drawn in Section 5.

2 Problem Formulation

2.1 Image Replica Detection vs Content Based Image Retrieval

The goal of a query by example content based image retrieval system (CBIR) is to return a set of database images that are related to the query image in a broad sense of similarity [18]. On the other hand, an image replica detection system (RDS) should retrieve a database image only if the query image is a replica of this image, otherwise no image should be retrieved. Thus, the notion of similarity in an RDS is considerably different than similarity in the sense of general purpose CBIR. Moreover, an RDS should be robust to malicious image manipulations and resilient to security attacks whereas such a requirement does not generally apply to CBIR. Security attacks, either try to forge a database image and cripple the system's reliability, or produce false negatives by exploiting information related to the specific attributes of the feature extraction algorithm. Finally, unlike typical CBIR applications, retrieval of more than one image is usually unacceptable for an RDS. Our intention is to briefly review the techniques utilized in the field of CBIR and focus on the ones that most coherently satisfy the aforementioned requirements. Afterwards we will proceed with the enhancements introduced in this work for coping with issues that are specific to RDS.

Let I and I_q denote an original and a query image respectively. The original images constitute the Original Image Set (database) S_I. Additionally, we define a result set S_R corresponding to the images retrieved by the system when queried with a specific image. The functionality of a CBIR system can be formulated by the following function:

$$Q(S_I, I_q) = S_R, \quad 0 < |S_R| \leq |S_I|, \tag{1}$$

where $Q(S_I, I_q)$ denotes querying S_I with I_q. The cardinality $|S_R|$ depends on the system settings and usually contains a specific number of images $|S_R| << |S_I|$, possibly sorted according to their visual resemblance with I_q. On the other hand, the functionality of a replica detection system can be expressed by the following function:

$$Q(S_I, I_q) = \begin{cases} I, \text{ if } I_q = R(I), I \in S_I \\ \emptyset, \text{ otherwise} \end{cases} \tag{2}$$

where $R(.)$ is an allowable replica generator function, i.e., a function producing attacked images such as compressed, scaled, cropped, etc. Although common

techniques are utilized in both cases there are differences between their operational models, to be detailed subsequently.

2.2 Operational Models of Image Replica Detection and Image Retrieval Systems

Some of the elements that are typical to CBIR and are also fundamental for RDS are; a) Image representation scheme, where an image I is represented by a feature vector $\mathbf{x}^I$ that is ideally of small dimension and retains all or most of the original image significant information, b) Similarity Metric $D(\mathbf{x}^{I_1}, \mathbf{x}^{I_2})$, that evaluates the resemblance between two images by measuring the distance of their feature vectors $\mathbf{x}^{I_1}$ and $\mathbf{x}^{I_2}$ and c) decision strategy which sets the rules by which the result set is selected. A few details regarding the decision strategies that are most widely used in CBIR systems will help us identify the one that most consistently adhere to RDS requirements.

Nearest Neighbor is the decision strategy used by a system attempting to answer the question "which of the images included in the database resembles most the query one". According to Nearest Neighbor the system retrieves the images that are found to be closer to the query, with respect to a specific similarity metric. Returning an empty result set can only be made feasible by imposing a dissimilarity threshold on the results. On the other hand, *Range query* is a different strategy that incorporates a threshold on the level of similarity between images, instead of specifying the cardinality of the result set. Range queries can be envisaged as defining feature space neighborhoods $N(I)$ surrounding the feature vector of each image, thus answering the question "which images resemble the query up to a specified degree". The shape of the neighborhood is determined by the similarity metric and can be a a hyper-parallelepiped (L_1 norm), a hyper-sphere (L_2 norm) or a hyper-ellipse (Mahalanobis distance). The result set cardinality $|S_R|$ may vary since the query image is likely to reside in more than one image neighborhoods. S_R might also be an empty set.

In the trivial case, the functionality requirements of an RDS are identical to those of a range query based CBIR. However their difference, apart from selecting the feature extraction method that best serves the purpose of each application, is in the way we define the image neighborhoods in the feature space. In the case of image retrieval, similarity has ideally a semantic dimension and such a system should be able to retrieve images that depict conceptually similar scenes or objects as those included in the query image. In practice, we define the feature space neighborhood $N_{IR}(I)$ of an image I on the ground of any visual similarity scheme (e.g., color similarity, contour similarity), and operate under the assumption that there is visual similarity between seman-

tically adjacent images. In the case of an RDS the images that are considered similar to a certain image are only those that have resulted from this image through some manipulation. As a consequence, the feature space neighborhood $N_{RD}(I)$ of an image I should be ideally defined in such a way so that $R(I) \in N_{RD}(I)$, where $R(.)$ is a function generating all manipulated versions of I.

Obviously, the different notion of similarity between an RDS and a CBIR system, introduces new issues that can not be confronted efficiently by simply selecting a robust feature extracting scheme and a more "tight" neighborhood in a range query decision strategy. Learning techniques, that employ training for finding optimal neighborhoods, and classification approaches, that make dynamic use of discriminant techniques to achieve better class discrimination, are the solutions introduced in this work for tackling these issues.

3 Proposed Replica Detection System

3.1 System Overview

The process of engineering the proposed system can be separated in two phases. The first deals with the database organization. Each time a new image is added into the database it is subjected to a series of predefined manipulations. These manipulations are selected according to the system specifications and simulate all types of attacks that we wish the system to be able to withstand. Feature vectors are extracted from each attacked version resulting in a matrix (from here on called the training matrix $\mathbf{T}_r$) consisting of the feature vectors of the training replicas. The training matrix is used for calculating an extent vector that is associated with the newly added original image. Afterwards, the feature vector of the new image is indexed within a multidimensional structure using the extent vector to set its neighborhood boundaries for each dimension. The formulated hyper-rectangle corresponds to the feature space neighborhood of I that will be used for range search. A graphical representation of the database organization is depicted in Fig. 1.

The second phase implements the actual replica detection functionality. An arbitrary image is submitted as a query to the indexing structure and a set of candidate originals or an empty set is returned. In order to select one of the competing candidates their feature vectors are projected into a different space, that is determined by dynamically applying LDA preceded by PCA. This is achieved by treating each candidate original and its training replicas as a separate class. Finally, the system picks the neighborhood (candidate original) whose center is closest to the query image in the projected space (see

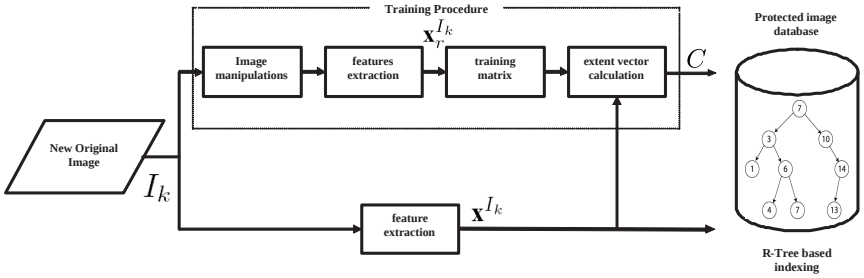


Fig. 1. Populating the image replica detection system: Training is employed for every new original image. All original images are indexed within an R-tree structured database using the corresponding extent vector.

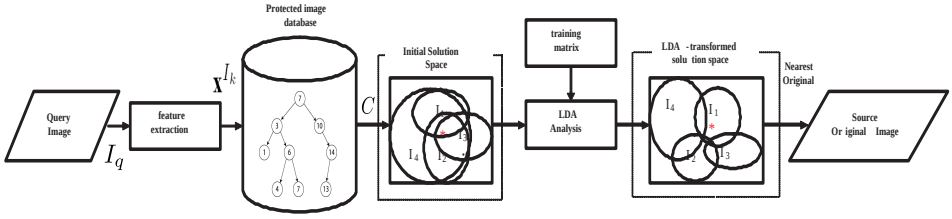


Fig. 2. Querying the image replica detection system: R-tree traversal produces a number of candidate image classes. LDA generates a more discriminant feature space where the classification function picks the source original image.

Section 3.5). If the query image is found to reside outside the neighborhoods of the candidate originals the result is an empty set. The way the queries are handled is demonstrated in Fig. 2.

3.2 Feature Extraction

Various feature extraction approaches have been proposed in the literature each one carrying different advantages and disadvantages. Important advantages include low dimensionality and reduced computational cost, high discrimination i.e., ability to distinguish between images that although share many visual characteristics they depict different scenes/objects, and robustness i.e., ability to extract very similar feature vectors for images that have been generated from the same source original using manipulations. Moreover, the particularities of the proposed scheme, outlined in Section 3.1, pose two additional constraints that limit the range of applicable feature extraction methods. Specifically, the method should be able to describe an image using a unique, fixed length vector of scalar values and estimate image similarity by calculating the distance between the corresponding vectors. Other types of descriptors such as, local descriptors combined with voting schemes that assess image similarity by counting the number of point to point matches, are not readily "compatible" with the proposed approach. Thus, a visual de-

scriptor that would be ideal for the proposed replica detection system should be a) discriminate, b) robust, c) low dimensional, d) with low computational complexity, and e) global in the sense that a single vector should be enough to represent the image.

For the purposes of the proposed system we have investigated the suitability of histogram-based color descriptors, that were tested in a replica detection setting [19] against common attacks (i.e., cropping, compression, smoothing, rotation, additive noise and luminance change). Moreover, we have examined the histogram-based descriptors provided by MPEG-7 standard and discussed in [20] and [21]. These descriptors capture different aspects of color, texture and shape, and have been widely used in a number of applications. The reason for primarily investigating histogram-based descriptors is because they satisfy most of the aforementioned requirements. Histogram-based descriptors are known to be robust against a number of attacks (i.e., geometric transformations, compression, filtering, etc), they are global, they have a relatively low number of dimensions and their extraction is usually of low computational cost. Details of the investigated descriptors are provided subsequently.

The normalized histogram of colors quantized according to the *Macbeth Color checker* chart [22] was found in [19] to outperform all other descriptors that have been tested. The image colors were quantized to the 24 colors of the Macbeth chart by assigning to each pixel the closest (in the Euclidean distance sense) chart color. Once the image colors have been quantized the 24-dimensional descriptor, from here on referred as ColorHistogram, is extracted using the following equation:

$$H_{I_i} = \frac{N_{I_i}}{N_I}, \quad i = 1 \dots C_p \tag{3}$$

where N_{I_i} is the number of pixels with color i, N_I is the total number of pixels in the Image I and C_p is the number of colors in the palette.

ColorLayout (CL) [23] is a compact descriptor that captures the spatial layout of the dominant colors on a grid. An input image is divided into 64 (8×8) blocks and their average colors are derived. These colors are transformed into a series of coefficients by performing 8×8 DCT. The descriptor is extracted by performing zigzag scanning and selecting a few low-frequency coefficients. CL operates on the YCbCr color space and yields a resolution independent 18-dimensional representation of the image.

ColorStructure (CS) [24] aims at identifying localized color distributions using an 8×8 structuring element. It counts the number of times a particular color is contained within the structuring element as the structuring element scans the image. It is defined using four different color quantization options with 184,

120, 64, and 32 bins. Each bin of the resulting histogram h_m represents the number of locations at which a pixel with color c_m falls inside the structuring element. The bin values are normalized by the number of locations of the structuring element and lie in the range [0.0, 1.0]. In our experimental study we have tested the 32-dimensional version of CS descriptor.

ScalableColor (SC) [24] consists of a normalized histogram that does not take any spatial information into account. It is initially computed in the Hue Saturation Value (HSV) color space according to a uniform quantization (16 levels in H, 4 levels in S and V) and subsequently converted through a Haar transform in a 4-bit per bin representation, assigning higher significance to small values. The resulting space is 256-dimensional and can be further reduced to half by summing all pairs of adjacent vector elements. Despite the fact that performing this process iteratively, histograms of 128, 64, 32 and 16 dimensions can be obtained, we decided to use the 128-dimensional feature vector for our experiments.

EdgeHistogram [25] represents the local distribution of edges in an image. The image is first subdivided into sub-images and local edge histograms for each of these sub-images is computed. Edges are broadly grouped into five categories: vertical, horizontal, 45 diagonal, 135 diagonal, and isotropic (nonorientation specific). Thus, each local histogram has five bins corresponding to the above five categories. By partitioning the image into 16 sub-images we get a 80-dimensional descriptor. For the purposes of our work the Canny algorithm [26] was employed for performing edge detection.

HomogeneousTexture [27] is based on the use of Gabor filters and provides a quantitative characterization of image texture. It is computed by first filtering the image with a bank of orientation and scale sensitive filters and calculating the mean and standard deviation of the filtered outputs in the frequency domain. It exhibits scale and rotation invariance and the resulting feature space consists of 62 dimensions.

Finally, we have generated a 208-dimensional descriptor, named ScalableColorEdgeHistogram (SCEH), by concatenating the 128-dimensional version of ScalableColor with EdgeHistogram. The reason for testing this descriptor was to evaluate the performance of a feature extraction approach mixing different elements of perception.

3.3 Indexing Multidimensional Feature Vectors

Multidimensional indexing structures have been widely used for performing fast search in large scale datasets. These structures can be classified in two categories [28]. The first includes the so-called space partitioning methods,

which are based on kd-trees [29] and have been shown to perform well for point data. These methods aim at automatically generating an optimal partitioning of the entire multidimensional space yielding mutually disjoint sub-partitions. The second category includes the data partitioning methods, which are based on R-trees [15] and have been shown to perform well for hyper-rectangular data. Data partitioning methods do not subdivide the entire space but evaluate and store (possibly overlapping) hyper-rectangles that enclose the data to be partitioned. From the overview of the proposed scheme (Section 3.1) it is evident that data partitioning methods are more appropriate for constructing customly-defined hyper-neighborhoods, determined using training data. An experimental verification of this fact is provided in Section 4.3.2.

An R-tree [15] is a height-balanced tree with index records in its leaf nodes (containing pointers to data objects). Typically, R-trees index spatial objects using their Bounding Boxes (BBs). When a query is submitted the R-tree returns all records with BBs enclosing the query. In our case, since each image is represented by a d-dimensional feature vector, an R-tree structure can be constructed by associating a hyper-BB with each original image in the database. Selecting optimal hyper-BBs is crucial for the performance of the proposed replica detection system. Indeed, if the hyper-BBs are too large many of them overlap resulting in the retrieval of a large number of candidate originals and rendering the subsequent application of linear discriminant techniques ineffective. On the other hand, if the hyper-BBs are too small a replica is likely to fall outside the hyper-BB of its original image and will not be included in the response. Therefore, we employ training to define the size of the corresponding hyper-BB for each original image.

As already mentioned in Section 3.1 hyper-BBs are defined using an extent vector. In order to determine the extent vector for each original image I, we use the corresponding training matrix $\mathbf{T}_r^I$ that contains the feature vectors extracted from the training replicas of I. More specifically, if $\mathbf{x}_r^{I_k} = [x_{r,1}^{I_k}, \ldots, x_{r,d}^{I_k}]$ is the feature vector of the r-th training replica of the original image I_k, the hyper-BB for this image is defined by the vector $\mathbf{c}^{I_k} = [c_1^{I_k}, \ldots, c_d^{I_k}]$ which controls its extent for each dimension and is calculated as follows:

$$c_i^{I_k} = \max_r |x_{r,i}^{I_k} - x_i^{I_k}|, \qquad i = 1, \ldots, d \tag{4}$$

where $\mathbf{x}^{I_k} = [x_1^{I_k}, \ldots, x_d^{I_k}]$ is the feature vector of the original image I_k. The values that determine the boundaries for each dimension i are calculated by:

$$BE_{-,i}^{I_k} = x_i^{I_k} - c_i^{I_k}, \qquad BE_{+,i}^{I_k} = x_i^{I_k} + c_i^{I_k} \qquad i = 1, \ldots, d \tag{5}$$

The goal of this procedure is to find a hyper-rectangle that encloses the feature

vectors of all training replicas. The feature vector of a replica generated by a manipulation less severe than those used to build the R-Tree is expected to be enclosed in the BB associated to its original.

An inherent drawback of R-tree based methods is the so-called dimensionality curse which states that the computational gains in retrieval performance degrades exponentially as a function of dimensionality. For this purpose, we reduce the dimensionality of the original feature space by projecting the initial feature vectors (described in Section 3.2) on a fixed PCA (Principal Component Analysis) basis. We pre-calculate this basis by finding the principal components of the data space formed by the feature vectors corresponding to the total amount of database images and their training replicas. Given the large amount of samples, PCA manage to robustly detect the existing patterns in data and reduce the dimensionality of the indexed feature vectors without loosing much of the significant information. For the purposes of our work we reduce the feature space dimensionality to 24 dimensions in all cases, except for the ColorHistogram and ColorLayout descriptors that were left to their original 24 and 18 dimensions respectively. Concerning the R-tree branching factor, we have used $M = 8$ and $m = 4$, as the maximum and minimum number of allowed entries (i.e., children) in a node.

3.4 Achieving better class separability using Linear Discriminant Analysis

The fact that the R-tree may return more than one candidate images does not allow the system to decide unambiguously on the true original image. In such cases, in order to obtain a single result we propose the use of Linear Discriminant Analysis (LDA) [16] preceded by PCA. In the context of the proposed replica detection system PCA-LDA is applied as follows. Let $I = \{I_1, I_2, \ldots, I_K\}$ be the set of images returned by the R-tree. Considering that each image I_i and its training replicas define a class C_i, a set of classes $C = \{C_1, C_2, \ldots, C_K\}$ is dynamically formed every time a query is submitted. PCA is employed to find the principal components of the data space formed by the feature vectors corresponding to the images in I and their training replicas. These components are used to reduce the dimensionality of the initial feature vectors described in Section 3.2. Fishers discriminant criterion [30] is subsequently employed to define a new feature space that ensures better class separability between the classes of C than the original feature space.

The result of this analysis is a linear transformation matrix $\mathbf{W}_o$ that is used to project the initial feature vectors to the new feature space. Since PCA-LDA is applied after the R-tree traversal the number of classes is not known in advance. Thus, a new matrix $\mathbf{W}_o$ has to be calculated every time a new query is submitted to the system, each time resulting in a different projection

space. Although applying PCA-LDA on the fly may seem to hinder the process from the viewpoint of computational efficiency, it is necessary for allowing the proposed framework to achieve the best possible discrimination between the candidate classes and optimize replica detection on a per query basis. Moreover, as will be demonstrated in Section 4.3.7, the computational cost introduced by applying PCA-LDA prior to classification is marginal.

The reason for employing PCA prior to LDA is to maximize the discrimination power of the final feature space. In pattern recognition problems (especially in object and face recognition problems) PCA has been combined with LDA in several cases. The motivation for combining PCA with LDA was initially the singularity of the within class scatter matrix due to the Small Sample Size (SSS) problems that occur when the number of the training samples is smaller than the dimensionality of the samples [31], [32]. Theoretical work has been developed which proves that the PCA step is necessary in order to train LDA in SSS problems [33], [31], [32]. However, it has been also verified that by rejecting some dimensions that correspond to the eigenvectors of the total scatter matrix with small (in magnitude) eigenvalues (and not only the necessary ones so that the within scatter matrix is invertible) the classification performance of LDA is increased [34], [35]. Such an approach has been introduced in [34], [36] as the Enhanced Fisher's Linear Discriminant (EFLD) method. EFLD aims to seek a proper number of PCA components that balance between the need to keep enough spectral energy of raw data and the requirement that the eigenvalues of within-class scatter in the reduced PCA space are not too small. In our case, we have adopted a similar approach where the number of retained PCA components is the one that preserves 95% of the total variance.

An important consideration concerning the application of PCA on the data corresponding to the candidate classes, is whether the number of available samples is enough for learning the statistical properties of the dataset. As will become apparent in the experimental section, the number of samples that need to be handled by PCA-LDA in most of the cases is well above the number of dimensions of the employed feature space, which is the condition for robustly applying PCA. Indeed, if we accept that the average number of candidate images returned by the R-Tree is approximately 13 and taking into consideration that 40 training replicas are generated for each original image (see Section 4.1), the number of available training samples which is approximately 520 is well above the number of feature vector dimensions, even for the SCEH descriptor that exhibits the largest number of dimensions (208).

Moreover, our choice of applying LDA rather than some other state-of-the-art approach for classification was driven by the special requirement of dynamically resolving cases unsuccessfully handled by the R-tree. The power of the proposed approach lies on maximizing the efficiency of discriminant classi-

fiers by only having to cope with a relative low number of classes. However, this entails that training should always be performed on the fly based on the candidate classes returned by the R-tree. On the other hand, in order for a replica detection system to be useful in real applications it should be able to exhibit low response time. This poses strict limitations on the computational complexity of the employed training method. The complexity properties that characterize LDA was the main reason for choosing this approach over other state-of-the-art solutions. Indeed, the complexity of LDA training is dominated by the calculation of the within class scatter matrix and its inverse, which is $O(d^2n)$ with d being the feature space dimensionality and n the total number of training samples. An important characteristic of this complexity is that the multiplication factor of n depends on the number of feature space dimensions. This is particularly desirable since d depends exclusively on the system's configuration settings and is not affected by the number of images accommodated by a replica detection system. This is not the case for other discriminative classification approaches. Let us consider for example the case of Support Vector Machines (SVM) which are considered to deliver state-of-the-art performance in real world pattern recognition problems. Using the standard training procedure the computational complexity of training depends on the number of necessary support vectors n_{SV} and is $O(n \cdot n_{SV} + n_{SV}^3)$. Driven by the theoretical result of Steinwart [37] who showed that n_{SV} grows as a linear function of n, it is clear that in contrast to the previous case the multiplication factor of n depends on the number of accommodated images. Since the amount of images (and as a consequence n) that needs to be handled by a replica detection system can be arbitrary big, we decided to opt for a classification approach the complexity of which behaves optimally with respect to the number of accommodated images.

3.5 Classification Function

By projecting the members of C to the new feature space derived from the maximization of Fisher's criterion, we obtain $\acute{C}$ where better class separability is expected. Since we require the system to strictly return one original image or an empty set, we need to define a classification function that will deal with this issue in the new feature space. For each class $\acute{C}_i$ representing a candidate original I_i we calculate the mean vector $\bar{\acute{\mathbf{x}}}^{I_i}$ (class center) and a threshold $\acute{T}_{I_i}$ that defines its new neighborhood as:

$$\acute{T}_{I_i} = \max_{r=1,\ldots,M_i} (\parallel \acute{\mathbf{x}}_r^{I_i} - \bar{\acute{\mathbf{x}}}^{I_i} \parallel_2) \tag{6}$$

where $||.||_2$ denotes the L_2 norm. The response of our system to the query image I_q is determined by the following function.

$$D(I_q) = \begin{cases} \emptyset, \ \| \acute{\mathbf{x}}^{I_q} - \bar{\acute{\mathbf{x}}}^{I_i} \|_2 > \acute{T}_{I_i} \quad \forall i \in [1, \ldots, K] \\ I_r, \ r = \arg\min_i (\| \acute{\mathbf{x}}^{I_q} - \bar{\acute{\mathbf{x}}}^{I_i} \|_2) \quad \& \quad \| \acute{\mathbf{x}}^{I_q} - \bar{\acute{\mathbf{x}}}^{I_r} \|_2 < \acute{T}_{I_r} \end{cases} \tag{7}$$

where I_r is the member of the database that is considered as the original of the query image I_q. The reason for incorporating a threshold on the classification function was to provide the system with a criterion for rejecting non-replica images, which was also the reason for employing the SIFT variant of the system detailed in the following section.

3.6 SIFT variant of the proposed system

In certain cases, histogram-based descriptors do not contain enough information to discriminate between simply similar images and images that are connected with a replica-original relation. As a consequence, a non-replica image might be included in the neighborhood of a similar original image and be erroneously characterized as its replica. False replicas constitute a hard problem for replica detection systems that can only be confronted using a highly discriminative feature extraction algorithm. In order to overcome this deficiency, we enhanced our system functionality by incorporating an additional module that involves feature vectors generated using the SIFT [14] algorithm. SIFT is based on detecting highly distinctive, scale and rotation invariant key-points and describing them using 128-dimensional orientation histograms. The resulting representation constitutes a Kx128 matrix, where K is the number of identified key-points. Although, it is known to produce distinctive keypoints that exhibit robustness against a substantial number of image manipulations, it was not possible to incorporate SIFT as the basic feature extraction scheme, due to their high computational cost and non-compliance with the fixed-length global representation requirement.

The reason for using SIFT is not to replace the main feature extraction scheme, but to add a verification step before producing the final output. In more detail, prior to reaching a final decision the SIFT feature extraction and matching scheme [14] is used to assess the similarity between the query and the database image that has been selected by the classification function. Similarity assessment is performed by individually comparing the 128-dimensional feature vector of each keypoint from the query image to the feature vectors of all keypoints of the selected original. A match is counted every time the Euclidean distance between the feature vectors of two keypoints is below a certain threshold determined by Lowe in [14]. Eventually, if the number of matches exceeds a threshold, equal to one tenth of the total number of key-

points identified in the original image, the system response is validated as correct. Otherwise, the system initial suggestion is rejected and the query image is characterized as non-replica. Although the experimental results show that SIFT can be a very effective countermeasure against falsely accepting non-replicas as replicas, the total time required for executing a query with this setup is considerably increased (see Section 4.3.7).

3.7 Security Considerations

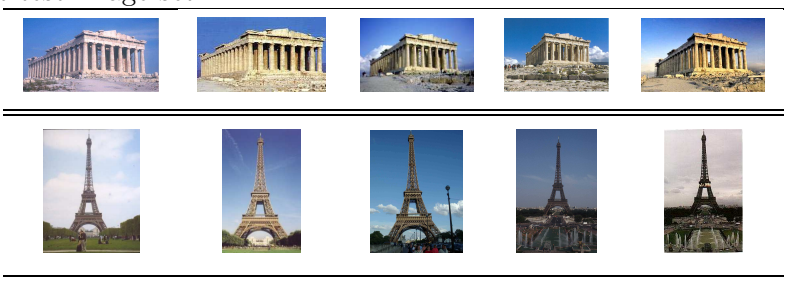
Since the target applications of a replica detection system might include ownership identification and content-based media monitoring for legitimate use, one should expect that intentional attacks coming from an adversary that tries to hinter its functionality would occur. Two different types of intentional attacks can be encountered. The first aims at producing false negatives by intentionally modifying the content of a protected image in order to go undetected (false negative attacks). The second type of attacks includes those actions that generate false positives and cripple the reliability of system by trying to forge, for example, a protected image through the modification of an arbitrary image (false positive attacks). In accordance to Kerckhoff's principle, the attacker is expected to have full knowledge of the protection mechanism details.

With respect to false negative attacks, the factor limiting the attacker actions is related to the amount of distortion that should be introduced in the protected image in order to render it undetectable. The fact that our feature vector is histogram-based, eases the task of an adversary in producing false negatives since he knows the elements he should focus on in order to achieve his goal. However, the fact that SCEH (which was experimentally selected as being the most appropriate for our system, see Section 4.3.1) combines both color and edge histograms, renders difficult the creation of a false negatives with sufficient quality even if the algorithm details are known. Moreover, the fact that the proposed system is constructed so as to be robust to a wide range of manipulations (see Section 4.1), hinders the task of an adversary to create false negatives without severely distorting the image. In what refers to false positive attacks, the requirements are similar to those imposed by the collision-free property of hash functions. This property refers to the fact that, given an image I and a hash function $g(\cdot)$, it is computationally hard to find a second image $\acute{I}$ such that $g(I) = g(\acute{I})$. The fact that the SCEH detector involves both color and edge information along with the fact that a final verification step based on SIFT descriptors is utilized by our algorithm, makes the creation of forged originals through the modification of a protected image difficult and time-consuming.

Table 1
Monument Test Image Set

Monument Name	Parthenon (Greece)	White Tower (Greece)	Liberty Statue (USA)	Sagrada Familia (Spain)	Lighthouse (Arbitrary Photos)	Coliseum (Italy)
#Images	343	78	71	247	56	128
Monument Name	Eiffel Tower (France)	Piza Tower (Italy)	Pyramids (Arbitrary Photos)	Sphinx (Egypt)	Duomo (Florence-Italy)	Big Ben (England)
#Images	233	78	63	193	266	476

Table 2
Sample images from two content categories (Parthenon and Eiffel tower) in the Monument test image set



4 Experimental Study

4.1 Test Set Characteristics

Prior to presenting the results, it is necessary to describe the particular characteristics of the experimental testbed. Two image sets were used in experiments. A sample of 2.232 color images were downloaded from the Internet to compose the first set, from here on referred as the Monument set. Images were selected so as to form 12 content categories, each containing different views of a famous monument, as shown in Table 1. The rationale was to construct a test set consisting of images featuring high perceptual and/or semantic similarity within each category. Table 2 depicts samples from two of these categories. This selection strategy was dictated by the high dependency between the performance of a replica detection system and the level of visual similarity among the database members. Evaluation against such a challenging test set was performed in an effort to assess its behavior under an unfavorable situation and introduce a sense of fairness compared to other technological approaches (e.g., watermarking), whose performance is largely unaffected by the image content. Moreover, in order to validate the efficiency of our system on a much larger database we applied the optimally configured replica detection system on a portion of the Corel database containing 9.908 images. The reason for choosing Corel as our second test corpus was to be in accordance with the aforementioned selection strategy, since this collection was originally constructed to form groups of pictures depicting the same theme.

For training, we generated manipulated copies for each original image by applying the following 40 transformations. a) Colorizing: Colorize the Red, Green, and Blue channel by 10% by blending the fill color with each pixel

in the image, b) Contrast Changes: Increase or decrease the intensity differences between the lighter and darker elements using the default parameter provided by ImageMagick, c) Cropping: Symmetrically remove the outer borders of an image to reduce its size by 5%, 10%, 20%, and 30% and then scale the cropped image back to its original size, d) Despeckling: The amount of speckle noise is reduced through ImageMagick's despeckling operation while preserving the edges of the original image, e) Downsampling: Downsample by seven percentages 10%, 20%, 30%, 40%, 50%, 70%, 90%, f) Flopping: Create a mirror image by reflecting the scanlines along the horizontal direction, g) Color Quantization: Reduce the color palette to 256 colors, h) Framing: Four framed images are produced by adding an outer frame covering 10% of the total image area. A different frame color is utilized for each image, i) Rotation: Rotation by 90^o, 180^o, and 270^o, j) Scale up then down by a factor of 2,4, and 8. Respectively scale down then up by a factor of 2,4, and 8, k) Saturation Change: Modulate the color saturation amplitude by 70%, 80%, 90%, 110%, and 120%, l) Intensity Change: Modulate the image intensity by 80%, 90%, 110%, and 120%. These manipulations were initially proposed by Meng et al. in [38].

4.2 Evaluation Metrics

For evaluating the performance of the proposed system, the false positive and false negative rates were considered. In the context of an image replica detection system a false positive occurs when a query image is erroneously considered to be a replica of a certain image. This includes both the case of a non-replica image being identified as a replica, as well as the case where a replica image is identified as such but is associated with a wrong original. Respectively, the system produces a false negative when a query image that is a replica of a certain original is not evaluated as such. Let N_{org} be the number of original images, N_{nrep} the number of non-replicas and N_{rep} the number of replicas per original image. Let also T be the number of cases that a replica is identified as such but is classified to a wrong original, W be the number of cases that a non-replica is evaluated as a replica and S the number of cases that a replica is considered as a non-replica. Then false positive and false negative rates are defined as, $FP = \frac{T+W}{N_{org} \cdot N_{rep} + N_{nrep}}$ and $FN = \frac{T+S}{N_{org} \cdot N_{rep}}$. Recall (R) and Precision (Pr) are two other well established metrics that are commonly used in the area of image retrieval but have been also considered in image replica detection. Using the notations described above recall and precision are defined as, $R = \frac{(N_{org} + N_{org} \cdot N_{rep} + N_{nrep}) - (T+W+S)}{N_{org} + N_{org} \cdot N_{rep} + N_{nrep}}$ and $Pr = \frac{(N_{org} + N_{org} \cdot N_{rep} + N_{nrep}) - (T+W+S)}{N_{total}}$, where N_{total} is the total number of results produced by the system.

Although, false positive and false negative rates were selected for measuring

the efficiency of the proposed system, in order to allow comparisons with other schemes, recall and precision were also evaluated taking into account that due to the adopted configuration $N_{total} = N_{org} + N_{org} \cdot N_{rep} + N_{nrep}$ and thus $R \equiv Pr$. Receiver Operating Characteristic (ROC) curves that are commonly used to represent the tradeoff between FP and FN were used for measuring the system's performance in experiments that involved a tunable system parameter. The equal error rate (EER), i.e., the point of the ROC where FP=FN, was also used as an indicator of the system's performance.

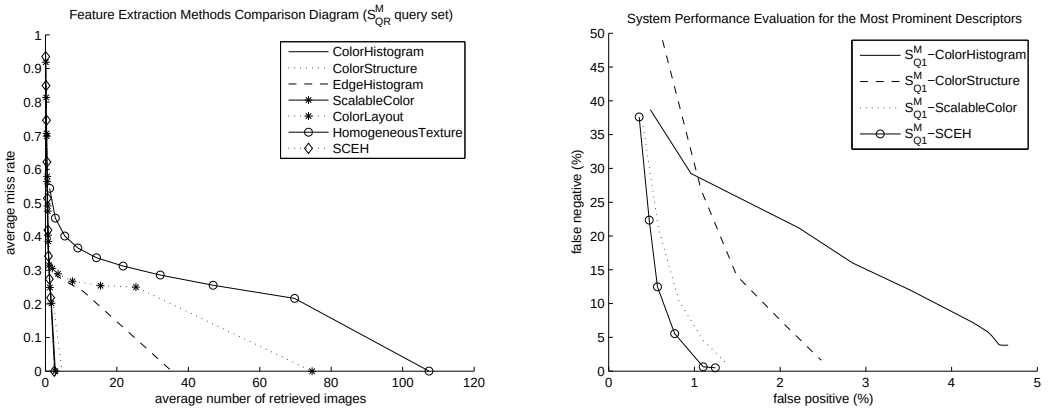
4.3 Experiments on the Monument Set

Out of the 2232 images included in the Monument set, we selected a set S^M_{org} of 2000 for populating the original image database while the remaining 232 formed the set S^M_{nrep} of non-replicas. The query set S^M_{Q1} was constructed of original images, test replicas and non-replicas. 200 images randomly chosen from S^M_{org} were used to compose the set of original images S^M_{seed} that was included in the query set S^M_{Q1}. These images were also used to generate the set S^M_{rep} of 8000 test replicas. This set was generated by applying the transformations described in Section 4.1 to the images in S^M_{seed} (40 transforms per original image). Finally, S^M_{nrep} was appended to the other test images resulting in a query set containing a total of 8432 images, $S^M_{Q1} = S^M_{seed} \cup S^M_{rep} \cup S^M_{nrep}$.

4.3.1 Evaluation of Feature Extraction Methods

For evaluating the performance of the various extraction methods presented in Section 3.2, we used the R-tree to measure the average miss rate (i.e., the probability that the R-tree fails to retrieve the correct original) against the average number of retrieved images. For generating the evaluation curves we varied the extent of the R-tree hyper-neighborhoods by multiplying their boundaries with a scaling factor σ_R. Retrieving a relatively small number of candidates is crucial for the proposed system efficiency, since LDA will not manage to attain good class discrimination on the feature space if the number of participating classes is large. Given the fact that it is far more important for the R-tree not to miss any real replicas than to retrieve more than one candidates, we are interested in the point where zero miss rate is achieved. It is important to notice that the query set S^M_{QR} used in this experiment is different from S^M_{Q1} in the sense that non-replica images are not included, $S^M_{QR} = S^M_{seed} \cup S^M_{rep}$.

As demonstrated in Fig.3a the average number of retrieved images for zero miss rate differs substantially between the various types of features. The SCEH descriptor, incorporating both color and edge elements, achieves the lowest av-



(a) (b)

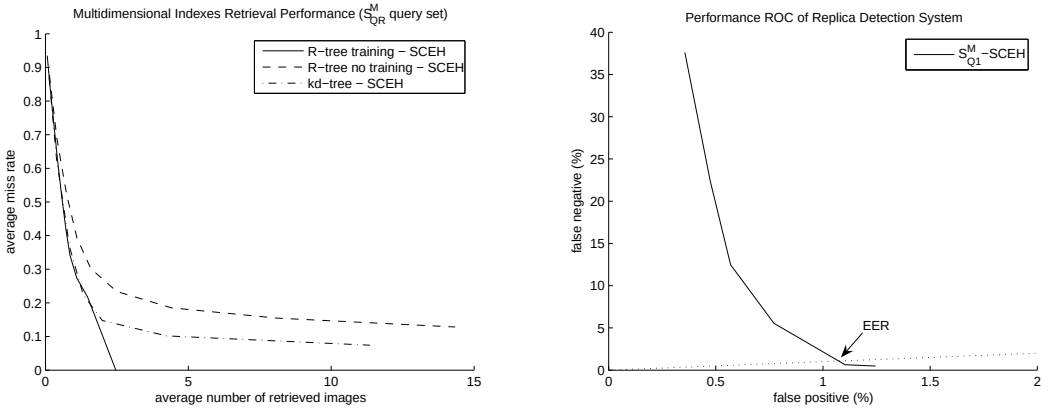
Fig. 3. a) Comparative diagram of all feature methods using S^M_{QR} as query b) Auxiliary experiment evaluating the overall system performance for the most prominent of the examined descriptors using S^M_{Q1}

erage number of retrieved images (≈ 2.4) for zero miss rate and was adopted as the feature extraction method throughout our experimental study. The superiority of SCEH features was also verified on experiments involving the performance of the complete image replica detection system. The ROCs depicted in Fig.3b that compare the performance of the most prominent descriptors, prove that SCEH outperforms the other features extraction schemes. The regulation parameter used for drawing the ROCs of Fig.3b was a scaling factor σ_{LDA} changing the classification neighborhoods formulated in the LDA-transformed feature space by multiplying the class threshold $\acute{T}_{I_i}$ used in equations (6) and (7).

4.3.2 Data Partitioning vs Space Partitioning Methods and the Influence of Training

In order to verify that data partitioning outperforms space partitioning in the context of repica detection, we compared the performance of the basic representatives from the two categories, namely R-tree and kd-tree. The same experiment attempts to highlight the benefits of selecting optimal hyperneighborhoods using the training matrices $\mathbf{T}_r$, as described in Section 3.3. In order to do this we examine the case where no training is involved, i.e., a constant extent value denoted as v is utilized for all feature dimensions n, and for all database images. As in the previous case S^M_{QR} was utilized for testing and the average number of retrieved images required to achieve zero miss rate was used as the performance criterion.

For generating the performance curve in the training case, we varied the extent



(a) (b)

Fig. 4. a) Data partitioning vs Space partitioning methods using S^M_{QR} as query set b) Replica Detection Performance ROC using S^M_{Q1} as query set

of the hyper-neighborhoods by multiplying the elements of the extent vector c^I_i with a scaling factor σ_R, while in the other case the value of v was modified. The kd-tree performance curve was obtained by varying the threshold σ_{kd} of the Euclidean distance between the query image and the ones already indexed within the multidimensional structure. The results are depicted in Fig. 4a. It is clear that although kd-tree performance (dashed-dotted curve) is superior from the R-tree when no training is involved (dashed curve), it is considerably outperformed by the R-tree constructed using the training samples of $\mathbf{T}_r$ (solid curve). The solid curve shows that in order to construct a system with miss rate 0% (which is crucial for obtaining small false negative rate), we should allow the "trained" R-tree to retrieve approximately 2.4 images per query, on average. For the other cases on the other hand, in order to achieve zero miss rate the average number of returned originals should let to grow very large (the two curves converge towards the horizontal axis very slowly).

4.3.3 System performance when the training and query sets are identical

This experiment is aimed at measuring the performance of the proposed system using $\sigma_R = 1$ for the R-tree. In this case, the tuning parameter used to create the ROC presented in Fig. 4b. is the scaling factor σ_{LDA} described in Section 4.3.1. The image set S^M_{Q1} that contains the same replica images as the ones used for training, was used as a query set. The EER of 1.1% is obtained when the regulation parameter σ_{LDA} is equal to one, which suggests that training has indeed selected optimal neighborhoods for the classification function of Section 3.5.

Table 3
Manipulations used to construct the set of replicas $\acute{S}^M_{rep}$ utilized in the query set S^M_{Q2}

Framing	Colozing	Color Quantization	Downsampling	Contrast Change	Cropping
9%	6%	as previous	19%, 28%, 38% 41%, 52%, 80%, 88%	as previous	6%, 8% 11%, 22%
Despeckling	**Flipping**	**Intensity Change**	**Rotation**	**Saturation Change.**	**Scaling**
as previous	vertical	93%, 98% 108%, 116%	95^o, 183^o 268^o	75%, 85% 95%, 115%	3, 5 7

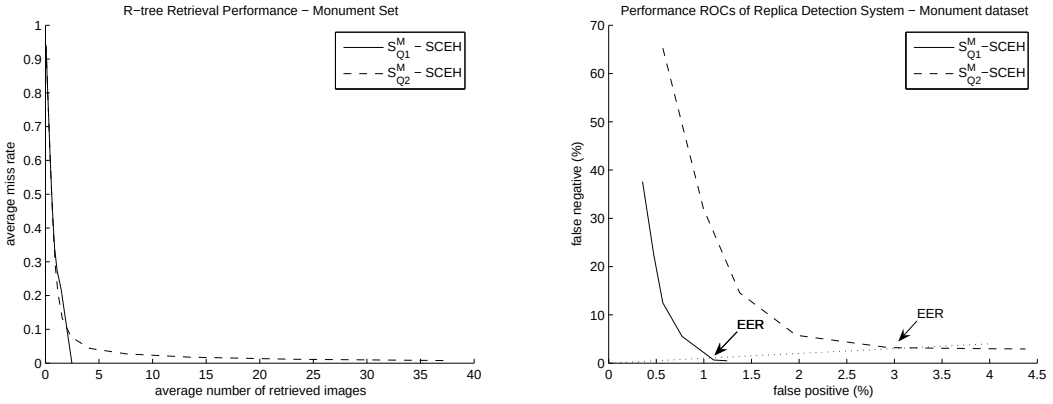
4.3.4 Employing a query set different than the one used for training

The fact that the set of images used for training was exactly the same with the set of replicas included in the query set may lead to biased performance evaluation. For assessing the system performance more rigorously, a query set that includes manipulated images that were not used during training was constructed. In order to produce the new query set we utilize S^M_{seed} and S^M_{nrep} but instead of S^M_{rep} we used a different set of replicas that was generated by exposing the original images of S^M_{seed} to the same type of attacks, but with different attack parameters. The new attack parameters depicted in Table 3 (40 manipulations per original image), were chosen so that they reside inside the parameter range used for constructing the training set. The resulting replicas $\acute{S}^M_{rep}$ were combined with the original images and non-replicas in order to produce the new query set, $S^M_{Q2} = S^M_{seed} \cup \acute{S}^M_{rep} \cup S^M_{nrep}$.

Fig. 5 depicts the R-tree retrieval performance as well as the overall system's performance when S^M_{Q2} is used. For comparison purposes the curves corresponding to S^M_{Q1} are also drawn. Examination of the R-tree performance in Fig. 5a shows that unlike the case where the query images were the same as those used for training, the R-tree is difficult to achieve a zero miss rate while maintaining a relatively small number of retrieved images. Therefore, instead of rendering PCA-LDA ineffective due to the increased number of participating classes, one can fix the size of the neighborhoods so that the miss rate of the R-tree is acceptable but no zero. The miss rate was fixed to 0.027 and the average number of images retrieved for this value is approximately 13. Fig. 5b shows replica detection performance for S^M_{Q1} and S^M_{Q2} query sets. The EER obtained for S^M_{Q2} is equal to 3.0%, not significantly worse from the one obtained when using S^M_{Q1}.

4.3.5 Employing the SIFT module

As already mentioned in Section 3.6 falsely accepting non-replicas as replicas constitutes a hard problem for replica detection systems. In order to check this experimentally we have recorded the performance of our system using strictly non-replica images as queries. We did so by using S^M_{nrep}, that only consists of non-replica images, for testing. The value of the scaling factor was set to $\sigma_{LDA} = 1$, so as to tune our system to the operating point where



(a) (b)

Fig. 5. System Performance for query sets S^M_{Q1} and S^M_{Q2} a) R-tree retrieval performance b) ROC curves for the overall system performance

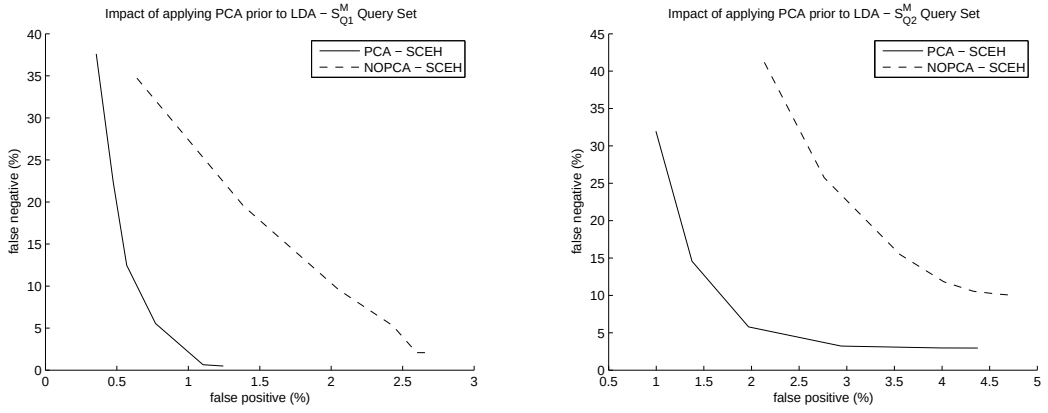
Table 4
System Error Rates - Monument dataset

	WITHOUT-SIFT		**SIFT**	
	S^M_{Q1}	S^M_{Q2}	S^M_{Q1}	S^M_{Q2}
FN	1.1%	3.0%	1.1%	3.2%
FP	1.1%	3.0%	0.58%	1.9%

EER is attained. The experiments showed that 23.28% of the non-replica images were erroneously identified as replicas. However, after incorporating the SIFT variant described in Section 3.6 the percentage of falsely accepted non-replicas reduces to 3%. The overall impact of appending the SIFT module to the system is depicted in Table 4. The decrease in the false positive rate stems mainly from the drastic decrease of errors in case of queries with non-replicas. Specifically, given that the proportion of non-replica images in the both query sets is approximately 2.5%, the improvement from 23% to 3% recorded against the images of S^M_{nrep}, translates to an improvement of the false positive rate from 1.1% to $0,58\%$ when testing with S^M_{Q1} and from 3.0% to 1.9% when testing with S^M_{Q2}. However, in the case of S^M_{Q2} the use of the SIFT module results in a small increase ($\approx 0.2\%$) of the false negative rate, which is nevertheless smaller than the corresponding decrease of the false positive rate.

4.3.6 Combining PCA with LDA

The goal of this experiment was to verify the improvement in performance introduced by employing PCA prior to LDA. Fig. 6. demonstrates the performance curves for both query sets S^M_{Q1} and S^M_{Q2} with and without applying PCA. One can see that in both cases the system performance benefits substantially



(a) (b)

Fig. 6. Optimizing performance using PCA prior to LDA a) S^M_{Q1} Query Image Set b) S^M_{Q2} Query Image Set

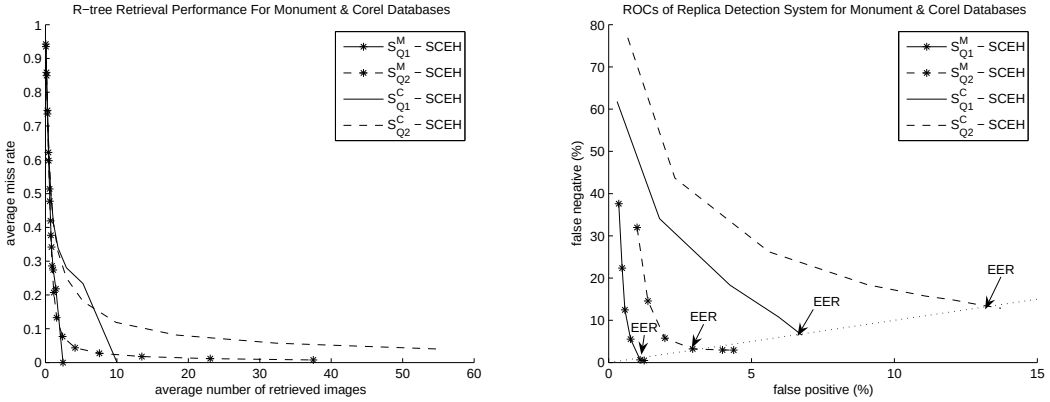
from the use of PCA.

4.3.7 Computational Time

Table 5 presents the computational time spent during each phase of the query procedure. All experiments were conducted on the Monument set using an Intel Pentium-M/Centrino processor, running at 1.86Ghz with 1.00 GB of RAM. An average of 0.0305 sec is required by the overall system to handle a single query, when the SIFT module is not incorporated, which can be considered satisfactory even for real-time applications. Obviously, the time required for traversing the R-tree is influenced both by the number of database images and the amount of overlapping between their hyper-rectangles. Respectively, LDA is more time consuming when the number of participating classes increases. However, since R-tree is logarithmic to the number of indexed images and taking into consideration that LDA operates only on the small number of classes returned by the R-tree, it is safe to conclude that the increase in computational time induced by the growing number of database images, will not render the proposed system impractical even for large scale applications. When the SIFT module is incorporated to the system the total execution time for a single query increases to 11.756 sec. This amount of time is prohibitive for real time applications but can be tolerated for applications like off-line copyright infringement detection. Additionally, since the SIFT module always operates on just two images (i.e., candidate original and query) the average time consumed by this module is unaffected by the amount of database images.

Table 5
Query Execution Time (Monument database)

Query Image Set	**Time Required Per Query (sec)**				
	R-tree	LDA	SIFT	Total (NO SIFT)	Total (SIFT)
S^M_{Q1} Query Image Set	0.003	0.015	11.738	0.018	11.756
S^M_{Q2} Query Image Set	0.005	0.038	11.815	0.043	11.858
Average	0.004	0.0265	11.776	0.0305	11.807



(a) (b)

Fig. 7. Evaluating the optimally configured replica detection system using S^M_{Q1}, S^M_{Q2}, S^C_{Q1}, S^C_{Q2} a) R-tree retrieval performance b) Overall replica detection system performance

4.4 Experiments on the Corel Image Set

After evaluating each module independently and fine-tuning the proposed replica detection system, a portion of the Corel image collection was utilized to inquire the efficiency of our approach when the number of database images increases. From the Corel set we utilized 9.908 images depicting 120 different themes. These images were divided into 9000 originals S^C_{org}, and 908 non-replicas S^C_{nrep}. 2000 images were drawn from S^C_{org} to form the seed image set S^C_{seed}, while the same strategy was followed to construct $\mathbf{T}_r$, S^C_{Q1} and S^C_{Q2}, consisting of 360000, 82908 and 82908 images respectively. The performance curves for the Monument set are also included in the diagrams to allow comparisons.

It is clear from Fig. 7a, that as the number of database images increases, the feature space becomes more crammed and the efficiency of our system is affected. Using S^C_{Q1}, the average number of retrieved images returned by the R-tree in order to achieve zero miss rate increases to 10 from the corresponding 2.4 in the Monument set case. Respectively, the curve showing the R-tree performance for S^C_{Q2} shows that we will have to let the average number of

Table 6
System Error Rates - Corel dataset

	WITHOUT-SIFT		SIFT	
	S^C_{Q1}	S^C_{Q2}	S^C_{Q1}	S^C_{Q2}
FN	6.5%	13%	6.59%	15.74%
FP	6.5%	13%	6.58%	10.89%

retrieved images grow at approximately 30 images for achieving a miss rate of 0.057.

Similar conclusions can be derived by inspecting the diagram of Fig.7b where the performance curves of the overall system without incorporating the SIFT module are depicted. The EER achieved for S^C_{Q1} is approximately 6.5 % and it grows to 13% when the system is evaluated using S^C_{Q2}. The impact of incorporating the SIFT variant is shown in Table 6 where the performance results for the Corel dataset are summarized. Although the dependence between the performance of our system and the number of database images is clear, recalling that both image sets were intentionally constructed to feature a high degree of similarity between their members, it is reasonable to claim that the proposed scheme can be safely used for detecting replicas with sufficient accuracy.

4.5 Performance Evaluation Review of Existing Replica Detection Systems

Comparing the performance of different methods addressing the same problem is a difficult task particularly when the associated research community lacks a standardized benchmarking methodology. Inconsistencies regarding the experimental test-bed configuration, the utilized data set, the variety in robustness tests and performance metrics need to be overcome before drawing safe conclusions. This subsection is an effort to review the performance figures achieved by different replica detection systems. However, readers should have in mind that due to the different testbed configurations and sets of original images used by the various authors, the presented performance results are not directly comparable and should be treated as such.

In their work, Qamra et al [6] utilize original and modified versions of copyright protected images for populating the test database. Training and querying are both based on the same set of manipulations proposed by Meng et al in [38]. The method's efficiency is measured using recall and precision and the equally balanced tradeoff is ≈ 0.82. On the other hand, Maret et al [10] choose a configuration where only the original versions of the copyright protected images are included in the database. The set of modifications proposed in [38] is used for training the classifier and for generating test replicas. Performance is measured in terms of false positive and false negative rates and according to the authors, the method is able to detect, on average, 92% of the replicas while achieving a fixed false positive rate of only $1 \cdot 10^{-4}$. An important limitation of

this system is that a different classifier is trained for every original image in the database, thus, in the worst case scenario the query image has to be evaluated against all classifiers before reaching a decision. As a consequence considerable amount of time might be required for handling a query as the number of database images grows. In some of their subsequent works [11], [12] the authors try to alleviate this problem by employing multidimensional indexing schemes. The database configuration that Ke et al [9] use in their system is similar to that of [6] where original images and their modified versions coexist. No training is required by the method and the authors employ two different image manipulation sets for evaluating the performance. The manipulation set of [38] and a more challenging set of transformations that includes cropping by 50%, 70% and 90%, shearing by 5^o, 10^o and 15^o, changes of intensity by 50% and 150% and changes in contrast (10 transformations per image). The method achieves 99.85% recall and 100% precision and these figures decrease to 98.40% and 99.86% for the more challenging manipulation set. However, the number of features extracted from every image ranges from a few hundreds to a few thousands and the time required for their extraction is considerably large. A similar test-database configuration is adopted by Kim [7]. For training, the author uses a set that includes various modifications while a subset of this set is used for testing. The evaluation experiments showed that the system achieves, on average, 83% recall and 96% precision. Roy and Chang in [8] implement a database containing only the original versions of copyright protected images. For such a setup recall and precision coincide and the common figure provided by the authors is 96.8%. Synthetic training examples are constructed by adding Gaussian noise in the feature domain and the query set is produced using a subset of the Meng set [38] (9 transformations per image). An important difference of this method compared to the other experimental setups is that no non-replica images are included in the query set.

The proposed system adopts the configuration where only the original version of the images are stored in the database. Performance figures for the Monument set have been calculated both in terms of false positive and false negative as well as recall and precision metrics. Note that in the latter case, due to the database configuration, the two metrics have the same value. Table 7 summarize the results obtained by the various methods evaluated in terms of recall-precision and false positive-false negative rates respectively.

5 Conclusions

In this manuscript, we describe a replica detection system that operates upon a database of stored originals. Motivated by the fact that replica detection has many common characteristics with a classification problem, we worked towards the employment of proper training strategies for improving efficiency.

Table 7
Performance Figures Review of Existing Replica Detection Systems

Method	Image Database Configuration	Image Manipulation Set	Relation between Train and Query Set	Recall (%)	Precision (%)
Qamra [6]	Original and Modified	Meng [38]	Identical	82	82
Ke [9]	Original and Modified	Meng [38]	No Training	99.85	100
		Meng Extension		98.40	99.86
Kim [7]	Original and Modified	Miscellaneous	Subset	83	96
Roy [8]	Original	Subset of Meng	Identical	96.8	96.8
Proposed Method	Original (Monument)	Meng	Identical	98.73	98.73
			Different	95.79	95.79
Proposed Method (SIFT)	Original (Monument)	Meng	Identical	99.25	99.25
			Different	96.74	96.74
				False Negative (%)	**False Positive (%)**
Maret [10]	Original	Meng	Identical	8	0.01
Proposed Method	Original	Meng	Identical	1.1	1.1
			Different	3.0	3.0
Proposed Method (SIFT)	Original	Meng	Identical	0.65	0.58
			Different	3.22	1.99

This training strategy is used to drive both image indexing conducted using an R-tree and the construction of robust classifiers in a transformed feature space. This feature space is generated by dynamically applying PCA-LDA on the candidate classes produced by the R-tree. The power of our approach lies on maximizing the efficiency of discriminant classifiers by only having to cope with a relative low number of classes. Two very challenging image sets were used in our experimental study. Although, the obtained performance figures reveal some dependency on the size of the dataset, they can be considered rather satisfactory for the purposes of replica detection.

Acknowledgment

This work was developed within ECRYPT IST-2002-507932, European Network of Excellence in Cryptology (http://www.ecrypt.eu.org/), funded under the European Commission IST FP6 programme.

References

[1] A. Tefas, N. Nikolaidis, I. Pitas, Watermarking techniques for image authentication and copyright protection, 2nd Edition, The Handbook of Image and Video Processing, Elsevier, 2005.

[2] A. Swaminathan, Y. Mao, M. Wu, Image hashing resilient to geometric and filtering operations, in: IEEE Workshop on Multimedia Signal Processing (MMSP'04), Siena, Italy, 2004, pp. 355–358.

[3] A. Joly, O. Buisson, C. Frelicot, Content-based copy retrieval using distortion-based probabilistic similarity search, IEEE Transactions on Multimedia, 9 (2) (2007) 293–306.

[4] B. Coskun, B. Sankur, N. Memon, Spatiotemporal transform based video hashing, IEEE Transactions on Multimedia 8 (6) (Dec. 2006) 1190–1208.

[5] P. Cano, E. Batle, T. Kalker, J. Haitsma, A review of algorithms for audio fingerprinting, IEEE Workshop on Multimedia Signal Processing (9-11 Dec. 2002) 169–173.

[6] A. Qamra, Y. Meng, E. Chang, Enhanced perceptual distance functions and indexing for image replica recognition, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2005) 379–391.

[7] C. Kim, Content-based image copy detection, Signal Processing: Image Communication 18 (2003) 169–184.

[8] S. Roy, E.-C. Chang, K. Natarajan, A unified framework for resolving ambiguity in copy detection, in: Proceedings of the 13th annual ACM international conference on Multimedia, 2005, pp. 648–655.

[9] Y. Ke, R. Sukthankar, L. Huston, An efficient parts-based near-duplicate and sub-image retrieval system, in: Proceedings of the 12th annual ACM international conference on Multimedia, 2004, pp. 869–876.

[10] Y. Maret, F. Dufaux, T. Ebrahimi, Adaptive image replica detection based on support vector classifiers, Signal Processing : Image Communication 21 (8) (2006) 688–703.

[11] Y. Maret, S. Nikolopoulos, F. Dufaux, T. Ebrahimi, N. Nikolaidis, A novel replica detection system using binary classifiers, r-trees, and pca, in: ICIP, 2006, pp. 925–928.

[12] Y. Maret, D. Marimon, F. Dufaux, T. Ebrahimi, Hierarchical Indexing using R-trees for Replica Detection, in: SPIE, Lecture Notes in Computer Science, 2006.

[13] Y. Ke, R. Sukthankar, Pca-sift: a more distinctive representation for local image descriptors, in: Proceedings IEEE Computer Vision and Pattern Recognition, 2003.

[14] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vision 60 (2) (2004) 91–110.

[15] A. Gutmann, R-trees: a dynamic index structure for spatial searching, in: Proc. ACM International Conference on Management and Data (SIGMOD'88), Siena, Italy, 1988, pp. 47–57.

[16] K. Fukunaga, Introduction to Statistical Pattern Recognition, CA: Academic, San Diego, 1990.

[17] S. Nikolopoulos, S. Zafeiriou, P. Sidiropoulos, N. Nikolaidis, I. Pitas, Image replica detection using r-trees and linear discriminant analysis, in: IEEE International Conference on Multimedia and Expo (ICME '06), 2006, pp. 1797–1800.

[18] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Trans. Pattern Anal. Mach. Intell. 22 (12) (2000) 1349–1380.

[19] M. A. Gavrielides, E. Sikudova, I. Pitas, Color-based descriptors for image fingerprinting, IEEE Transactions on Multimedia 8 (4) (2006) 740–748.

[20] B. S. Manjunath, J. R. Ohm, V. V. Vinod, A. Yamada, Colour and texture descriptors, IEEE Trans. Circuits and Systems for Video Technology, Special Issue on MPEG-7 11 (6) (2001) 703–715.

[21] M. Bober, Mpeg-7 visual shape descriptors, IEEE Trans. Circuits and Systems for Video Technology, Special Issue on MPEG-7 11 (6) (Jun 2001) 716–719.

[22] C. McCamy, H. Marcus, J. Davinson, A color-rendition chart, Journal of Applied Photographic Engineering 2 (3) (1976) 95–99.

[23] A. Kasutani, E.; Yamada, The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval, Image Processing, 2001. Proceedings. 2001 International Conference on 1 (2001) 674–677 vol.1.

[24] D. Messing, P. van Beek, J. Errico, The mpeg-7 colour structure descriptor: image description using colour and local spatial information, International Conference on Image Processing, 2001. 1 (2001) 670–673 vol.1.

[25] M. Eom, Y. Choe, Fast extraction of edge histogram in dct domain based on mpeg7, in: International Conference on Enformatika, Systems Sciences and Engineering (ESSE 2005), Istanbul, Turkey, 2005.

[26] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intell. 8 (6) (1986) 679–698.

[27] Y. M. Ro, M. Kim, H. K. Kang, B. Manjunath, J. Kim, Mpeg-7 homogeneous texture descriptor, ETRI Journal 23 (2) (2001) 41–51.

[28] B. Nam, A. Sussman, A comparative study of spatial indexing techniques for multidimensional scientific datasets, Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on (21-23 June 2004) 171–180.

[29] J. L. Bentley, Multidimensional binary search trees used for associative searching, Commun. ACM 18 (9) (1975) 509–517.

[30] R. A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.

[31] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.

[32] D. L. Swets, J. Weng, Using discriminant eigenfeatures for image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 831–836.

[33] J. Yang, A. F. Frangi, J. yu Yang, Z. Jin, Kpca plus lda: A complete kernel fisher discriminant framework for feature extraction and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 27 (2) (2005) 230–244.

[34] C. Liu, H. Wechsler, Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition, IEEE Transanctions on Image Processing 11 (4) (2002) 467–476.

[35] C. Kotropoulos, A. Tefas, I. Pitas, Frontal face authentication using morphological elastic graph matching, IEEE Transactions on Image Processing 9 (4) (2000) 555–560.

[36] C. Liu, H. Wechsler, Robust coding schemes for indexing and retrieval from large face databases, IEEE Transanctions on Image Processing 9 (1) (2000) 132–137.

[37] I. Steinwart, Sparseness of support vector machines—some asymptotically sharp bounds, in: NIPS, 2003.

[38] Y. Meng, E. Chang, B. Li, Enhancing dpf for near-replica image recognition, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03) 02 (2003) 416.