Identification of Image Variations based on Equivalence Classes

Y. Maret, G.N. Garcia and T. Ebrahimi

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Institut de Traitement des Signaux
CH-1015 Lausanne, Switzerland

ABSTRACT

This paper presents a fingerprinting method based on equivalence classes. An equivalence class is composed of a reference image and all its variations (or replicas). For each reference image, a decision function is built. The latter determines if a given image belongs to its corresponding equivalence class. This function is built in three steps: synthesis, projection, and analysis. In the first step, the reference image is replicated using different image operators (like JPEG compression, average filtering, etc). During the projection step, the replicas are projected onto a distance space. In the final step, the distance space is analyzed, using machine learning algorithms, and the decision function is built. In this study, three machine learning approaches are compared: orthotope, support vectors machine (SVM), and support vectors data description (SVDD). The orthotope is a computationally efficient ad-hoc method. It consists in building a generalized rectangle in the distance space. The SVM and SVDD are two more general learning algorithms.

Keywords: Image Replica, Identification, Machine Learning, SVM, SVDD

1. INTRODUCTION

The problem of search and retrieval of multimedia content is an exciting field of research, which has attracted an increasing attention from both scientific and business communities. The activities in MPEG-7 standardization, and the more recent Still Image Search project within JPEG (JPSearch) are evidences of this growing interest.

In this paper, we describe a particular subset of search and retrieval problem which aims at the identification of all equivalent forms of a given multimedia content. By equivalent, we mean, all instances of a given content, which have been subject to a series of equivalence operators. For instance, an image of Albert Einstein (reference image), and all variants of that particular image, after application of a JPEG compression with different parameters, its zoomed versions, its filtered versions, etc. Such identification system can be of interest in applications in which one is interested in identifying all versions of a same content. Applications include search of content with illicit nature (child pornography and other illicite images), or variations of a content with copyright (images made by an artist).

In the following, we will call *replicas* all instances of a reference image. The *equivalence class* of a reference image is formed by all its replicas.

Current methods permitting to identify replicas are mainly based on two approaches: robust *watermarking* and robust *fingerprinting* (or perceptual hashing). In watermarking,[1] a signature is embedded in the reference image before broadcasting. A given image is equivalent to the reference image only if the same watermark is present. Watermarking techniques require to modify the reference image, which might be problematic in some cases (for example, when the reference image has already been broadcasted without embedding any watermarks). In fingerprinting, the reference image is analyzed to produce a signature correlated with its content. A given image is equivalent to the reference image only if their signatures are close enough. Fingerprinting techniques often rely on a single feature, for example typical points of the Radon transform,[2] log-mapping of the Radon transform,[3] or intra-scale variances of the wavelet coefficients.[4] Our method is a fingerprinting approach in the

Further author information: (Send correspondence to Y.Maret)
E-mail: Yannick.Maret@epfl.ch, Telephone: +41 21 693 4621, Fax: +41 21 693 7600

sense that there is no need to modify the reference image. However, contrary to most fingerprinting techniques, the presented method combines multiple features.

In,[5] feature distances such as structure, color and texture are linearly combined to form a unique distance quantifying the *similarity* of two images. The combination weights are empirically estimated; it was also shown that adapting them to the nature of the considered images increased the efficiency of the retrieval system. In this paper, distances are non-linearly combined to define an equivalence distance function, which is specific to each reference image.

In,[6] the orthotope approach non-linearly combines multiple features. The orthotope is a computationally efficient ad-hoc method. It consists in building a generalized rectangle in the distance space. This paper extends the approach by using Support Vector Machine (SVM) and Support Vector Data Description (SVDD) for the machine learning part of the method. The rest of this document is structured as follows. The proposed approach is detailed in Sec. 2. Practical considerations and experimental results are reported in Sec. 3.

2. PROPOSED APPROACH

The goal of a replica identification system is to determine if a given image is a replica of a reference image. To achieve this, a *decision function* is built for a particular reference image. The decision function can be used to determine if any given image is a replica of the reference image for which it is built.

In this work, we propose to train the decision function by using an *analysis by synthesis* approach.[7] More precisely, the decision function is constructed in three steps: *synthesis*, *projection*, and *analysis*. In the first step, the reference image is replicated using different image operators (like JPEG compression, average filtering, etc). During the projection step, the replicas are projected onto a distance space. In the final step, the distance space is analyzed, using machine learning algorithms, and a decision function is built. These three steps are presented in more details in the following subsections.

2.1. Synthesis Step

We first consider a *set of operators*. An element of this set is an operator returning a (un)modified version of the argument image. In the following, we assume that the identity operator is by default the first element of this set. The set of operators is thus non-empty. The operators to include in this set depend on the targeted application. For image copyright applications, the considered operators can be:[8] JPEG compression, geometric transformations, enhancement techniques, noise addition, or printing-scanning simulation.

Additionally, each operator can be parameterized. For example, a JPEG compression operator takes a quality factor as a parameter. Hence, a *set of parameterizations* is associated with an operator. Each of its element gives a possible parametrization of the associated operator. As for the operators, this set depends on the application requirement.

A replica of the reference image is obtained by applying an operator (with a given parameterization) to the reference image. The operator is chosen from the set of operators, and parameterized according to an element of the associated set of parameterizations. Thus, if N_o is the number of operators, and k_i is the number of parameterizations[*] corresponding to the operator i, then $\sum_{i=2}^{N_o} k_i$ replicas can be obtained.

It is possible to nest two or more operators to form a new operator known as a *composition*. In the following, we furthermore assume that an operator cannot be nested more than once in the same composition. Hence, the number of replicas that can be synthesized, using up to two nesting levels, amounts to $\sum_{i=1,j=2}^{N_o} k_i k_j (1 - \delta_{ij})$, where δ_{ij} is the Kronecker delta.

[*] For non-parametric operators, k is equal to 1. We also assume that the number of parameterizations is finite.

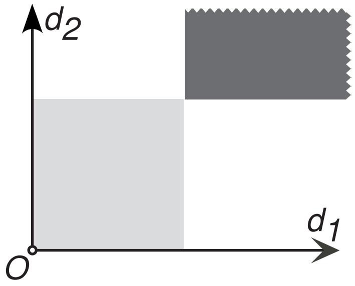


Figure 1. *An ideal case for subregions in* Ω. The light gray rectangle represents the subregion of replicas, and the dark gray rectangle that of non-replicas. d_1 (d_2) corresponds to the relative distances for feature 1 (2). The white subregions are 'undefined'.

2.2. Projection Step

In order to compare the similarity between two images, visual features are extracted. For each feature a metric quantifying the images similarity can be defined. Several visual features can be envisioned: color, texture, shape, etc. For an extensive survey on general features extraction, refer to.[9] The features choice depends on the image type, but also on the operators considered in the set of operators. For instance, if rotation is considered, it would make sense to choose features that are rotation invariant.

A metric is associated with each feature; it measures the similarity between two images. Hence, there are as many measures characterizing the similarity between two images as features. Hence, a positive vector can be associated with each image by grouping the similarity measures between the considered image and the reference image. We call this vector a *relative distance vector.*

Let Ω be the region of the real vector space corresponding to positive vectors. That is, the region in which the relative distance vectors lie. In this region, the replica identification problem amounts to *determine the subregion of* Ω *that corresponds to replicas of the reference image.* In an ideal case, Ω can be separated into two mutually exclusive subregions corresponding to replicas and non-replicas, respectively. Clearly, the origin is included in the former and corresponds to the reference image. It is also relatively intuitive that relative distance vectors corresponding to replicas lie near the origin. Figure 1 depicts an ideal distribution of replicas and non-replicas in a two-feature case.

Figure 2 shows some experimental distributions of replicas and non-replicas in Ω. They are obtained as explained in Sec. 3.1 by using the reference images shown in Fig. 3. It can be observed that several distributions resemble the ideal case depicted in Fig. 1. A notable exceptions is `church`, and to a lesser extent `car`, `Marie` and `statue`. Those are images containing many sharp edges, or non-homogeneous textures. Hence, they are not very well described by the chosen features.

We call a distribution in the Ω space a near-ideal one if it resembles the ideal distribution in Fig. 1. Otherwise it is called a *non-ideal distribution.*

2.3. Analysis Step

The analysis step amounts to determine which subregion of Ω corresponds to replicas of the reference image. It is a binary classification problem, where the two classes correspond to the replicas and non-replicas, respectively. Many classification algorithms can be used. The goal of those algorithms is to build, using a limited number of training examples, a classifier that generalizes well to novel patterns. In the following, three methods are compared: the orthotope approach,[6] the Support Vector Machine (SVM),[10, 11] and the Support Vector Data Description (SVDD).[12]

The orthotope approach is a method specifically designed to discriminate replicas for images with near-ideal distributions. It consists in building a generalized rectangle (or orthotope) in Ω such that it contains most of the relative distance vector corresponding to replicas but leaves outliers out. By construction, it performs well for images with near-ideal distributions. On the contrary, it does not do as well for images with non-ideal

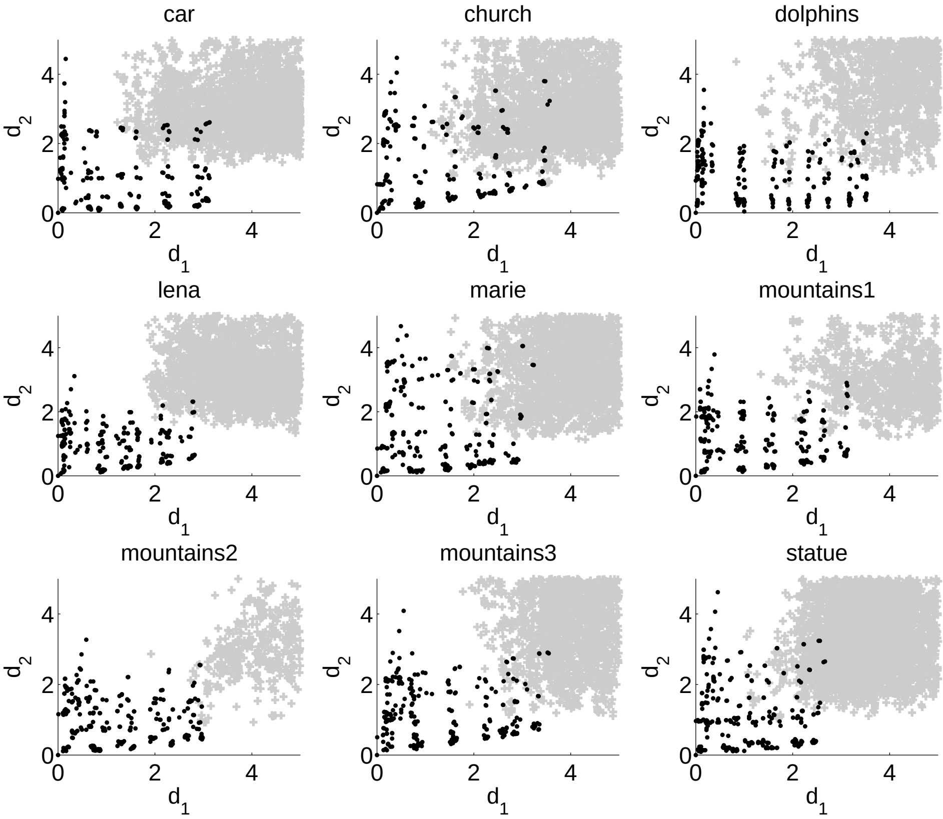


Figure 2. *Experimental examples.* Black dots represent replicas, and light-gray crosses stand for images that are not replicas.

distributions. Moreover, the orthotope approach is not readily generalized to use non-replica examples. However, it remains an interesting approach due to its computational simplicity.

SVM, and SVDD are more general approaches than the orthotope. They perform better than the latter on images with non-ideal distributions, but at the cost of an increased computational complexity.

In the following, subscripts in Roman letters index vectors (or scalars), and subscripts in Greek letters index elements of vector. Relative distance vectors are, in general, referred as patterns. Training patterns (or examples) are written $\mathbf{x_i}$, test patterns $\mathbf{z}$. The patterns are positive valued vectors of dimension D (i.e. D features are used). During the training phase, labels y_i are assigned to patterns. A pattern corresponding to a replica is simply called a *replica*, and is labeled '+1'. Otherwise it is called a *non-replica*, and is labeled '−1'. m_{+1} and m_{-1} are the number of training replicas and non-replicas examples, respectively. $m = m_{+1} + m_{-1}$ is the total number of training patterns.

2.3.1. Orthotope

The orthotope approach is based on the distribution of replicas only. A generalized rectangle, with the origin as one of its vertices, is built. The orthotope is fully characterized by its vertices having all but one coordinates equal to zero, i.e. $(w_1, 0, \ldots, 0)$ to $(0, \ldots, 0, w_D)$. Hence, the classification problem amounts to determine the orthotope limits w_1 to w_D. This can be achieved by the following constrained optimization problem:[6]

$$\min_{\mathbf{w},\boldsymbol{\xi}_i} \quad \sum_{\gamma=1}^{D} w_\gamma + C \cdot \sum_{i=1}^{m_{+1}} \sum_{\gamma=1}^{D} \xi_{i\gamma} \tag{1a}$$

$$\text{subject to} \quad x_{i\gamma} \leq w_\gamma + \xi_{i\gamma} \text{ and } \xi_{i\gamma}, w_\gamma \geq 0, \tag{1b}$$

where $\xi_{i\gamma}$ are positive variables which allow for some replicas to be outside the orthotope. However, the number of rejections should be limited. To achieve this, the sum of the slack variables is penalized by a positive tradeoff constant C. Note that C is the only parameter of this classification technique. The constrained optimization problem given in Eq. 1 can be solved be means of standard linear programming techniques.

The decision function associated to the orthotope is given by:

$$f(\mathbf{z}) = \operatorname{sgn}\left(\min\{1 - \frac{z_\gamma}{w_\gamma}\}_{\gamma=1,\ldots,D}\right), \tag{2}$$

where $\operatorname{sgn}(\cdot)$ is the signum function. That is, $f(\mathbf{z})$ is equal to '+1' if the tested pattern $\mathbf{z}$ is inside the orthotope, and '-1' otherwise.

To take into account non-replica examples, an additional constraint has to be added. Namely, the non-replica examples should not be inside the orthotope. This condition can be expressed mathematically by $\sum_{\gamma=1}^{D} \mathbf{1}_{\{x_{i\gamma} \leq w_\gamma + \xi_{i\gamma}\}} < D$, where $\mathbf{1}_{\{\cdot\}}$ is the indicator function[†]. Unfortunately, this additional constraint is neither linear, nor quadratic, and implies the use of more complex optimization techniques.

2.3.2. Support Vectors Machine

The basic Support Vectors Machine is a binary classifier that separates two classes with an hyperplane. This approach can be generalized by using a kernel. The kernel (non-linearly) maps the space of patterns into a new space where the two classes can be better discriminated by a hyperplane.

We use the ν-parametrization[10] of the Support Vector Machine, and a radial basis function as kernel. The dual constrained optimization problem is given in Eq. 3. In the dual form, only the Lagrangian multipliers α_i are optimized.

$$\max_{\boldsymbol{\alpha}} \quad L(\boldsymbol{\alpha}) = -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \tag{3a}$$

$$\text{subject to} \quad \sum_{i=1}^{m} \alpha_i y_i = 0, \sum_{i=1}^{m} \alpha_i = 1, \text{ and } 0 \leq \alpha_i \leq 1/(\nu m), \tag{3b}$$

with the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma^2)$. The parameters of this classification technique are $\nu \in [0, 1]$ and $\sigma \in \mathbb{R}^+$. The parameter ν can be shown to be an upper bound on the fraction of training errors, and a lower bound on that of support vectors.[10] The kernel parameter σ controls the complexity of the decision boundary. The constrained optimization problem given in Eq. 3 can be solved by means of standard quadratic programming techniques.

The decision function is given by:

$$f(\mathbf{z}) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{z}, \mathbf{x}_i) + b\right), \tag{4}$$

where the constant b is determined by the support vectors, that is $b = y_k - \sum_{i=1}^{m} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_k)$ for all $\mathbf{x}_k$ such that $0 < \alpha_k < 1/(\nu m)$. The name support vectors stems from the fact that many of the optimized α_i are equal to 0. Hence, only a relatively small fraction of the training patterns defines the decision function.

[†] $\mathbf{1}_{\{\cdot\}} = 1$ if the condition '$\cdot$' is true, otherwise $\mathbf{1}_{\{\cdot\}} = 0$

2.3.3. Support Vectors Data Description

The basic Support Vectors Data Description is a single-class classifier that describes patterns by fitting an hypersphere around them. As in the SVM case, this approach can be generalized by using a kernel. The kernel (non-linearly) maps the space of patterns into a new representation where the single-class can be better described by an hypersphere. Moreover, the SVDD can take into account training examples that are totally outside the description (in our case, the non-replicas).

In the case of a radial basis kernel, the dual constrained optimization problem can be expressed as in Eq. 5.[12] Like for the SVM, only the Lagrangian multipliers α_i are optimized in the dual form.

$$\max_{\boldsymbol{\alpha}} \qquad L(\boldsymbol{\alpha}) = -\sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j), \tag{5a}$$

$$\text{subject to} \quad \sum_{i=1}^{m} \alpha_i y_i = 1 \text{ and } 0 \leq \alpha_i \leq 1/(\nu_{y_i} m_{y_i}), \tag{5b}$$

with the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-||\mathbf{x}_i - \mathbf{x}_j||^2/\sigma^2)$. The parameters of this classification method are $\nu_{+1,-1} \in [0,1]$ and $\sigma \in \mathbb{R}^+$. The parameter ν_{+1} gives the fraction of replica examples that will be rejected, while ν_{-1} can be shown to be an upper bound on the fraction of accepted non-replica examples.[12] The kernel parameter σ controls the complexity of the decision boundary. The constrained optimization problem given in Eq. 5 can be solved by means of standard quadratic programming techniques.

The decision function is given by:

$$f(\mathbf{z}) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{z}, \mathbf{x}_i) + d\right), \tag{6}$$

where is constant d is determined by the support vectors, that is $d = -\sum_i^m y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_k)$ for all $\mathbf{x}_k$ such that $0 < \alpha_k < 1/(\nu_{y_k} m_{y_k})$. As for SVM, many of the optimized α_i are equal to 0. Hence, only a relatively small fraction of the training patterns defines the decision function.

3. EXPERIMENTAL RESULTS AND DISCUSSION

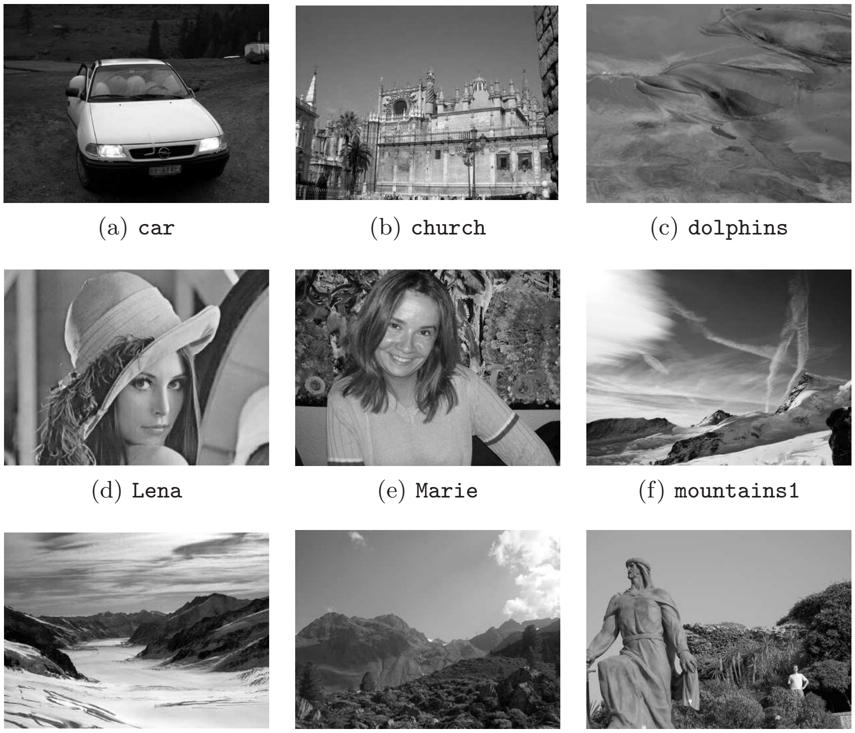
The proposed approach, using the three classification methods, is tested on the nine reference images depicted in Fig. 3. An images database is used to mimic the non-replicas. It contains 536 images, including photographs of people, landscapes, and buildings.

Section 3.1 details the generation of training patterns. Then, several experiments are carried out. The first simulation concerns the determination of the three classifiers parameters by means of cross-validation (Sec. 3.2). The second experiment bears on the global performances of the approach (Sec. 3.3). A third experiments determines the classifiers sensitiveness to training conditions (Sec. 3.4). Finally, the advantages, and drawbacks, of the three classifiers are compared (Sec. 3.5).

3.1. Patterns Generations

All classification methods require replica examples. They are generated as explained in Sec. 2.1. The operators, and their associated parameterizations, are shown in Tab. 1. That is, 257 replicas of the reference image are synthesized by using up to two nesting levels of compositions. The SVM and SVDD classification methods also require non-replica examples. They are provided by modifying images from the non-replica database. Hence, about 10000 non-replicas are generated by using a single nesting level of composition.

Replica (non-replica) patterns are generated by computing the relative distances, in Ω, between the replicas (non-replicas) and the reference image, as explained in Sec. 2.2. Two features are used to project the images onto Ω. The first feature is the *gray level histogram.* The gray level values are grouped into 16 bins. The corresponding distance metric is based on the histogram intersection algorithm.[13] The second feature is the *first order statistics of each subband of the Gabor transform.* This transform is performed as in.[14] More precisely,



(a) `car` (b) `church` (c) `dolphins`

(d) `Lena` (e) `Marie` (f) `mountains1`

(g) `mountains2` (h) `mountains3` (i) `statue`

Figure 3. *Reference images.*

Table 1. *Operators and their parameters.* The large amount of Gamma parameterizations is due to the SVDD sensitiveness to training conditions (Sec. 3.4).

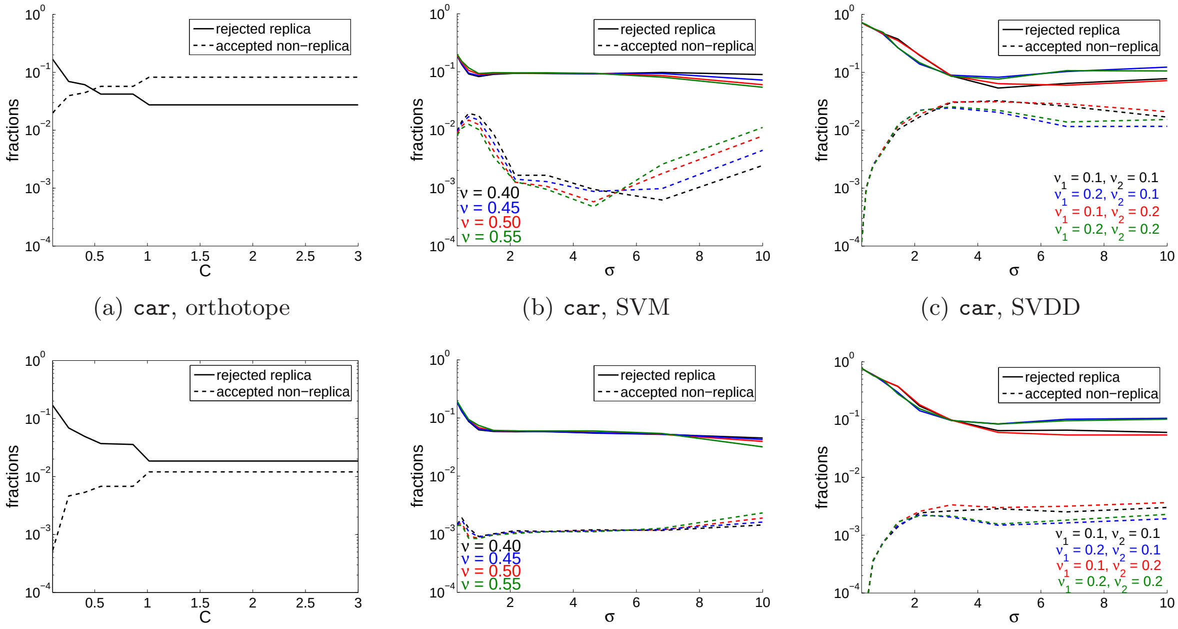
Operators	**Parameterizations**
JPEG compression	$Q = 50, 60, 70$
Gaussian noise addition	$\sigma = 2.5, 7.5$
Resizing	scale= $0.8, 1.2$
Averaging filter	order= $2, 3$
Gamma correction	γ =0.8, 0.85, 0.9, 0.95, 1.05, 1.1, 1.15, 1.2
Horizontal flipping	NA

the used parameters are 0.75 for the upper center frequency, 0.05 for the lower center frequency, five scales and six orientations. Hence, there is a total of 30 subbands to which correspond 30 mean values. The corresponding distance metric is the L_1-norm of the difference between two vectors of mean values.

Due to computational limitations, it is not possible to use that many non-replicas examples. In order to limit the number of non-replica examples, they need to be preselected. For SVM and SVDD, it is a well known result[11, 12] that training patterns that are not support vectors do not influence the resulting decision function. Hence, the number of non-replica examples is reduced by selecting the examples lying near the decision boundary. This is achieved by computing a symmetric Hausdorff distance between each non-replica pattern and the set of replica examples. Then only, 257 non-replicas with the smallest distances are included in the training set.

3.2. Determination of the Classification Parameters

For the orthotope, only the tradeoff parameter C has to be determined. For the SVM, the classification parameters to determine are ν and the kernel parameter σ. For the SVDD, they are ν_1, ν_2, and the kernel parameter σ. In the following, we use a normalized version of the radial basis function kernel: $k(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\frac{1}{\kappa}||\mathbf{x_i} - \mathbf{x_j}||^2/\sigma^2\right)$.



(a) `car`, orthotope (b) `car`, SVM (c) `car`, SVDD

(d) `mountains1`, orthope (e) `mountains1`, SVM (f) `mountains1`, SVDD

Figure 4. *Examples of cross-validation results.*

The normalization constant κ is set to the first decile of the distribution of the squared intra-replica distances. It ensures that the optimal value of σ is larger than one.

In order to find the optimum values of the classification parameters, a ten-fold cross-validation procedure is carried out.[15] In this procedure, the training examples are subdivided into ten mutually exclusive subsets, and ten runs are carried out. For each run, one set is put aside (validation group), and the decision boundary is computed using the union of the remaining sets (training group). The non-replicas examples not selected in Sec. 3.1 are added to the validation group.

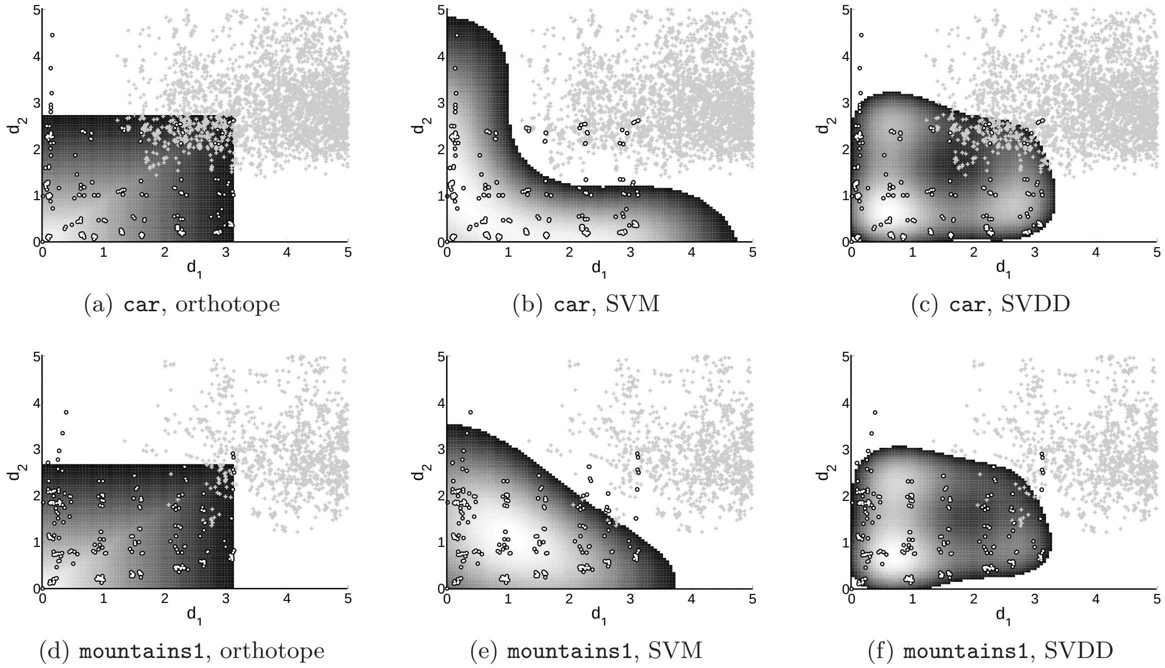
For each training group, the three classifiers are trained. The replica rejection and non-replica acceptance fractions (empirical errors) are computed by testing the trained decision functions on the associated validation group. The average empirical errors are reported in Fig. 4 for two reference images: `car` and `moutains1`. They are examples of a non-ideal, respectively near-ideal, distributions. The choice of the classification parameters depends on the targeted application. Indeed, the classification parameters influence not only the global performances, but also the ratio between replica rejection and non-replica acceptance fractions.

For this paper, we choose parameters minimizing the replica rejection fraction, while maintaining the lowest possible non-replica acceptance fraction. In the following, experiments are carried out with parameters selected by the above criterium. A value of $C = 0.5$ is found to give a good compromise between the number of accepted non-replicas and the number of rejected replicas, for a large class of images. For the SVM, $\nu = 0.5$, and $\sigma = 5$ gives good results. For the SVDD, $\nu_{+1} = 0.1$, $\nu_{-1} = 0.2$, and $\sigma = 5$ are selected. Figure 5 depicts the resulting decision boundaries, for the three classifiers, on the reference images `car` and `mountains1`.

Figures 5(a) and 5(d) clearly illustrate that the orthotope method does not take into account the distribution of non-replica examples.

Figure 5(b) shows that the SVM decision boundary is almost an hyperplane for a near-ideal distribution. Figure 5(e) depicts a 'boot' shape for a non-ideal distribution. That is, the decision boundary avoids most of the non-replica examples.

The SVDD decision boundaries are qualitatively similar for both the near-ideal distribution, Fig. 5(c), and non-ideal distribution, Fig. 5(f). That is, the distribution of non-replica examples does not influence the boundary



(a) `car`, orthotope (b) `car`, SVM (c) `car`, SVDD

(d) `mountains1`, orthotope (e) `mountains1`, SVM (f) `mountains1`, SVDD

Figure 5. *Examples of decision boundary.* The black circles depicts replicas, and the gray crosses non-replicas. The background color shows the decision surface corresponding to replicas.

as strongly as for the SVM. Moreover, the SVDD is not well suited for the application. Indeed, in both cases the reference images are consider as non-replicas. It can be explained by the typical tightness of the SVDD boundary around a class it describes (here the replicas), and by the relative sparseness of replica examples near the origin. Indeed, in Sec. 3.4 it is shown that the SVDD is very sensitive to the training conditions.

3.3. Global Performances

The global performances, averaged on the nine reference images, can be found in Tab. 2. Moreover, it shows the average performances for the near-ideal distributions (`dolphins, Lena, mountains1, mountains2`, and `mountains3`), and non-ideal distributions (`car, church, Marie`, and `statue`).

It can be seen that the orthotope approach performs better than the SVM and the SVDD concerning the fraction of rejected replicas, both for near-ideal and non-ideal distributions. Moreover, this fraction is equal for both distributions. This result is not surprising considering that the orthotope determination is based on replica examples only. Concerning the fraction of accepted non-replicas, the orthotope outperforms the SVM and the SVDD for near-ideal distributions. For non-ideal distributions, the orthotope approach is largely outperformed by the SVM, but only slightly by the SVDD.

The SVM and the SVDD perform equally well concerning the fraction of rejected replicas for near-ideal distributions. In that case, the fraction of accepted non-replica is less for the SVM than for the SVDD. For non-ideal distributions, the SVDD has a better fraction of rejected replica, but the fraction of accepted non-replica is almost the double. It can be explained by the choice of the SVDD classification parameter $\nu_{+1} = 0.1$. This choice implies a upper bound on the fraction of rejected replicas. On the other hand, for the SVM, the error bound ν applies on both rejected replica and accepted non-replica fractions.

3.4. Sensitivity to the Training Conditions

We tested the three classification methods sensitivity to the training conditions. These training conditions include the number of training examples, and the ratio $r_{+/-}$ between replica and non-replica examples (for SVM and SVDD). They are tested by varying the number of replica examples, and that of non-replica examples. A

Table 2. *Global performances of the three classification methods.*

		fraction of rejected replicas	fraction of accepted non-replicas
Orthotope	all distributions	0.045	0.024
	near-ideal distributions	0.046	0.0021
	non-ideal cases	0.045	0.053
SVM	all distributions	0.076	0.014
	near-ideal distributions	0.05	0.003
	non-ideal cases	0.11	0.028
SVDD	all distributions	0.054	0.024
	near-ideal distributions	0.054	0.0086
	non-ideal cases	0.054	0.045

Table 3. *Summary of classification methods comparison.*

	Orthotope	SVM	SVDD
Performance	+	++	++
Simplicity	+++	+	+
Flexibility	+	++	+++
Suitability	++	++	+

ten-fold cross-validation procedure is carried out for each couple of training sizes. In this procedure, hierarchical training sets are used. They are necessary to obtain consistent results. That is, a training group contains several nested training sets (one for each training size). For example, if the group one contains replica training sets of sizes 100 and 150, the latter includes elements of the former.

Figures 6(a) and 6(d) show the orthotope method sensitivity to the training conditions. It can be seen that the fraction of rejected replica is sensitive to the number of training examples. On the other hand, the fraction of accepted non-replica is relatively insensitive to the number of training example. This can be explained by the larger amount of non-replicas in the validation set.

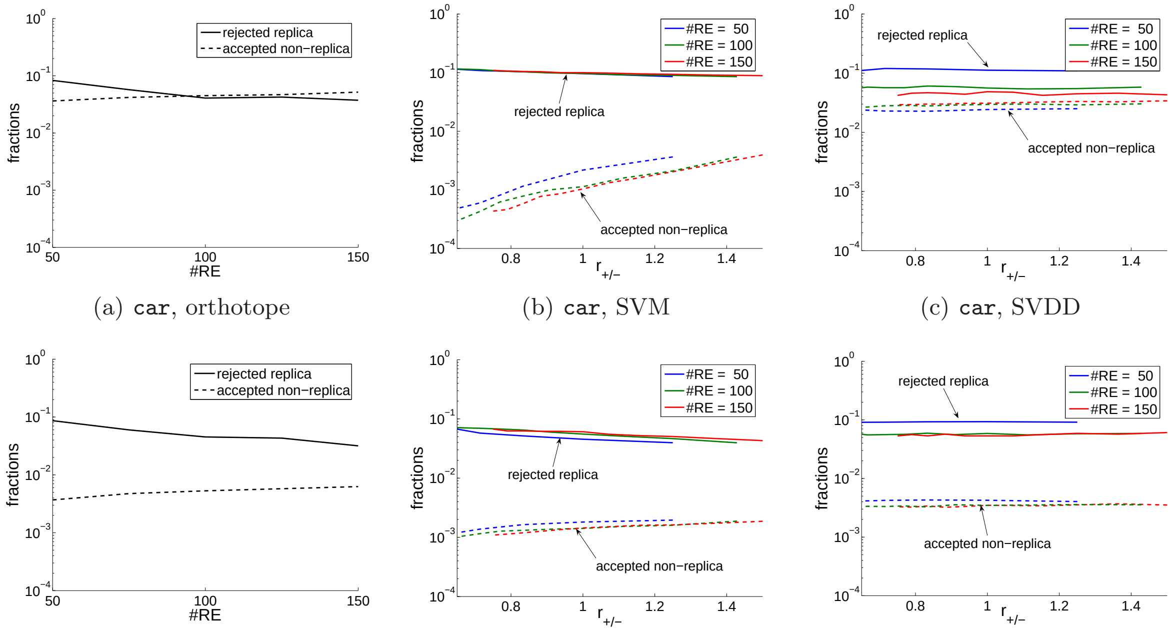
Figures 6(b) and 6(e) show the SVM sensitivity to the training conditions. The fraction of rejected replica is insensitive to the number of training replicas, and relatively insensitive to the ratio $r_{+/-}$. On the other hand, the fraction of accepted non-replica is sensitive to the number of training replicas and to the ratio $r_{+/-}$. This effect can be explained by the SVM tendency to "fill the gap" when less examples are available, as it can be seen in Fig. 7(b).

Figures 6(c) and 6(f) show the SVDD sensitivity to the training conditions. The empirical errors are insensitive to ratio $r_{+/-}$. On the other hand the fraction of rejected replicas are very sensitive to the number of training replicas. This can be explained by the typical tightness of the SVDD decision boundary around replicas examples. Indeed, as it can be seen in Fig. 7(c), the number of gamma parameterizations influence greatly the decision boundary.

3.5. Comparison between the Orthotope, SVM and SVDD Classifiers

The three classification methods have advantages and drawback that are summarized in Tab. 3. The advantages of the orthotope approach are its low complexity (linear programming) and its good performance for near-ideal distribution. Moreover, it is well suited to the replica identification as it takes into account the particularities of Ω. On the other hand, the orthotope method cannot be readily extended to make use of non-replica examples. Also its performances on non-ideal distributions are relatively poor, and cannot be tuned to the application requirements.

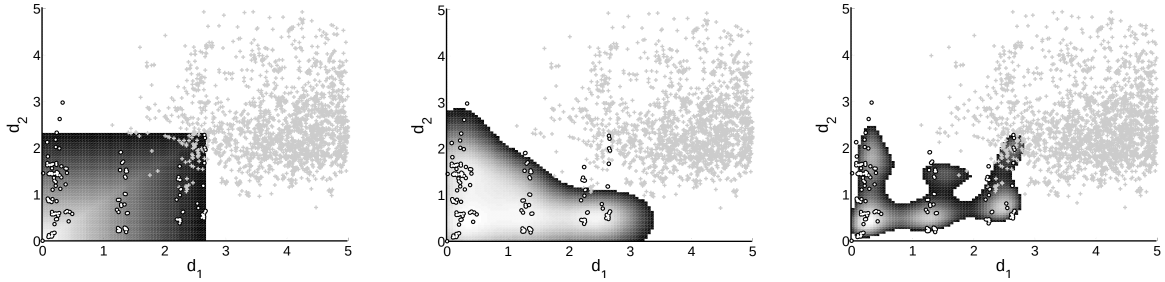
The advantages of the SVM are its good performances, and its suitability to the application. Moreover, it makes naturally use of non-replica examples. On the other hand, it is more complex than the orthotope (quadratic programming), and more parameters need to be tuned. Also, error bounds cannot be specifically set to the fraction of rejected replicas, or accepted non-replicas.



(a) car, orthotope (b) car, SVM (c) car, SVDD

(d) mountains1, orthotope (e) mountains1, SVM (f) mountains1, SVDD

Figure 6. *Training conditions.* *#RE* is the number of replica examples used for training, and $r_{+/-}$ gives the ratio between the number of replica and non-replica examples.



(a) mountains1, orthotope (b) mountains1, SVM (c) mountains1, SVDD

Figure 7. *Influence of the parameterizations of the gamma operator.* In this case only $\gamma = 0.8, 0.9, 1.1, 1.2$ are considered.

The advantages of the SVDD are its good performances, and the flexibility given by the two parameter ν_{+1} and ν_{-1}. Indeed, error bounds can be specifically set for the fractions of rejected replica, or accepted non-replica. On the other hand, it is more complex than the orthotope. Also, it is less suited to the application since the reference image is sometime not considered to be a replica. Moreover, it requires a carefully parameterizations of some of its operators. That is, it is very sensitive to the distribution of replica examples.

4. CONCLUSIONS AND FUTURE WORK

This paper reports an original approach for image identification based on equivalence classes. Equivalence of a reference image is defined as all admissible variations of that image when subjected to a set of operators. The approach is based on the determination of the equivalence class surface in the space of features distance vectors. The surface is determined by machine learning algorithms: orthotope (a previously reported ad-hoc method), SVM, and SVDD. All these algorithms take into account the distribution of variation examples in the considered space. Additionally, the SVM and SVDD take also into account the distribution of images that are not variations. It is experimentally found that for reference images with near-ideal distributions (about fifty percent of the images) the orthotope approach performs better than the SVM or SVDD. However, the SVM outperforms both the orthotope and the SVDD for the remaining images. Moreover, it is remarked that the latter need a carefully optimized synthesis step, since it is very sensitive to the training conditions.

As a future work, a study on the sensitivity of the method with respect to choices of features and distance metrics might provide several useful insights into the relative importance of each feature. This will pave the way to a scheme based on automatic feature selections.

ACKNOWLEDGMENTS

The first author is partly supported by the Swiss National Science Foundation – "Multimedia Security", grant number 200021-1018411. The work was partly developed within VISNET, a European Network of Excellence (`http://www.visnet-noe.org`), funded under the European Commission IST FP6 programme. The authors would like to acknowledge Frédéric Dufaux for fruitful discussions and comments. Also, the authors would like to thanks the creators of libsvm,[16] and dd_tools.[17]

REFERENCES

1. F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE* **87**, pp. 1079 – 1107, July 1999.
2. F. Lefèbvre, B. Macq, and J.-D. Legat, "Rash: Radon soft hash algorithm," in *EURASIP European Signal Processing Conference*, (France), September 2002.
3. J. Seo, J. Haitsma, T. Kalker, and C. Yoo, "Affine transform resilient image fingerprinting," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, (Hong Kong), April 2003.
4. R. Venkatesan, S.-M. Koon, M.-H. Jakubowski, and P. Moulin, "Robust image hashing," in *IEEE International Conference on Image Processing*, (Vancouver), September 2000.
5. Q. Iqbal and J. Aggarwal, "Combining structure, color and texture for image retrieval: A performance evaluation," in *IEEE Conference on Pattern Recognition*, **2**, pp. 438–443, August 2002.
6. Y. Maret, G. Garcia Molina, and T. Ebrahimi, "Images Identification Based on Equivalence Classes," in *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005)*, April 2005.
7. R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley, 2000.
8. F. A. P. Petitcolas and M. Kutter, "Fair Evaluation Methods for Image Watermarking Systems," *Journal of Electronic Imaging* **9**, pp. 445–455, October 2001.
9. Y. Rui, T. Huang, and S. Chang, "Image retrieval: current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation* **10**, pp. 39–62, April 1999.
10. B. Schölkopf, A. Smola, R. Williamson, and P. Bartlett, "New support vector algorithms," *Neural Networks* **22**, pp. 1083–1121, 2000.
11. C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **2**(2), pp. 121–167, 1998.
12. D. M. Tax and R. P. Duin, "Support Vector Data Description," *Machine Learning* **55**, pp. 45–66, 2004.
13. M. Swain and D. Ballard, "Indexing via color histograms," in *Computer Vision*, pp. 390–393, December 1990.
14. B. Manjunath and W. Ma, "Texture Features for Browsing and Retrieval of Image Data," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**, pp. 837–842, August 1996.
15. C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
16. C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
17. D. Tax, "DDtools, the Data Description Toolbox for Matlab," March 2005. version 1.1.2.