

Image Replica Detection based on Support Vector Classifier

Y. Maret, F. Dufaux and T. Ebrahimi

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

ABSTRACT

In this paper, we propose a technique for image replica detection. By replica, we mean equivalent versions of a given reference image, e.g. after it has undergone operations such as compression, filtering or resizing. Applications of this technique include discovery of copyright infringement or detection of illicit content.

The technique is based on the extraction of multiple features from an image, namely texture, color, and spatial distribution of colors. Similar features are then grouped into groups and the similarity between two images is given by several partial distances. The decision function to decide whether a test image is a replica of a given reference image is finally derived using Support Vector Classifier (SVC).

In this paper, we show that this technique achieves good results on a large database of images. For instance, for a false negative rate of 5 % the system yields a false positive rate of only $6 \cdot 10^{-5}$.

Keywords: replica detection, features extraction, machine learning, support vector classifier

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 the following, we will call *replicas* all instances of a reference image. On the contrary, *non-replicas* are images that are not a modified version of the reference image. For example, consider an image of Albert Einstein. Its replicas are all variants of that particular image, after application of a JPEG compression with different parameters, its zoomed versions, its filtered versions, etc.

In this paper, we describe a particular subset of search and retrieval problem which aims at the detection of all replicas of a *reference image*. Such detection system can be of interest to detect 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 from a photograph).

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. This technique permits to detect replicas with relatively high probability, and to falsely detect non-replica as replica with a very low probability. However, 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 Those techniques are usually robust and computationally efficient and can be used to index database for fast retrieval. However, they detect non-replicas as replicas with a relatively high probability. It means that perceptual hashing is not yet mature enough to be used in a replica detection system.

Send correspondence to yannick.maret@a3.epfl.ch

Recently, researches have been conducted on the replica detection problem. For example, the method of Ke *et al*⁵ is based on the extraction of features called Key Points (KPs) which are stable points in the scale-space representation of the image. An image is described by thousands of KPs. Local Sensitive Hashing (LSH) is used to match KPs between a test image and the reference images. The test image is a replica if its KPs match with those of a reference image. This method achieves very good performance, but at the price of a computationally complex feature extraction step. Another method is that of Qamra *et al*.⁶ It is based on the computation of a perceptual distance function (DPF). More precisely, a distance function, measuring the similarity between two images, is built for each reference image. The main idea of DPF is to activate different features for different image pairs. That is, it ranks the features from the most similar to the most dissimilar and uses the former to compute the distance. This method achieves good performance but less than that of Ke *et al*.

Our approach is similar to that of DPF. Indeed, our replica detection system is based on the construction of a *replica detector* for a particular *reference image*. The detector is used to determine if a given *test image* is a replica of the reference image. If several reference images are present, as many detectors are needed, and a test image need to be tested against all of them. This can be relatively costly when the number of reference images grows very large. However, this problem can be circumvented by using LSH and perceptual hashing to reduce the number of candidate reference images before applying the proposed technique.

The main idea of our approach is to set a limit on how much the features of replica images can vary with respect to those of the reference image. For examples, consider that images are described by three features A , B and C . A given reference image has some fixed values for those features. A particular replica may correspond to only slightly modified values for features A and B but to a quite different value for feature C . On the other hand, another replica could correspond to nearly similar feature B and C but to a very different feature A . A third replica could even corresponds to mildly different values for all features. This behavior can easily be observed when a feature is invariant to certain operations, and another feature is invariant to other operations. Partial distances permit to control how much a feature, or a group of features, of replica images is allowed to vary with respect to that of the reference image.

The proposed approach differs from that of DPF in several aspects. For one, the similarity between two images is described by several partial distances in our approach, while it is described by a single distance in DPF. Moreover, we activate all features to compute those distances, while only the most similar ones are activated in DPF. The method of KPs is quite dissimilar from both DPF and the proposed method. Indeed, in KPs no distances are directly computed and it is the number of matching key points between two images that quantifies their similarity.

The rest of the paper is organized in two main parts. In the first part, the method is introduced and explained in details. In the second part, the method is tested on a large database and compared with the methods of Ke *et al* and that of Qamra *et al*.

2. METHOD

The detection system consists in three steps that are shown in Fig. 1. In the first step features are extracted from the test image. In the second step, the features are used to compute partial distances between the test and reference images. In the last step, the partial distances are used to determine if the test image is a replica of the reference image. It makes use of a decision function which is built using machine learning algorithms. These steps are described in more details thereafter.

2.1. Feature Choice and Extraction

In order to compare the similarity between two images, visual features are extracted. The goal of feature extraction is twofold. First, it maps images onto a common space where they can be more easily compared. Second, it reduces the space dimensionality by keeping only the relevant information.

Several visual features can be envisioned: color, texture, shape, etc. For an extensive survey on general features extraction, refer to.⁷ The features choice obviously depends on the image type. In the case of the image replica detection problem, it also depends on the type of replicas that are to be detected. For instance, if rotated images are considered, it would make sense to choose features that are rotation invariant.

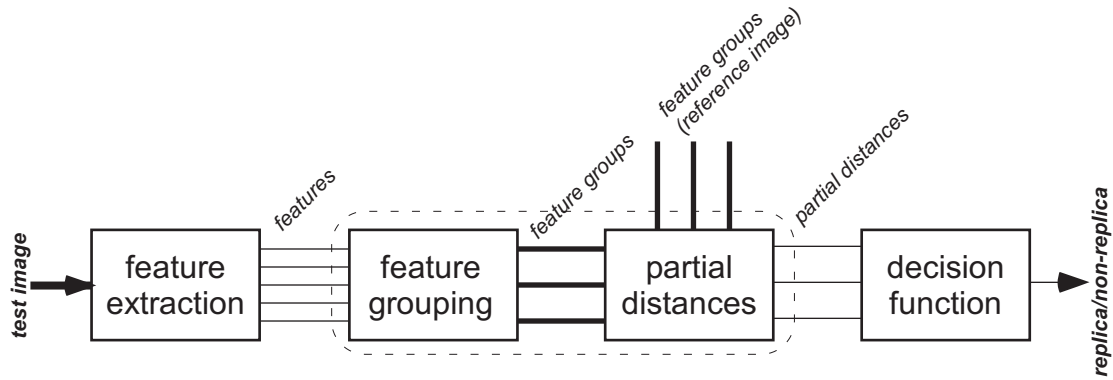


Figure 1. Block diagram of the replica detection system.

The features used in this work are of three types: texture, color, and spatial distribution of colors. They are similar to those used in,⁶ and are found to give good results in image retrieval applications. In all, we extract a total of 138 features as explained in the following subsections.

Before extracting features, an image is first cropped such that only 90 % of the center region is kept. It introduces a weak robustness to operation such as framing. Then, it is resized such that it contains approximately 2^{16} pixels (corresponding to a square image of 256×256 pixels), while keeping its original aspect ratio. It introduces a weak form of scale invariance and permits to speed up the feature extraction time by reducing the image size.

2.1.1. Texture

The texture feature is composed of the first and second order statistics of each subband of the Gabor transform. The latter is performed as in.⁸ More precisely, the used parameters are 0.75 for the upper center frequency, 0.05 for the lower center frequency, five scales and six orientations. Mean and variance estimates of the squared coefficients are computed for each of the 30 subbands. It results in a total of 30 mean and 30 variance estimates.

2.1.2. Color

The color feature is based on the Hue Saturation Value (HSV) color space. Each pixel in the image is classified into one of ten *color classes* depending on its position in the HSV space. The classes are the achromatic colors *black*, *gray* and *white*, and the chromatic colors *red*, *orange*, *yellow*, *green*, *cyan*, *blue* and *purple*. Mean and variance estimates of the Value channel are computed for each class. Mean and variance estimates of Saturation and Hue channels are also computed for each chromatic colors. It results in a total of 24 mean and 24 variance estimates. Moreover, a color classes histogram is also computed, giving the proportion of each color in the image. It results in an additional 10 values.

This is similar to the ‘culture’ color approach proposed in.⁶ In this study, they also consider pink and brown, whereas in our case these two colors are classified as red or orange. Brown and pink have the same Hue as red or orange, but differ in the Value and/or Saturation channels. Operations such as saturation or intensity changes are common in image processing, and modify the Value and the Saturation channels but not the Hue channel. If brown and pink are considered, red or orange pixels could be transformed into brown or pink pixels, or vice versa. For this reason, we have decided to include brown and pink within the red and orange classes.

2.1.3. Spatial Distribution of Colors

In addition of color statistics, the shape of spatial distribution of colors are computed. This is achieved by computing two shapes characteristics for each color class: spreadness and elongation.^{9–11} The first characteristic measures the compactness of the spatial distribution of a color. The second one reflects how much the spatial distribution has more length than width. These two features are scale and orientation invariant. We consider an elongation and a spreadness measures per color class. It results in 10 spreadness and 10 elongation measures.

2.2. Partial Distances

In many works, images are described by a single high-dimensional vector containing all the features.⁶ The similarity between two images is then given by computing a distance between the two corresponding feature vectors in this high-dimensional space. Several distance can be used, for example distance based on the L^1 or L^2 metrics, or on perceptual and psychological considerations. Given the distances between a reference image and several test images, the replica detection problem amounts to determine an optimal threshold under which the corresponding test images are likely to be a replica of the reference image. Good results have been obtained by constructing a distance function adapted to the replica detection problem and to a given reference image.⁶ All the complexity of the task is thus concentrated into the construction of the distance function.

In this work, we propose an alternative approach based on *partial distances*.

2.2.1. Partial Distances Choices and Computation

In this study, we propose to first group similar features. As a result, an image is described by several feature vectors, that we call *feature groups*, instead of a single one. The similarity between two images is then given by several *partial distances*.

Features that share similar characteristics are grouped together. It permits to group features that are *a priori* invariant to the same operations. More specifically, images are described by the following eleven features groups. There are two groups for texture: mean and variance estimates of the energy. There are seven groups related to color: mean estimates of the Hue, Saturation and Value channels; variance estimates of the Hue, Saturation and Value channels; and color classes histogram. Finally, there are two groups for color spatial distribution: spreadness and elongation measures.

For each feature group g , the corresponding distance d between two images \mathbf{I}_1 and \mathbf{I}_2 is given by:

$$d_g^2(\mathbf{I}_1, \mathbf{I}_2) = \sum_i w_{g,i} \left(f_{g,i}^{(1)} - f_{g,i}^{(2)} \right)^2, \quad (1)$$

where the $f_{g,i}^{(1,2)}$ is the i -th feature corresponding to group g for image one, respectively two, and the $w_{g,i}$ are positive weights such that $\sum_i w_{g,i} = 1$. For the two texture groups, and for the color classes histogram the $w_{g,i}$ are 1/32, respectively 1/10. That is, the same weights are given to each feature in the group. For the remaining groups, we use $w_i = (p_i^{(1)} + p_i^{(2)})/2$, where the $p_i^{(1,2)}$ give the proportions of each of the ten colors for image one, respectively two, and is straightforwardly derived from the color histogram. This distance gives more weights to the colors mostly present in both images.

2.2.2. Replica Detection Problem with Partial Distances

Let P be the region of the real vector space corresponding to positive vectors. That is, the region in which the partial distances lie. Given the partial distances between a reference image and several test images, the replica detection problem amounts to determine the optimal subregion of P in which the corresponding test images are likely to be replicas of the reference image.

In an ideal case, P can be separated into two mutually exclusive subregions corresponding to replicas and non-replicas of the reference image, respectively. Clearly, the origin is included in the former and corresponds to the reference image. It is also relatively intuitive that partial distances corresponding to replicas lie near the origin, and that those corresponding to non-replicas lie away from the origin. The ideal case is depicted in Fig. 2, using two feature groups.

2.3. Replica decision function Generation

The decision function needs to determine which subregion of P 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.

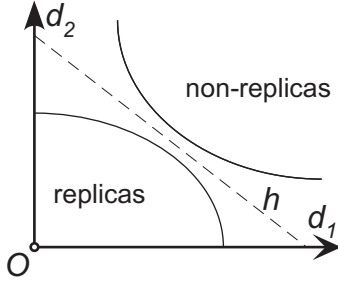


Figure 2. An ideal case for subregions in P . Two feature groups are used, corresponding to two partial distances d_1 and d_2 . The region corresponding to replicas can be separated from that corresponding to non-replicas by a hyperplane h .

In the partial distances approach, the feature space is mapped into a new space where the generic positions of replicas and non-replicas patterns are known. It enables the use of machine learning algorithms with more control, as the geometry of the problem is known. Indeed, intuitively a hyperplane seems to well separate the two classes. We showed¹² that Support Vector Classifier (SVC) yielded good performances for this problem. In fact, the basic SVC^{13,14} is nothing else than a binary classifier that separates two classes with an hyperplane.

Note that by using kernels in a SVC classifier, patterns are mapped into a space where they can be better discriminated by a hyperplane. In our case, the large number of features prevents to directly use the SVC on the raw features. It would require a prohibitively large number of training patterns, and the optimization process would probably yield an overtrained decision function due to the ‘curse of dimensionality’.¹⁵

2.3.1. Support Vector Classifier

In the following, subscripts in Roman letters index vectors (or scalars). Partial distances are, in general, referred to as patterns. Training patterns (or examples) are denoted as \mathbf{x}_i , with $i = 1, \dots, m$ where m is the total number of training patterns. The patterns are non-negative vectors of dimension D (i.e. D partial distances are used). During the training phase, a label y_i is assigned to each pattern \mathbf{x}_i . A pattern corresponding to a replica is simply called a *replica* and labeled $y_i = +1$. Otherwise it is called a *non-replica* and labeled $y_i = -1$.

We use the ν -parametrization¹³ of the SVC, and a radial basis function as kernel. The dual constrained optimization problem is given in Eq. 2. In the dual form, the Lagrangian L is maximized by optimizing the Lagrangian multipliers α_i :

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

$$\text{subject to} \quad \sum_{i=1}^m \alpha_i y_i = 0, \sum_{i=1}^m \alpha_i = 1, \text{ and } 0 \leq \alpha_i \leq 1/(\nu m). \quad (2b)$$

We use a radial basis function kernel $k(\cdot, \cdot)$ given by:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right). \quad (3)$$

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.¹³ The kernel parameter σ controls the complexity of the decision boundary. The constrained optimization problem given in Eq. 2 can be solved by means of standard quadratic programming techniques.

The decision function indicates to which class the (novel) test pattern \mathbf{z} belongs. The function is given by:

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

Table 1. *Training and patterns generation.* Image operations and their parameters.

Operations	Parameters
JPEG compression	$Q = 30, 50, 70, 90$
Gaussian noise addition	$\sigma = 3/255, 5/255, 910/255$
Resizing	$s = 0.5, 0.7, 1.3, 1.5$
Averaging filter	order= 3, 4
Gamma correction	$\gamma = 0.5, 0.8, 0.9, 1.1, 1.2, 1.5$
Horizontal flipping	NA
gray level conversion	NA
cropping	keep 90 %, 70 % and 50 % of the image
V channel change	-10 % and +10 %
S channel change	-10 % and +10 %

where the constant b is determined by the support vectors. More precisely, $b = y_k - \sum_{i=1}^m y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_k)$ 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.

2.3.2. Training Pattern Generation

Examples of both replica and non-replica pattern are needed to train the SVC classifier.

Example of replica images can be generated artificially. The reference image is first modified using different operators, resulting in different replicas. For image copyright applications, the considered operators can be:¹⁶ JPEG compression, geometric transformations, enhancement techniques, noise addition, or printing-scanning simulation. Namely, they are generated by the operations listed in Tab. 1.

Examples of non-replica images can be obtained by using an image database.

The features are extracted as explained in Sec. 2.1. Then they are normalized using a statistical normalization.¹⁷ More precisely, let μ_k and σ_k be the mean and standard deviation of values of the k -th feature over the training set. Then the normalized feature $w[k]$ is given by:

$$w[k] = \frac{v[k] - \mu_k}{3 \cdot \sigma_k}, \quad (5)$$

where $v[k]$ is the original feature. This normalization ensures that 99 % of the $w[k]$ are in the interval $[-1, 1]$ in the case of Gaussian distribution.

Finally, the training replica (non-replica) patterns are obtained by computing the partial distances between the replica (non-replica) images and the reference image, as described in Sec. 2.2.

2.3.3. Determination of the Classification Parameters

In the ν -SVC, the kernel parameter σ and the parameter ν are to be determined. They need to be set such that the *generalization error* is minimized. The generalization error is the error obtained when testing novel pattern with a trained decision function.

More precisely, we want to minimize the quantity given by $e = w_1 \cdot e_1 + w_2 \cdot e_2$ where e_1 is the generalization error for false positive (novel non-replicas classified as replicas), e_2 is the generalization error for false negative (novel replicas classified as non-replicas), and $w_{1,2}$ are two positive constants summing to one. The choice of $w_{1,2}$ depends on the application. In this study, these weights were set to take into account the skewness between the probability that the test image is a true non-replica and that it is a true replica. There are many more non-replicas than replicas, so that $w_1 = 0.999$ and $w_2 = 0.001$ is an appropriate choice.

Cross-validation is a popular technique for estimating generalization error. In k -fold cross-validation, the training patterns are randomly split into k mutually exclusive subsets (the folds) of approximately equal size. The SVC decision function is obtained by training on $k - 1$ of the subsets and then tested on the remaining subset. This procedure is repeated k times, with each subset used for testing once. Averaging the test error

over the k trials gives an estimate of the expected generalization error. This method was shown to yield a good estimation of the generalization error.¹⁸

In the following, we use a normalized version of the radial basis function kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\kappa \cdot \sigma^2}\right). \quad (6)$$

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

In this work, parameters for σ and ν are estimated in two steps as follows. In the first step, the parameter ν is set to some prefixed value while σ varies across a large range of values. For each σ , a 15-fold cross-validation is carried out. The σ for which the estimated e is the lowest is then chosen. In the second step, σ is set accordingly while ν varies in the range $[0, 1]$. For each ν , a 15-fold cross-validation is carried out. The ν for which the estimated e is the lowest is then chosen. This is a suboptimal algorithm but it avoids to run a large number of cross-validations and still achieve interesting results.

3. RESULTS

To simulate the performance of the proposed approach, we used the same image database as in:⁵ MM270k. This database can be downloaded from <http://www-2.cs.cmu.edu/~yke/retrieval/>. It contains 18,785 photographs including (but not limited to) landscapes, animals, constructions, and people. The image sizes and aspect ratios are variables, for example 900×600 , 678×435 , or 640×480 . They are mostly color images, except for a few hundreds that are gray levels.

For training, we randomly selected 750 images in the database. Among the selected pictures, 250 are randomly chosen to be reference images, and the remaining are used as non-replica examples during the training phase. For each reference image, a replica detector is built as described in Sec. 2.

The replica detectors are then tested on the remaining images in the database. This permits to estimate the false positive rate for each reference images. The false negative rate is estimated by testing the replica detectors on test replica examples. They are generated by the transforms listed below. These operations are the same than the ones used in.^{5,6} They are implemented using the open source library `ImageMagick`^{*}.

- *Colorizing*. Tint the Red, Green, or Blue channel by 10 %;
- *Changing contrast*. Increase or decrease the contrast using ImageMagick's default parameter;
- *Cropping*. Crop by 5 %, 10 %, 20 %, or 30 %;
- *Despeckling*. Apply ImageMagick's despeckling operation;
- *Downsampling*. Downsample by 10 %, 20 %, 30 %, 40 %, 50 %, 70 %, or 90 % (without antialiasing filtering);
- *Flipping*. Flip along the horizontal axis.
- Reduce the number of colors. Reduce the color palette to 256 colors;
- *Framing*. Add an outer frame of 10 % the image size. Four images are produced with different frame color.
- *Rotating*. Rotate by 90°, 180° or 270°.
- *Scaling*. Scale up by 2, 4, 8 times, and down by 2, 4, 8 times (use antialiasing filter).
- *Changing Saturation*. Change the values of the saturation channel by 70 %, 80 %, 90 %, 110 %, or 120 %.
- *Changing Intensity*. Change the intensity channel by 80 %, 90 %, 110 %, or 120 %.

^{*}<http://www.imagemagick.org>

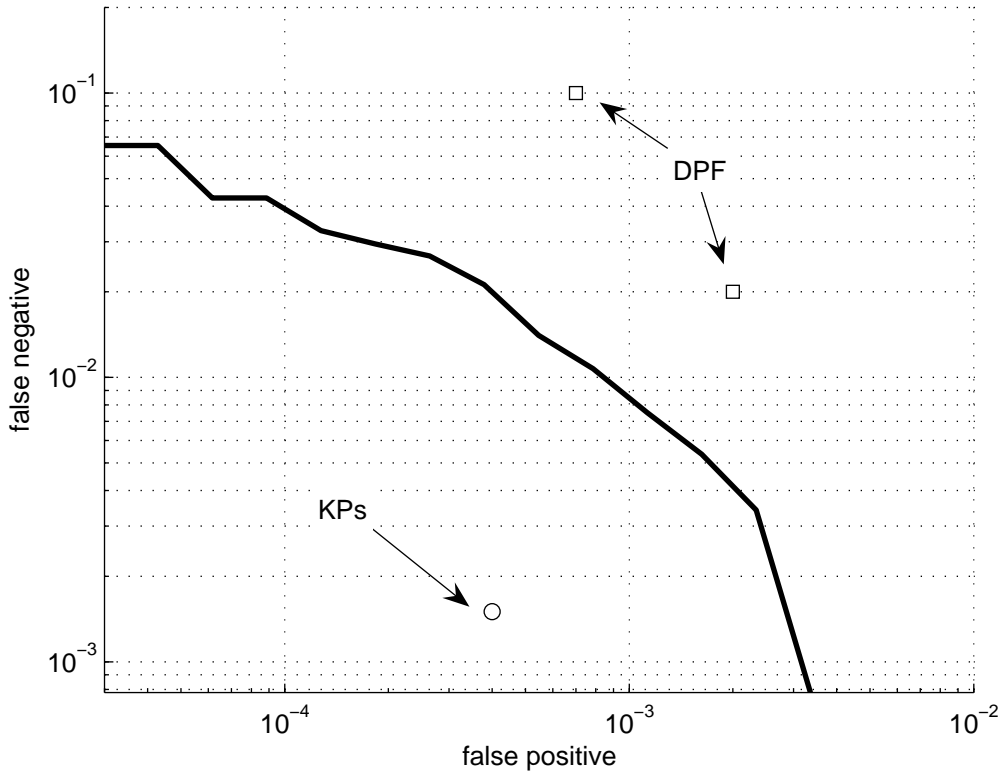


Figure 3. DET Curve for the MM270k database.

The Receiver Operating Characteristic (ROC) curve¹⁹ is often used to represent the tradeoff between error types. In this study, we use a variant of the ROC curve called Detection Error Tradeoff (DET) curve.²⁰ In DET curve error rates are plotted on both axis. Moreover, both axis use a log scale.

The average DET curve is represented on Fig. 3. Square points represent the performance of DPF⁶ and the circle point that of KPs.⁵ For DPF, the two points are obtained by inspecting the ROC graphs and computing the corresponding points on the DET Curve. For the second method, the point is computed using the available information from.⁵

It can be seen that the proposed method achieves very good performance. For instance, a false negative rate of 5 % corresponds to a false positive rate of $6 \cdot 10^{-5}$. On the one hand, our method outperforms that of DPF. Moreover, it should be noted that the features used in the current work are mainly a subset of those used in DPF: we use 138 features against 298 in the latter study. On the other hand, the proposed method is outperformed by KPs. In our method, most of the wrongly classified replicas (false negative errors) correspond to replicas for which the illumination or the intensity have been changed to a great extent. The method KPs uses features that are invariant to this change but that are more computationally complex to extract.

4. CONCLUSIONS

In this paper, we have described a technique to classify whether a test image is a replica of a given reference image. We performed experiments on a large database containing 18,785 photographs representing a wide range of content. We were able to detect about 95 % of the replicas while achieving false positive rate of only $6 \cdot 10^{-5}$.

ACKNOWLEDGMENTS

The first author is partly supported by the Swiss National Science Foundation — “Multimedia Security”, grant number 200021-1018411. The work was partially developed within VISNET, a European Network of Excellence (<http://www.visnet-noe.org>), funded under the European Commission IST FP6 programme.

REFERENCES

1. F. Hartung and M. Kutter, “Multimedia watermarking techniques,” *Proceedings of the IEEE* **87**, pp. 1079–1107, July 1999.
2. F. Lefebvre, B. Macq, and J.-D. Legat, “Rash: Radon soft hash algorithm,” in *EURASIP European Signal Processing Conference*, (France), September 2002.
3. J. Seo, J. Haitisma, 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. Y. Ke, R. Sukthankar, and L. Huston, “An Efficient Parts-Based Near-Duplicate and Sub-Image Retrieval System,” in *ACM International Conference on Multimedia*, pp. 869–876, 2004.
6. A. Qamra, Y. Meng, and E. Y. Chang, “Enhanced Perceptual Distance Functions and Indexing for Image Replica Recognition,” *IEEE Transaction on Pattern Analysis and Machine Intelligence* **27**, pp. 379–391, March 2005.
7. 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.
8. B. S. Manjunath and W. Y. Ma, “Texture Features for Browsing and Retrieval of Image Data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**, pp. 837–842, August 1996.
9. M.-K. Hu, “Visual Pattern Recognition by Moment Invariants,” *IEEE Transaction on Information Theory*, pp. 179–187, 1962.
10. D. H. Ballard, *Robot Vision*, McGraw-Hill, 1986.
11. J.-L. Leu, “Computing a Shape’s Moments from its Boundary,” *Pattern Recognition* **24**(10), pp. 949–957, 1991.
12. Y. Maret, G. G. Molina, and T. Ebrahimi, “Identification of Image Variations based on Equivalence Classes,” in *Visual Communications and Image Processing (VCIP) 2005*, SPIE, July 2005.
13. B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett, “New support vector algorithms,” *Neural Networks* **22**, pp. 1083–1121, 2000.
14. C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery* **2**(2), pp. 121–167, 1998.
15. D. L. Donoho, “High-dimensional data analysis: The curses and blessings of dimensionality,” August 1998.
16. F. A. P. Petitcolas and M. Kutter, “Fair Evaluation Methods for Image Watermarking Systems,” *Journal of Electronic Imaging* **9**, pp. 445–455, October 2001.
17. J. Smith and A. Natsev, “Spatial and Feature Normalization for Content Based Retrieval,” in *IEEE International Conference Multimedia and Expositions*, **1**, pp. 193–196, 2002.
18. K. Duan, S. S. Keerthi, and A. N. Poo, “Evaluation of simple performance measures for tuning SVM hyperparameters,” *Neurocomputing* **51**, pp. 41–59, 2003.
19. T. Fawcett, “ROC Graphs: Notes and Practical Considerations for Data Mining Researchers,” in *Technical Report HPL-2003-4*, HP Labs, 2003.
20. A. Martin, G. Daggintgton, T. Kamm, M. Ordowski, and M. Przybocki, “The DET Curve in Assessment of Detection Task Performance,” in *Proc. Eurospeech ’97*, pp. 1895–1898, (Rhodes, Greece), 1997.