

EFFICIENT IMAGE DUPLICATE DETECTION BASED ON IMAGE ANALYSIS

THÈSE N° 3797 (2007)

PRÉSENTÉE LE 11 MAI 2007

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Laboratoire de traitement de signaux 1

PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Yannick MARET

ingénieur électricien diplômé EPF
de nationalité suisse et originaire de Conthey (VS)

acceptée sur proposition du jury:

Prof. S. Süssstrunk, présidente du jury

Prof. T. Ebrahimi, directeur de thèse

Prof. H. Bunke, rapporteur

Prof. F. Leprévost, rapporteur

Prof. M. A. Shokrollahi, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2007

Abstract

This thesis is about the detection of duplicated images. More precisely, the developed system is able to discriminate possibly modified copies of original images from other unrelated images. The proposed method is referred to as content-based since it relies only on content analysis techniques rather than using image tagging as done in watermarking.

The proposed content-based duplicate detection system classifies a test image by associating it with a label that corresponds to one of the original known images. The classification is performed in four steps. In the first step, the test image is described by using global statistics about its content. In the second step, the most likely original images are efficiently selected using a spatial indexing technique called R-Tree. The third step consists in using binary detectors to estimate the probability that the test image is a duplicate of the original images selected in the second step. Indeed, each original image known to the system is associated with an adapted binary detector, based on a support vector classifier, that estimates the probability that a test image is one of its duplicate. Finally, the fourth and last step consists in choosing the most probable original by picking that with the highest estimated probability.

Comparative experiments have shown that the proposed content-based image duplicate detector greatly outperforms detectors using the same image description but based on a simpler distance functions rather than using a classification algorithm. Additional experiments are carried out so as to compare the proposed system with existing state of the art methods. Accordingly, it also outperforms the perceptual distance function method, which uses similar statistics to describe the image. While the proposed method is slightly outperformed by the key points method, it is five to ten times less complex in terms of computational requirements.

Finally, note that the nature of this thesis is essentially exploratory since it is one of the first attempts to apply machine learning techniques to the relatively recent field of content-based image duplicate detection.

Keywords: copyright infringement detection, illegal image detection, duplicate detection system, image analysis, machine learning, multidimensional indexing

Version Abrégée

Cette thèse concerne la détection de copies d'images. Plus précisément, la présente thèse propose l'étude d'un système permettant de détecter les copies d'images connues du système, même si celles-ci ont été légèrement modifiées. La technique proposée est basée sur le contenu car elle utilise des techniques d'analyse d'image plutôt que le marquage comme cela se fait dans le watermarking.

Le système de détection de copie d'image proposé classe une image teste en l'associant avec un label qui correspond à une des images originales connues. La classification est effectuée en quatre étapes. Dans la première étape, l'image test est décrite en utilisant des statistiques globales liées à son contenu. La deuxième étape consiste à sélectionner, en utilisant une technique d'indexation spatiale appelée R-Tree, les images originales qui ont les plus grandes probabilités d'être les originaux de l'image de test. Dans la troisième étape, des détecteurs binaires sont utilisés pour estimer les probabilités que l'image de test soit une copie des images originales sélectionnées à la deuxième étape. En effet, chaque image originale connue du système est associée à un détecteur binaire adapté, basé sur une machine à vecteurs de supports, qui permet d'estimer la probabilité qu'une image de test soit une de ses copies. Finalement, la quatrième et dernière étape consiste à choisir l'image originale la plus probable en sélectionnant celle ayant la plus haute probabilité estimée. Des expériences comparatives ont montré que le système proposé obtient de meilleures performances qu'un système utilisant des descriptions d'image similaire mais basé sur une fonction de distance plus simple en lieu et place de l'algorithme de classification. De plus, des expériences supplémentaires ont permis de comparer le système développé à des méthodes appartenant à l'état de l'art de la détection de copies basée sur le contenu. Le système est bien meilleur que la méthode appelée fonction de distance perceptuelle qui, de plus, utilise une description d'image plus compliquée. Bien que le système proposé soit légèrement moins performant que la méthode appelée points clés, cette dernière est cinq à dix fois plus complexe du point de vue computationnelle.

Finalement, l'auteur aimerait souligner la nature essentiellement exploratoire de cette thèse vu qu'elle est l'une des premières approches appliquant des techniques d'apprentissage automatique à la détection de copies d'images basée sur le contenu.

Mots-clés: détection de copies illégales, détection de matériels illégaux, système de détection de copies, analyse d'image, apprentissage automatique, indexation multidimensionnel

Remerciements

Cette thèse est le résultat de quelques années de labeur plus ou moins acharné que je n'aurais pas pu accomplir sans l'aide, le soutien, et les encouragements de nombreuses personnes.

En premier lieu, j'aimerais exprimer ma gratitude au Professeur Touradj Ebrahimi qui m'a donné la possibilité de faire une thèse au sein de son groupe. Je lui suis particulièrement reconnaissant pour la liberté académique et la confiance inconditionnelle qu'il m'a accordées. De plus, les tâches variées qu'il m'a confiées ont permis d'enrichir mon parcours professionnel de manière inestimable. Je suis aussi spécialement redevable au Docteur Frédéric Dufaux pour les nombreuses discussions scientifiques que nous avons eues ensemble et qui m'ont permises d'avancer sûrement dans ma recherche. Ma gratitude va aussi aux membres du jury de thèse, Professeure Sabine Süsstrunk, Professeur Horst Bunke, Professeur Frank Leprévost, et Professeur Amin Shokrollahi. J'aimerais spécialement remercier Sabine pour le remplacement au pied levé qu'elle a dû effectuer à la présidence du jury. Une partie des idées proposées dans cette thèse ont été développées durant mon séjour en Grèce en automne 2005. Pour leurs accueils chaleureux lors de cette occasion, j'aimerais remercier le Professeur Ioanis Pitas, le Docteur Nikos Nikolaidis ainsi que Spiros Nikolopoulos. Pour l'aide et le support reçu lors des tâches administratives et informatiques, mes plus sincères remerciements vont à Marianne Marion, Gilles Auric, et Simon Châtelain. Pour les nombreuses discussions techniques que nous avons partagées, j'aimerais enfin remercier tous mes collègues au sein de l'institut de traitement des signaux. En particulier, je salue Ulrich, mon ancien collègue de bureau, ainsi que David et Petit Suisse, mes nouveaux collègues de bureau.

Comme une thèse n'est pas faite que de dur labeur, j'aimerais remercier mes collègues et amis pour tous les bons moments que nous avons passés ensemble. Par exemple, j'ai pu affronter de nombreux joueurs de babyfoot durant la première demi-heure de midi, notamment Christophe, David, Julien, Mathieu, Nawal, Philippe, Ulrich, et Yann. Ou encore, grâce à Dan, David, Jonas, surtout Julien, Lorenzo, Mathieu, Patricia, et Yann, j'ai pu à nouveau pratiquer un sport que j'adore, l'escalade. De plus j'aimerais encore remercier les ingénieurs du violon, Nicolas et Olivier, qui m'ont fait un grand plaisir en officiant lors de mon mariage.

Enfin, ma plus grande gratitude va à ma famille, en particulier à mes parents, Gabriel et Germaine, mes trois soeurs, Roselyne, Marie-Françoise, et Anne-Murielle, ainsi qu'à ma femme Marie-Carmen car sans amour nous ne sommes rien...

Contents

1	Introduction	1
1.1	Motivations	1
1.2	Investigated Approach	2
1.3	Main Contributions	3
1.4	Organisation of the document	4
I	Background	5
2	General Material	9
2.1	Visual information description	9
2.1.1	Colour	10
2.1.2	Texture	11
2.1.3	Region	11
2.1.4	Salient point	12
2.2	Multimedia database indexing	13
2.2.1	Multidimensional access	13
2.2.2	Point access methods	14
2.2.3	Spatial access methods	14
2.3	Classification	15
2.3.1	An overview of statistical machine learning	15
2.3.2	Estimating the expected risk	17
2.3.3	From one-class, or two-class, to N -class classifiers	18
2.3.4	An introduction to support vector classifiers	19
2.4	Chapter summary	23
3	A State of the Art on Image Duplicate Image Detection	25
3.1	What is duplicate detection?	25
3.2	Two duplicate detection philosophies	26
3.2.1	Watermarking-based duplicate detection	26
3.2.2	Content-based duplicate detection	27

3.2.3	Applications and applicability	28
3.3	Content-based techniques	29
3.3.1	Fingerprinting techniques	31
3.3.2	Robust hashing techniques	37
3.3.3	Standardisation efforts	42
3.4	Chapter summary	42
II	Dissertation	45
4	A Framework for Content-based Image Duplicate Detection	49
4.1	Model of the duplicates of an image	49
4.2	Generic duplicate detection system	52
4.2.1	Generic duplicate detection — single original image system	52
4.2.2	Generic duplicate detection — multiple original image system	53
4.3	Performance evaluation methods	54
4.3.1	Test images	54
4.3.2	Performance metrics	58
4.4	Approach overview and common components	63
4.4.1	Image preprocessing	64
4.4.2	Features	65
4.5	Chapter summary	67
5	A Single Original Duplicate Detection System	69
5.1	System overview	69
5.2	Remarks on training	71
5.3	Binary detector	73
5.3.1	Features projection	74
5.3.2	Normalisation	74
5.3.3	Decision Function	75
5.4	Results	79
5.4.1	Baseline	79
5.4.2	Influence of the F-score metric parameterisation	83
5.4.3	Distribution of the false negatives error rates for no false positive error	84
5.4.4	Requirements on storage and computational effort	87
5.4.5	Comparison with existing duplicate detection methods	90
5.5	Exploratory works	92
5.5.1	Optimal training examples	92
5.5.2	Combining classifiers	93
5.6	Chapter summary	95

6	Multiple Original Images Duplicate Detection System	97
6.1	Approach motivation	97
6.2	System overview	98
6.3	Remarks on training	99
6.4	Pre-classifier	99
6.4.1	Feature projection and normalisation for indexing	100
6.4.2	R-Tree indexing	101
6.5	Results for the pre-classifier	104
6.5.1	Baseline	104
6.5.2	Scalability	109
6.6	Results for the system	113
6.6.1	Performance	113
6.6.2	Requirements on storage and computational effort	117
6.6.3	Comparison with existing duplicate detection methods	118
6.7	Exploratory and future works	119
6.7.1	Random projection	119
6.7.2	Hierarchical duplicate detection	120
6.8	Chapter summary	122
7	General Conclusions	123
7.1	Summary of the achievements	123
7.2	Perspectives	125
	Bibliography	127
	Curriculum Vitæ	137
	Personal Publications	141

Notations

Mathematical symbols

The used mathematical symbols are defined in the following list. Additionally, a more complete definition is also given the first time that the symbol appears within the text. Note that only the symbols used more than once are defined here.

\mathbf{f} vectors;

\mathbf{I} matrices;

$f(\cdot)$ functions;

\mathcal{S} sets;

\mathbb{R} fields;

$\Pr\{a > b\}$ probability;

$I_{\mathcal{A}}(x)$ indicator function: equals one if $x \in \mathcal{A}$ and zero otherwise;

s.t. such that;

$R(f)$ expected risk associated to classification function f ;

$R_{emp}(f)$ empirical risk associated to classification function f (and examples);

\mathbf{w} separating hyperplane;

y_i i -th label;

\mathbf{x}_i i -th example;

b margin;

ξ_i i -th slack variable;

C tradeoff parameter in C support vector machine;

ν tradeoff parameter in ν support vector machine;

α_i i -th Lagrange's multiplier in the dual form solution of support vector machine minimisation;

$f(\mathbf{z})$ decision function for test pattern \mathbf{z} ;

$\ker(\mathbf{x}_i, \mathbf{x}_j)$ kernel function between patterns $\mathbf{x}_{i,j}$;

γ inverse kernel width for a radial basis function;

σ kernel width for a radial basis function (that is $1/\gamma$);

$\mathcal{D}(\mathbf{I})$ set of the duplicates of image \mathbf{I} (no composition);

$\mathcal{E}_n(\mathbf{I})$ set of the duplicates of image \mathbf{I} (composition of n transformations);

$g_n(\mathbf{I}, \mathbf{p})$ functional containing n sequential operations;

$\mathcal{F}(\mathbf{I}, n)$ set of the duplicates of image \mathbf{I} (using the functionals $g(\cdot)$);

$\mathcal{V}(\mathbf{I}, \mathbf{v})$ set of the duplicates of image \mathbf{I} (permitting permutation of the transformations' order);

\mathbf{T} test image;

u probability threshold;

N number of original images;

$d_{\mathbf{O}}^1(\mathbf{T}, u)$ duplicate detector function for the single original image \mathbf{O} ;

$d_{\mathcal{O}}^N(\mathbf{T}, u)$ duplicate detector function for the N original images contained in \mathcal{O} ;

\mathcal{O} set of original images;

\mathcal{T} set of test images;

\mathcal{F} set of unrelated test images;

\mathcal{L} set of labels (-1 for unrelated images, $+1, \dots, N$);

c_t true class of the test image;

c_e estimated class of the test image;

$c(c_t, c_e)$ error indicator function;

$\text{fp}(c_t, c_e)$ false positive indicator function;

$\text{fn}(c_t, c_e)$ false negative indicator function;

p_{FP}, \hat{p}_{FP} real and estimated probability of false positive;

p_{FN}, \hat{p}_{FN} real and estimated probability of false negative;

\mathcal{C} set of candidates selected by the pre-classifier ($\mathcal{C} \subseteq \mathcal{O}$);

$\mathbf{R}, \mathbf{G}, \mathbf{B}$ red, green and blue channels of an image;

- H, S, I** hue, saturation and intensity channels of an image;
- TP number of true positives;
- FP number of false positives;
- P number of positives;
- $F(TP, FP, P)$ f-score computed using numbers of true positives, false positives and positives;
- ρ ratio between positives and negatives;
- $F_\rho(\hat{p}_{FP}, \hat{p}_{FN})$ f-score computed using estimated probabilities of errors;
- f** feature vector representing an image;
- W** dimensionality reduction matrix;
- x** pattern feeded (transformed version of **f**) to the machine learning algorithm;
- δ box size used when searching the R-Tree.

Acronyms

The following list gives the acronyms used throughout this document. The acronyms are also defined the first time they appear within the text.

- SVC** support vector classifier
- VC-dimension** Vapnik Chervonenkis dimension
- LOO** leave-one-out
- CV** cross-validation
- RBF** radial basis function
- HSI** hue saturation intensity
- ROC** receiver operating characteristic
- DET** detection error tradeoff
- KP** key point
- DPF** perceptual distance function
- DCT** discrete cosine transform
- MSE** mean square error
- PCA** principal component analysis

ICA	independent component analysis
MPEG	moving picture experts group
JPEG	joint picture experts group
SVDD	support vector data description
StirMark	StirMark benchmark version 3.1
Qamra	Qamra benchmark
FN	false negative
FP	false positive
TN	true negative
TP	true positive
P	number of positives
N	number of negatives
CGFA	CGFA — virtual museum [Ke <i>et al.</i> , 2004] —
MM270k	MM270k — commercial image collection [Ke <i>et al.</i> , 2004] —
F-score	F-score
KNN	k nearest neighbour
NN	nearest neighbour

1

Introduction

1.1 Motivations

The relatively recent simplicity with which digital contents can be produced, processed, and distributed has opened a new era — the all-digital world. Unfortunately, this revolution has also exacerbated old problems and created new ones. For instance, illegal distribution of copyrighted, or illegal, materials is nowadays very easy to undertake. Another problem relates to the management of the wealth of available documents.

A specific problem concerns the duplication — exact or approximate, legal or illegal — of documents. Indeed, many documents are stored on multiple servers, and often different versions cohabit. For different reasons, it becomes then necessary to detect copies of a given document. The main reason is simply to reduce documents' management hassle. But there exist many secondary yet important reasons as exemplified in the following applications: monitoring — tracking of document circulating on the Internet for, among other purposes, royalty collection, statistic gathering, copyright infringement detection, and illegal material detection; clustering — regrouping documents that are duplicates when querying a database or the Internet; version search — searching the right version of a document among a database or the Internet. As it can be seen, duplicate detection has many useful applications, and the need for efficient duplicate detection grows as the number of generated digital document soars. To give an idea, IDC* estimated that humankind generated 161 billion gigabytes of digital information in 2006 while the University of California estimated that only five billion gigabytes were generated in 2003, additionally tallying how much space would be consumed if non-electronic information, such as analogue radio broadcasts or printed office memos, were digitised. The amount of duplicated documents among such a sum of data is certainly staggering, for example IDC assumed that, on

*<http://www.idc.com/>

average, each digital file gets replicated three times.

While it is relatively easy to detect exact duplicate, detecting slightly modified duplicate, or near-duplicate, is by far a more difficult task. For text, for example, the basic idea is to represent each document by a vector whose binary entries signal the presence, or the absence, of a given keyword within the document. Then, documents can be easily compared by matching their binary vectors. This kind of technique is, for example, used in search engine such as Google. On the other hand, near-duplicate detection becomes even more arduous in the case of images. Indeed, perceptually equivalent images can have very different representations. Furthermore, images can be modified in many ways while keeping their main perceptual features intact. And moreover, no grammar for image yet exists and, thus, the decomposition of an image, such as performed for text, still pertains to the domain of the fiction. In short, image duplicate detection is an interesting and exciting problem that is far from being solved and deserves further research.

1.2 Investigated Approach

This dissertation presents a system to detect duplicates of images based on their content. The underlying idea is to create an adapted duplicate detector for each image whose duplicates have to be detected. Most other works on content-based duplicate detection are centred on finding image's description robust to certain transformations. Rather than study this already much explored territory, we propose to research how to distinguish between duplicates of an image and unrelated images given possibly non-robust image descriptions. More precisely, instead of finding an image representation that fits our needs, we indeed develop a detector that suits the characteristic of a given original image and its duplicates.

The proposed duplicate detection system is developed in two stages. In the first stage, a set of binary detectors is created, each adapted to a particular original image. More precisely, each detector is able to distinguish between the duplicates of its original and unrelated images. They are composed of the three steps outlined thereafter. In the first step, the test image is preprocessed so as to add some degree of invariance against common image processing operations is added. In the second step, global statistics are used to describe the image. Finally, in the last step, a non-linear decision function, based on a support vector classifier, is used to determine the probability that the test image is a duplicate of the original image.

In the second stage, the binary detectors are efficiently put together to form a multiple original images duplicate detector. In other words, the system is able to determine whether a test image is a duplicate of one of the originals or unrelated to any of them. The full system is composed of the three steps outlined thereafter. In the first step, the most likely originals are efficiently selected by means of an adapted indexing technique. They form the set of candidates. In the second step, the binary detectors developed in the first stage are applied to each element of the set of candidates. Finally, the original corresponding to the highest probability, among the set of candidates, is selected. The system estimates that the test image is a duplicate of this original if the corresponding probability is higher than a certain threshold.

The basic idea of the proposed architecture is adaptability. For example, it adapts the detection

metric to each original image known to the system. Additionally, new original can be added to the system without needing to retrain the already known original. Moreover, the system can be readily adapted to novel duplicates if it is noticed that they escape detection.

1.3 Main Contributions

The significant contributions of the work presented in this dissertation are summarised below.

- *State of the art on content-based duplicate detection.* To the best of the author knowledge, the state of the art in this thesis is the first comprehensive report on existing content-based duplicate detection techniques.
- *Definition of the subspace spanned by the duplicate of an image.* The subspace is defined in several stages. In the first stage, parameterisable transformations of the image are considered and no composition is allowed. In this case, each transformation generates a curve in the image space, and the subspace spanned by the duplicates corresponds to the union of these curves. In the second stage, the effect of composed transformations is studied. In this case, the subspace spanned by the duplicates represents a manifold if operations' order does not matter. The manifold dimensionality is upper bounded by the number of considered transformations.
- *Definition of a generic duplicate detection system.* The duplicate detection system presented in this dissertation is organised around the idea of adaptive detection. In other words, the system knows the original images for which it has to detect duplicates and can adapt itself to each original's characteristics. More precisely, the proposed system is able to decide whether an input image is a duplicate of one of the originals contained in its collection or unrelated to any of them. Compare this approach with the classical image retrieval paradigm where images similar to the query are retrieved from a database. The original might be the query or contained in the databases. While the image retrieval approach is more flexible it is hardly adaptable to the intrinsic characteristics of each original and its duplicates.
- *Definition and performance quantification of adaptive binary duplicate detectors.* A binary detector is a duplicate detector adapted to a specific original image. It gives an estimation of the probability that the test image is a duplicate of the original image. It is based on a low-level visual description of the image and built around a support vector classifier.
- *Definition and performance quantification of a duplicate pre-classifier.* A duplicate detection system based on binary detectors checks a test image with every binary detector, each corresponding to a known original image. This becomes cumbersome as the number of originals grows. To avoid this, the duplicate pre-classifier efficiently selects a subset of the original images such that the test image is likely to be a duplicate of one of them. It is based on low-level visual description of the image and built around a spatial access method.
- *Performance quantification of the duplicate detection system.* The performance obtained by the entire system is assessed on different image collections. It is found that the proposed

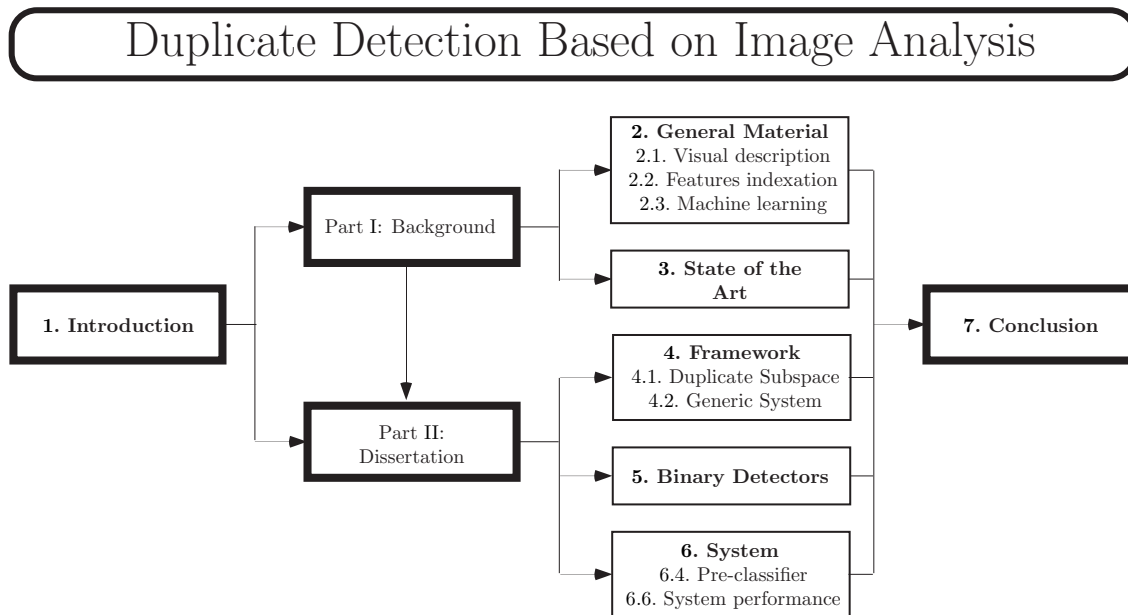


Figure 1.1: *Taxonomy of the content of the dissertation.*

approach performs very well when compared to state of the art content-based duplicate detection systems.

1.4 Organisation of the document

The taxonomy of the thesis's content is given in figure 1.1. The document is split into five chapters regrouped within two parts, namely background and dissertation. The last part, dissertation, forms the main body of the document and introduces our work: duplicate detection based on image analysis.

Background knowledge related to our work is reviewed in part I. More precisely, chapter 2 gives short introductions to general material such as image description, indexing, or machine learning. It is presented for the sake of completeness. Chapter 3, on the other hand, deals directly with the thesis topic by analysing the existing content-based duplicate detection techniques. Additionally, the content-based approach of duplicate detection is compared to watermarking.

Part II contains the dissertation on our framework for adaptive duplicate detection. More precisely, chapter 4 discusses adaptive duplicate detection in general. It is composed of two parts. In the first part, we define a generic framework for duplicate detection systems — including the duplicate subspace definition and the generic system. In the second part, we give an overview of the proposed duplicate detection system. Chapter 5 deals with our approach to image duplicate detection of a single original image or, more precisely, about binary detectors. It includes a performance analysis of the binary detectors. Chapter 6 extends the method presented in the previous chapter to multiple original images. It presents the pre-classifier algorithm and the performances' analysis of the pre-classifier and that of the whole system.

Part I

Background

As for me, all I know is that I know nothing.
Attributed to the Greek philosopher Socrates
(*cerca* 470 B.C. — 399 B.C.)

General Material

2

This chapter presents some key concepts and algorithms necessary to better understand the remaining chapters. The text is organised in three unconnected parts, namely visual information description, multimedia databases indexing, and machine learning. Visual information description is presented in section 2.1. It gives an overview of the different possibilities to describe an image using low-level features. Multimedia databases indexing is introduced in section 2.2. It proposes a general overview of the methods used to efficiently organise and retrieve objects from databases that index multidimensional features. Finally, machine learning is presented in section 2.3. It introduces the reader to the field of machine learning, and more specifically, to that of supervised classification.

2.1 Visual information description

Visual descriptors give statistics about an image. A good descriptor permits to discriminate between similar and dissimilar images. Note that the notion of similarity highly depends on the application. For instance, similarity means “visually consistent images” in the framework of image retrieval while it signifies “visually nearly identical” in duplicate detection. There exist many published surveys on image description, the reader can refer to [Rui *et al.*, 1999; Smeulders *et al.*, 2000] for surveys centred around image description for content-based image retrieval applications.

In the following, four types of low-level image descriptors are presented. The first type of descriptors, introduced in section 2.1.1, relates to the colour content of the image. The second type of descriptors, brought in in section 2.1.2, concerns texture, which refers to a structured visual motif. Due to their simplicity, and their relatively low computational cost, colour and texture are two of the most widely used low-level descriptors in image retrieval. The third type of descriptors, presented in section 2.1.3, concerns region-based description, which not only

includes the description of regions using colour or texture, but also that of the region shape. Local descriptions of an image are richer and more discriminative than its global description. However, the main drawback of region descriptors is the necessity of segmenting to obtain meaningful regions. Finally, the fourth type of descriptors, introduced in section 2.1.4, relates to salient points, which permit to obtain local descriptions of images while avoiding segmentation.

2.1.1 Colour

Colour descriptors are maybe the most widely used features in image retrieval [Rui *et al.*, 1999; Stricker and Orengo, 1995]. The main reason is that colour descriptors are relatively robust to background complication and independent of the image size.

The colour histogram is the most common colour descriptor. It gives a quantised estimation of the probability distribution of the colour channels' intensities. While easy to compute and containing much information, histograms have three important drawbacks. Firstly, they are often sparse and consequently quite sensitive to noise. Secondly, since histograms are quantised versions of the underlying probability distributions, it is not straightforward to compare two histograms. Many distance functions can be used for this purpose, for example refer to [Niblack *et al.*, 1993; Swain and Ballard, 1990]. Thirdly, histograms are difficult to index due to their high-dimensional nature, refer to section 2.2 for more information on multidimensional access methods.

Colour moments are often used to avoid the quantisation effects brought by using histograms to estimate the probability distributions [Stricker and Orengo, 1995]. They are also more robust to noise. The main idea behind using moments instead of a histogram is that probability moments fully describe the underlying probability distribution. However, due to the numerical difficulties arising during the estimation of higher order moments, most practical colour descriptor are limited to the first (mean), second (variance) and third (skewness) central moments. The distance function used to compare moment descriptors is mainly based on the weighted Euclidian distance.

To take into account the perceptual impact of colours, colour sets are used in [Smith and Chang, 1995]. In this approach, the *RGB* colour space is first transformed in a perceptually more uniform colour space, for example *HSV*. Subsequently, the perceptually uniform colour space is quantised into bins such that each bin corresponds to a colour that can be unequivocally labelled by a human viewer. This approach is based on the idea that there exists a small number of colours that are almost never confused [Boynton, 1989].

Developed more recently, dominant colours provides a compact, and easy to index, colour descriptor [Deng *et al.*, 2001]. Moreover, it is a standard descriptor in MPEG-7 [Manjunath *et al.*, 2001]. More precisely, colours in an image are clustered, by means of vector quantisation, into a small number of representative, or dominant, colours. The feature descriptor consists of the representative colours, their percentages, their spatial coherency, and their colour variance. One of the advantages of dominant colour is that it can be indexed in the 3D colour space and so avoids the high-dimensional indexing problems associated with the traditional colour histogram. Nonetheless, a drawback related to dominant colour is the vector quantisation step that can be relatively costly in terms of computational resources.

Except for the dominant colour descriptor, colour descriptors do not generally take into account

the spatial distribution of the colours. It has been noticed that image retrieval systems based only on colour statistics tend to return too many false positive answers [Faloutsos *et al.*, 1994]. For this reason, several methods exist on how to add spatial information to the colour descriptors. One simple, yet effective, way to do so is to divide the image into sub-block and to describe the colour content of each sub-blocks [Faloutsos *et al.*, 1994]. While effective, this approach lack of efficiency since it requires quite a large storage space. Some indications about the spatial distribution of colour can be added to colour descriptors that classify the colours into categories [Smith and Chang, 1995]. This is achieved by computing two shapes characteristics for each colour category, namely spreadness and elongation [Hu, 1962; Leu, 1991]. The first characteristic measures the compactness of the spatial distribution of a colour category. The second gives the ratio between the shape length and width. Note that even if pixels assigned to a colour form totally disconnected components, this feature still captures useful information (namely the spatial distribution of these components).

2.1.2 Texture

Texture refers to a structured visual motif. It is an important component of any visual object like forests, clouds or mountains. Texture features quantify random yet structured intensity (or colour) variations. More precisely, features measure the variation of the intensity of a surface and quantify properties such as smoothness and regularity. Texture, like colour, is a powerful low-level descriptor for image description. Textures describe important information about the structural arrangement of surfaces as well as their relationship to their surroundings [Rui *et al.*, 1999].

Statistical techniques characterise textures by the statistical properties of the grey levels of the pixels. Typically, these properties are computed from the grey level co-occurrence matrix of the surface [Haralick *et al.*, 1973]. Many researchers explored this type of approach, and it was experimentally found out that contrast, inverse deference moment and entropy are the three properties that give the best discriminatory power [Gotlieb and Kreyszig, 1994].

Additionally, some researchers explored textures' description from an angle linked to the human visual system [Tamura *et al.*, 1978]. More precisely, Tamura *et al.* developed several computational approximations of properties that psychological studies found out to be of importance. These properties are coarseness, contrast, directionality, line-likeness, regularity and roughness. All the aforementioned properties have a visual interpretation whereas it is not always so for the properties extracted from the co-occurrence matrix (for example, entropy is not visually meaningful).

Finally, a more recent advance in texture characterisation concerns multi-scales approaches. For example, some techniques consist in describing the textures as simple statistics of the wavelet transform (namely mean, variance or skewness) of the wavelet coefficients distribution for each sub-band [Smith and Chang, 1994]. There exist a profusion of possible wavelet transforms. However, it was determined that the Gabor wavelet gives the best discriminatory power [Manjunath and Ma, 1996].

2.1.3 Region

A region is a visually, or even semantically, meaningful part of an image. Not only can a region be represented by its colour or texture (using descriptors presented in section 2.1.1 or section 2.1.2),

but also it can be described by its shape. The shape description can be divided into two categories, namely boundary or region based. The first category uses only the region's contour while the second category makes use of the entire region [Rui *et al.*, 1996]. In any cases, one needs the region boundary to be defined in order to describe a region. This requires either a manual or an automatic segmentation of the image. While having quite evolved during the recent years, fully automatic image segmentation remains quite the Sangraal's quest of image analysis [Smeulders *et al.*, 2000].

Many boundary-based shape descriptors are based on Fourier descriptor [Zahn and Roskies, 1972]. In other words, the Fourier transform of the boundary is used as the shape feature. Later, some researchers perfected the Fourier descriptors by adding robustness to noise as well as geometric transformation invariance [Rui *et al.*, 1996].

Many region-based shape descriptors use moment invariants. Seven transformation-invariant moments were first proposed in [Hu, 1962]. Most of the subsequent works use variations of these seven moment invariants. While most useful invariants are found by trial-and-error [Rui *et al.*, 1999], methods exist to automatically generate a given geometry's invariant [Kapur *et al.*, 1995]. Finally, most existing approaches do not consider if the invariant remains truly invariant after digitisation, however some works exist on this particular topic [Gross and Latecki, 1995].

2.1.4 Salient point

Description of salient points is a possible method to propose local descriptions while avoiding segmenting the image. In short, salient points methods concentrate the local description into a few feature vectors, each corresponding to a fixed region around the salient point. Salient points are nothing else than specific image's pixels whose descriptions are the most salient (with respect to some criteria), among all image's pixels.

Since the image's description is condensed into a limited number of feature vectors, the salient points should be selected so has to have great saliency and proven robustness [Smeulders *et al.*, 2000]. One early and very popular work on salient point detections is that of [Harris and Stephens, 1988] where corner of objects are detected. In this case, the notion of corners and edges is used to select the salient points rather than a measure of robustness. This leads to points that might not be very robust to image transformations. For this reason, saliency is often defined as the points that survive longest to some transformations, for example to gradually blurring the image [Lindeberg and Eklundh, 1992].

The currently most successful salient point descriptor is that presented in [Lowe, 2004]. In this approach, the salient points are the local extrema in a scale-space representation of the image (obtained through a series of Gaussian blurring of the image). Each point then describes, in an invariant manner, the edges' orientations contained within a region surrounding the salient point. In general, Lowe's method describes a typical image using a few hundreds salient points but for complex scenes, several thousand points can be required. This method has given rise to many variants; a notable one is [Ke and Sukthankar, 2004] where principal component analysis is used to reduce the dimensionality of the descriptors while still improving their discriminatory power. Nonetheless, while Lowe's descriptor achieves good performance, it is computationally expensive. For this reason, some researchers developed methods to reduce its computational cost.

For example, [Lejsek *et al.*, 2006a] diminishes the number of points necessary to describe an image while achieving better matching results. Finally, the modification of Grabner *et al.* achieves a speedup in the order of eight to ten with respect to Lowe's original by approximating the Gaussian blurring [Grabner *et al.*, 2006].

2.2 Multimedia database indexing

Depending on the application, multimedia databases need different properties and need to support different types of queries. A retrieval query, or access method, on a multimedia database often requires the fast execution of a geometric search operation such as a point or region query. Both operations require fast access to those data objects in the database that occupy a given location in space. Additionally, multimedia objects often live in space containing many dimensions.

Many surveys exist on multidimensional indexing techniques used for multimedia databases. For more information, the reader is referred to [Boehm *et al.*, 2001; Gaede and Guenther, 1998]. Multimedia databases are of importance in many application areas such as geography, CAD, medicine, or image retrieval.

2.2.1 Multidimensional access

As seen previously, special multidimensional access methods are needed to support the search operations required by multimedia databases. The main problem in the design of such methods, however, is that there exists no total ordering among spatial objects so that spatial proximity is preserved. In other words, there is no mapping from two- or higher-dimensional space into one-dimensional space such that any two spatially close objects in the higher-dimensional space are also close to each other in the one-dimensional sorted sequence [Gaede and Guenther, 1998].

For this reason, the design of efficient access methods in multidimensional spaces is much more difficult than in traditional databases, where many efficient access methods are available. Examples of such one-dimensional access methods (also called single key structures) include the B-tree [Bayer and McCreight, 1972] and extendible hashing [Fagin *et al.*, 1979]. A popular approach to handling multidimensional search queries consists in using a single key structure per dimension. Unfortunately, this approach can be very inefficient since each index is traversed independently of the others. Consequently, the selectivity in one dimension can not be used to narrow down the search in the remaining dimensions [Kriegel, 1984]. In general, there is no easy and obvious way to extend single key structures in order to handle multidimensional data [Gaede and Guenther, 1998].

In other words, multimedia databases require real multidimensional indexing methods. Before continuing further, note that several mathematical effects can be observed as the dimensionality of the data space increases. Often, these effects cannot be intuitively reasoned out by simply extending two, or three-dimensional experiences, to high dimension spaces [Boehm *et al.*, 2001]. Some of the effects are only of mathematical interest while others have important implications on the performance of multidimensional index structures. Therefore, in the database world, these effects are summarised under the umbrella of the “curse of dimensionality [Donoho, 1998].” Qualitatively

speaking, important parameters such as volume and area depend exponentially on the number of dimensions of the data space. Consequently, many traditional indexing structures operate efficiently only if the number of dimensions is fairly small [Gaede and Guenther, 1998]. This means that most of them are unsuited to index multimedia databases.

Multidimensional data access methods can be classified into two categories, namely point and spatial access methods [Gaede and Guenther, 1998]. Point access methods are primarily designed to perform spatial searches on point databases in which only multidimensional points (without spatial extension) are stored. On the other hand, spatial access methods manage objects that have spatial characteristics in addition to their positions in the space. For instance, such objects are lines, polygons, or higher-dimensional polyhedra.

2.2.2 Point access methods

Generally, point access methods organise the point data in buckets, each corresponding to some sub-space of the universe. Some point access methods use one-dimensional hashing to index d -dimensional points. Although there is no total ordering of d -dimensional objects in one dimension, these methods use heuristic techniques to ensure that two objects close to each other in the multidimensional space are indexed the same [Nievergelt *et al.*, 1984]. Other point access methods use hierarchical data structures to manage point data [Bentley, 1975]. Finally, access methods such as the Buddy tree [Seeger and Kriegel, 1990] are hybrid since they incorporate both hierarchical and hashing techniques.

2.2.3 Spatial access methods

Point access methods cannot directly manage objects with a spatial extent but they are often extended to cover this need. Gaede and Guenther classify point access methods according to the techniques used to extend from point to spatial access methods. The most important extension approaches are outlined thereafter.

Object mapping methods map geometric objects into points in a higher-dimensional space. For instance, a rectangle in \mathbb{R}^2 corresponds to a point in \mathbb{R}^4 . Subsequently, existing point access methods are used to manage the points.

Object bounding methods are the most popular spatial access methods. In these approaches, the space is decomposed in a hierarchical manner. Objects are stored at the leaves of the hierarchical structures and intermediate nodes are used to perform efficient queries. Since the spatial extension of the nodes at the same level may overlap each other, the number of paths that have to be followed during a query varies. The most promising object bounding methods are the R-tree [Guttman, 1984] and R*-tree [Beckmann *et al.*, 1990].

Like the object bounding methods, clipping methods use hierarchical data structures. However, clipping is used to prevent overlapping of intermediate nodes at the same level. More precisely, objects are clipped, or subdivided, and stored in several nodes to guarantee non-overlapping intermediate nodes. By this mean, only one path of the hierarchical structure is traversed during a query [Sellis *et al.*, 1987].

2.3 Classification

Classification consists in associating an object to one of several classes according to a meaningful classification rule. A possible formalisation of this problem is as follows. Each class is represented by a label $y_i \in \mathcal{L} \subset \mathbb{Z}$, an object is modelled by a vector $\mathbf{x} \in \mathbb{R}^d$ denominated a pattern, and the classification rule is given by the function $f : \mathbb{R}^d \rightarrow \mathcal{L}$ mapping any d -dimensional vector \mathbf{x} to a label y . Machine learning provides many automatic methods for efficiently designing classification functions based on examples. Learning algorithms can be differentiated by the amount of information that is provided to them, namely supervised methods are supplied with examples in the form of couples (\mathbf{x}_i, y_i) while non-supervised algorithms have only access to the vectors \mathbf{x}_i .

In the following, we are mainly interested with supervised algorithms. The presentation consists in two parts. The first part introduces algorithm-independent machine learning notions while the second part gives an overview of a particular yet powerful classification technique, namely support vector classifier. Except for section 2.3.3, only the two-class case is treated; in other words, $|\mathcal{L}| = 2$.

Most of the notions presented thereafter are standards among the machine learning field, and can be found in any good book or tutorial on the topic. For example, the interested reader can refer to [Duda *et al.*, 2001] for a general introduction to classification, to [Vapnik, 2000] for a thorough coverage of statistical machine learning, or to [Muller *et al.*, 2001] for an introduction to kernel-based classification methods.

2.3.1 An overview of statistical machine learning

From expected to empirical risk

A good classifier is mainly one that generalises well to unseen (or novel) patterns. In other words, a good classifier should map novel patterns to the correct labels with high probability. This notion of generalisation can be formalised by introducing a loss function $l : \mathcal{L} \times \mathcal{L} \rightarrow \mathbb{R}$ that characterises the cost of mapping a pattern to a wrong class. Assuming that the couples (\mathbf{x}, y) of novel patterns and labels are independently drawn from an unknown probability distribution $p(\mathbf{x}, y)$, the expected risk R associated to a classification function f reads

$$R(f) = \int l(y, f(\mathbf{x})) dp(\mathbf{x}, y). \quad (2.1)$$

In theory, it suffices to select the function f , among a set of available classification functions \mathcal{F} , that minimises the expected risk to obtain the best classifier. In practice, however, several complications arise and approximations have to be carried out. For instance, using the loss function $l(y, \hat{y}) = I_{\{y\}}(\hat{y})$ gives the expected average number of classification errors. While theoretically attractive, this loss function leads to intractable optimisation problems. To obtain practically feasible algorithms the indicator function is usually approximated by smooth functions that are lower and upper bounded by 0 and 1, respectively. A further approximation concerns the expected cost, which cannot be practically computed since the underlying probability distribution p is unknown. To solve this problem, the empirical risk (average cost on the training set) is often used

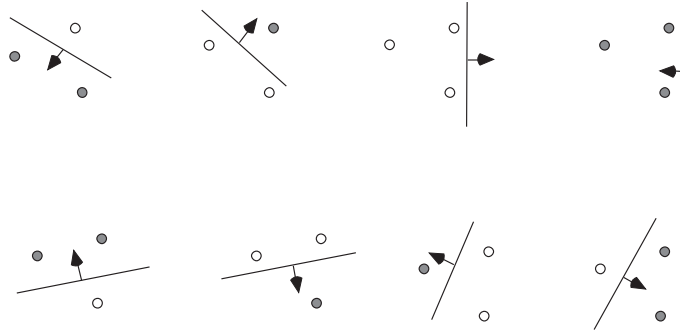


Figure 2.1: *Three points in \mathbb{R}^2 shattered by oriented half-planes.* The eight possible labels assignments can be correctly classifier using diverse oriented half-planes, which implies that the VC-dimension of the set of oriented half-planes is at least three (actually exactly three since the shattering is rendered impossible by the addition of a fourth point). In general, the VC-dimension of half-planes in \mathbb{R}^n is $n + 1$. In addition, note that $n + 1$ aligned points in \mathbb{R}^n cannot be shattered by oriented half-planes. This figure is courtesy of Burges [Burges, 1998].

instead of the expected risk

$$R_{emp}(f) = \frac{1}{M} \sum_{i=1}^M l(y_i, f(\mathbf{x}_i)). \quad (2.2)$$

In this case, a classification function f can be found as follows

$$f = \arg \min_{g \in \mathcal{F}} R_{emp}(g), \quad (2.3)$$

where the \mathbf{x}_i are M known examples, and the y_i are the corresponding labels. An interesting question is whether minimising the empirical risk leads to a minimal expected risk. The answer is affirmative only if the size of the training set tends to infinity. If however the number of training examples is limited, the minimised empirical risk can become smaller than the minimal expected risk. Consequently, minimising the empirical risk can be suboptimal and the classifier might not generalise as well as expected.

Capacity and its link to the number of training examples

For most applications the number of the training examples is limited and consequently minimising the empirical risk is likely to be suboptimal. The minimum number of training examples, needed to find a classification function f that performs well, usually depends on the capacity of the set of functions \mathcal{F} . The capacity measures the intrinsic complexity of a set of functions. A concrete example of a capacity measure is the Vapnik Chervonenkis dimension (VC-dimension) that quantifies the maximum number of points that can be shattered by a set of functions. More precisely, if for a given set of h points each of the 2^h possible label assignments is correctly labelled by a function of \mathcal{F} then the h points are shattered by \mathcal{F} . To illustrate this point, let us consider the three points and the set of oriented half-planes depicted in figure 2.1.

While theoretically interesting the VC-dimension is not practical since difficult to compute in most cases. Nevertheless it brings an important hindsight. More precisely, the VC-dimension h of

a set of functions \mathcal{F} permits to bound the minimal expected risk by

$$\min_{g \in \mathcal{F}} R(g) \leq \min_{g' \in \mathcal{F}} R_{emp}(g') + \sqrt{\frac{h \left(\ln \frac{2M}{h} + 1 \right) - \ln \frac{\delta}{4}}{M}}, \quad (2.4)$$

where the inequality holds with a probability larger than $1 - \delta$ for $M > h$. If the number of examples M tends to infinity, the bound becomes tighter since the second term on the right hand side of the equation tends to zero. In this case, a function f that minimises the empirical risk also minimises the expected risk. For a fixed number of examples, however, the only way to obtain a tighter bound is to diminish the VC-dimension h . This observation leads to the following rule of thumb. If the number of training examples is small, the capacity of the set of functions should also be small. Conversely, if the number of examples grows, better classifiers should be obtained by using sets of functions with larger capacities. This rule is in accordance with the intuitive Occam's razor principle: "All things being equal, the simplest explanation is the best one."

Regularisation as a mean to control capacity

In case of a limited number of examples, a possible way to obtain good classifiers consists in having a parametric set of functions $\mathcal{F}(\lambda)$ where λ controls its capacity. While the direct definition of such a set is not trivial, it can be easily constructed indirectly using an approach called *regularisation*. Indeed, the regularised empirical risk minimisation reads as follows

$$f_\lambda = \arg \min_{g \in \mathcal{F}(\lambda)} R_{emp}(g) = \arg \min_{g \in \mathcal{F}} \sum_{i=1}^M l(y_i, g(\mathbf{x}_i)) + \lambda \cdot \Omega(g) \quad (2.5)$$

where the regularisation functional $\Omega : \mathcal{F} \rightarrow \mathbb{R}$ takes values proportional to the complexity of the functions g , and λ is a non-negative real. As a result, λ permits to effectively control the capacity of \mathcal{F} since the larger λ the smaller the capacity of the corresponding set of functions. In practice, the regularisation functional is often the L_2 -norm if the functions f are elements of a Hilbert space, in which case $\mathcal{F}(\lambda)$ biases more and more toward smooth functions as λ increases.

While the empirical risk usually decreases as the capacity increases, the expected risk first decreases to reach a minimum before increasing again. Figure 2.2 illustrates this phenomenon, note that the zone on the right of the optimal capacity corresponds to overfitted classifiers while that on the left corresponds to underfitted classifiers. The capacity entailing the optimal expected risk has to be found, which implies an estimation of the expected risk.

2.3.2 Estimating the expected risk

Most classifiers have free parameters that need to be tuned in order to obtain good results, often indirectly controlling the underlying set of functions' capacity. As stated in section 2.3.1, the optimal choice of these parameters corresponds to the minimal expected risk. It implies that the expected risk has to be estimated.

The cross-validation procedure is a popular technique for estimating the expected risk for arbitrary classification algorithms. In the k -fold cross-validation algorithm, the training patterns

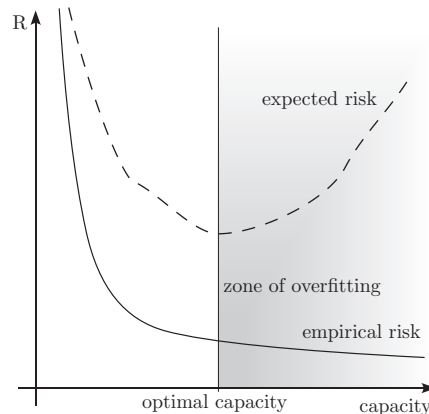


Figure 2.2: *The empirical and expected risks for different capacities.* The empirical risk decreases as the capacity augments while the the expected risk reaches a minimum before increasing again. The zone on the right hand of the optimal capacity corresponds to overfitted classifiers.

are randomly split into k mutually exclusive subsets (the folds) of approximately equal size. The classification function is obtained by training on $k - 1$ of the subsets, and is 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 risk. This method has been shown to yield a good estimation of the generalisation error [Duan *et al.*, 2003]. On the other hand, it entails many computations since the classifier needs to be trained $k + 1$ times instead of just once.

The leave-one-out estimate is an extreme case of the cross-validation technique. The leave-one-out estimate consists in using as many folds as there are training examples. While computationally expensive, it is known that the leave-one-out estimate is almost unbiased. There are many ongoing research on how to efficiently bound the leave-one-out estimate. Most of the bound depends however on the used classification technique.

2.3.3 From one-class, or two-class, to N -class classifiers

Two-class classifiers assign one of two classes to patterns. In this case, the labels are usually denoted $\mathcal{L} = \{-1, +1\}$. A special case of binary classifiers is the one-class classifier where a class support is estimated. One-Class classifiers can be seen as a two-class classifier for which the negative class span all possible patterns that do not belong to the positive class.

N -Class classifiers assigns one of N labels to patterns. An N -class classifiers can make use of a native N -class algorithms or they can be constructed using several two-class or one-class algorithms. Most machine learning algorithm are first designed for the simplest two-class problems and then extended to the N -class problem. For some algorithms, the extension is straightforward while for other approaches it is complex. In the latter case, the extension is often performed by combining several two-class classifiers. Additionally, the second approach is preferred if the number of classes is *a priori* unknown. In the following we are only interested by N -class classifiers obtained by combining several two-class, or one-class, classifiers.

There are two well-known ways to combine two-class classifiers in order to construct an N -class classifier, namely *one-vs-all* and *all-pairs* [Allwein *et al.*, 2000]. In both cases, an unknown pattern is classified with all classifiers and their outputs are combined in order to determine the associated class label. In the one-vs-all approach, there are N classifiers, each estimating the probability that a pattern falls in the corresponding class or not. The class whose classifier gives the highest probability is then used to label the pattern. In the all-pairs approach, there are $N(N - 1)/2$ classifiers, each corresponding to a possible pair of labels. For each class, an average probability is then computed and class with the highest probability is assigned to the pattern. The all-pairs approach becomes quickly impractical as the number of classes increases. Additionally, it is not possible to use it when the number of classes is *a priori* unknown.

Finally, note that an important requirement on combining binary classifiers is that the binary classifiers are calibrated. More precisely, if the binary classifiers output probability estimates, the estimates of the different classifiers have to be comparable to each other. In other words, a probability of, say 0.6, has to signify the same for every binary classifier. For more information on the topic, the reader is referred to [Zadrozny and Elkan, 2002].

2.3.4 An introduction to support vector classifiers

The support vector classifiers (SVCs) are a set of optimal margin classifiers, which are nowadays widely used. The following gives a brief introduction to some of the basic ideas underlying SVCs. A more detailed review, and other kernel-based learning algorithms, can be found in [Burges, 1998; Muller *et al.*, 2001; Schoelkopf *et al.*, 2000].

Linear support vector classifiers

We first consider the simple case where the training examples, drawn from two categories, can be exactly separated by a hyperplane. In this case, the training data are said to be linearly separable. In this instructive example, the SVC training algorithm chooses a separating hyperplane that maximises the Euclidean distances between the hyperplane and the closest training example. In the SVC literature, this distance is called the margin and the hyperplane maximising it is said to be optimal. The assumption underlying the maximisation of the margin is “the larger the margin, the better the generalisation of the classifier.” In other words, the probability that a novel pattern falls on the wrong side of the hyperplane is expected to be low by maximising the margin.

Let $y_i = \{-1, +1\}$ denote the class labels, $\mathbf{x}_i \in \mathbb{R}^d$ a feature vector, and \mathbf{w} the optimal hyperplane. It can be shown that the margin is equal to the quantity $\|\mathbf{w}\|_2^{-1}$; as a result the maximisation of the margin is achieved by minimising the quantity $\|\mathbf{w}\|$ or equivalently $\|\mathbf{w}\|^2$. Consequently, the hyperplane that maximises the margin is found through the following constrained optimisation problem

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1. \end{aligned} \tag{2.6}$$

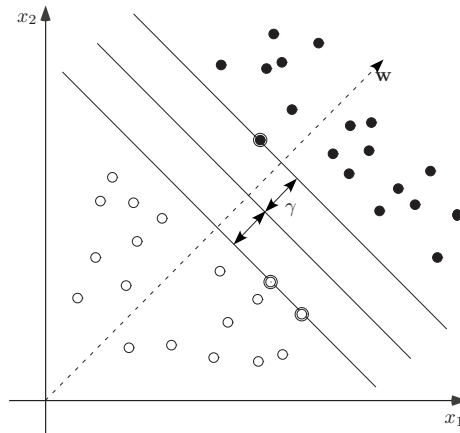


Figure 2.3: *Linear support vector classifier in two dimension for separable training examples.* The figure shows the margin $\gamma = 1/\|\mathbf{w}\|$ and the weight vector \mathbf{w} . The three points on the margin are called support vectors and fully define the solution. In other words, the solution does not change if the other points are moved and stay on the same side of the margin.

A geometrical interpretation of this optimisation problem, for the two-dimensional case, is depicted in figure 2.3.

Unfortunately, no solution (respecting all the constraints) exists when the data are not linearly separable. To deal with non-separable datasets, the constraints are relaxed by introducing non-negative slack variables ξ_i . There are several ways of introducing them; one possible realisation is called the *C-SVC* and uses a parameter $C \in \mathbb{R}^+$. The optimisation problem reads as follows

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C \cdot \sum_i \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{2.7}$$

As before, the margin's maximisation is performed by minimising $\|\mathbf{w}\|$ but this time the number of misclassified examples is controlled by $\sum_i \xi_i$. The parameter C controls the tradeoff between the number of misclassified examples and the maximisation of the margin. As C tends to infinity, the solution of equation (2.7) becomes equivalent to that of equation (2.6). Conversely, for small values of C some training examples are allowed to lie inside the margin, or even on the wrong side of the hyperplane.

Another possible realisation is called the ν -SVC and uses a parameter $\nu \in [0, 1]$. The constrained optimisation problem is given by

$$\begin{aligned} \min_{\mathbf{w}, \xi} \quad & \frac{1}{2} \|\mathbf{w}\|^2 - \nu\rho + \frac{1}{m} \cdot \sum_i \xi_i \\ \text{subject to} \quad & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq \rho - \xi_i, \\ & \xi_i \geq 0, \\ & \rho \geq 0. \end{aligned} \tag{2.8}$$

The minimisation problem proposed in equation (2.8) is less intuitive than that of the C -SVC given by equation (2.7). However, it turns out that theoretical meanings can be given to the parameter ν of equation (2.8), whereas the parameter C of equation (2.7) has no significant meaning. Indeed, it can be shown that not only is ν an upper bound on the fraction of training errors, but also it is a lower bound on the fraction of support vectors.

An equivalent dual formulation can be obtained by introducing a *Lagrange multiplier* α_i for each constraint in equation (2.8). The detailed derivation of the dual problem can be found in [Muller *et al.*, 2001], the resulting constrained optimisation problem is as follows

$$\begin{aligned} \max_{\alpha} \quad & -\frac{1}{2} \cdot \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \\ \text{subject to} \quad & 0 \leq \alpha_i \leq \frac{1}{m}, \\ & \sum_i \alpha_i y_i = 0, \\ & \sum_i \alpha_i \geq \nu. \end{aligned} \tag{2.9}$$

Note that a similar derivation also exists for the C -SVC.

The dual formulation permits to express the separating hyperplane \mathbf{w} as a weighted sum of the training examples, and to incidentally obtain a simple decision function

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i, \tag{2.10}$$

$$f(\mathbf{z}) = \text{sgn} \left(\sum_i y_i \alpha_i \mathbf{z}^T \mathbf{x}_i + b \right) \tag{2.11}$$

where the constant b is determined by the support vectors. More precisely, $b = y_k - \sum_i y_i \alpha_i \mathbf{x}_i^T \mathbf{x}_k$, for all \mathbf{x}_k such that $0 < \alpha_k < 1/m$.

In many SVC implementations, the dual formulation is used instead of the primal one given in equation (2.7) because it can be solved through standard quadratic programming. Additionally, many alternate (and often more efficient) schemes have been developed. Finally, the solution of the dual formulation permits to explain the concept of support vectors. Indeed, many of the optimal α_i in equation (2.10) and equation (2.11) are equal to zero in practice, which implies that only the \mathbf{x}_i corresponding to non-zero α_i actually define the optimal hyperplane and the decision function. For this reason, these \mathbf{x}_i are called support vectors.

Using the kernel trick to produce non-linear support vector classifiers

Since the training examples appear only as dot-products in equation (2.9), it is possible to construct non-linear decision boundaries by simply replacing the standard Euclidean dot-product by a kernel function $\ker(\mathbf{x}_i, \mathbf{x}_j)$. The non-linear kernel has to satisfy the Mercer's condition so as to be a dot-product in some space [Burges, 1998]. In this case, the kernel function represents the dot-product in a (higher-dimensional) space obtained through a non-linear mapping $\Phi(\cdot)$, such that $\ker(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \Phi(\mathbf{x}_j)$. Note that this non-linear mapping is often not explicitly known, as it

is sufficient that the kernel satisfies the Mercer's condition. Moreover, it can be shown that data from two categories can always be separated by a hyperplane by using an appropriate non-linear mapping to a sufficiently high dimensional space.

The Gaussian radial basis function (RBF) kernel and the polynomial kernel are two widely used mapping functions. They are, respectively, given by

$$\ker(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \quad (2.12)$$

$$\ker(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d. \quad (2.13)$$

It can be shown, for example [Burgess, 1998], that the Gaussian radial basis function maps the features into a space of infinite dimension, while the polynomial kernel maps the features into the space of all monomials up to degree d . In the absence of any a priori information hinting otherwise, the Gaussian radial basis kernel should be considered first [Hsu *et al.*, 2003]. This particular choice is motivated by several considerations. Not only is the linear SVC a particular case of the RBF kernel, but also the sigmoid and the RBF kernels behave similarly for certain choices of parameters [Keerthi and Lin, 2003]. Additionally, the RBF kernel presents less numerical difficulties than, for instance, the polynomial kernel. Finally, the RBF kernel is governed by only one parameter instead of two for the polynomial kernel. The kernel parameter σ controls the complexity of the decision boundary.

Using cross-validation to determine good ν -support vector classifier parameters

While ν has an intuitive signification, it is not clear what should be its optimal value [Chen *et al.*, 2005; Steinwart, 2003]. It was shown that twice \bar{R} , a close upper bound on the expected optimal Bayes' risk, is an asymptotically good estimate [Steinwart, 2003]. While no such bound can be easily determined *a priori*, this theorem induces an algorithm to find a good ν by starting with the classification error of a well-trained classifier as an approximation of the optimal Bayes risk [Steinwart, 2003].

Unfortunately, a good *a priori* approximation of the optimal Bayes risk is not always available. In this case, good parameters for σ and ν can be estimated through a full grid search [Steinwart, 2003]. The procedure is divided in two steps: *coarse* and *fine* grid searches. In each step, a k -fold cross-validation is carried out for each feasible pairs (ν, σ) . The pair for which the estimated expected risk is the lowest is then chosen. The following tried pairs experimentally give good results

- Coarse search: (σ, ν) for $\nu = 0.05 \cdot 2^k, k = -4, \dots, 4$ and $\sigma = k, k = 1, \dots, 10$.
- Fine search: (σ, ν) for $\nu = \nu_1 \cdot (1 + k/6), k = -2, \dots, +2$ and $\sigma = \sigma_1 \cdot (1 + k), k = -2, \dots, +2$. Here, ν_1 and σ_1 denote the value determined in the first step.

2.4 Chapter summary

This chapter is about general materials that help to better understand the rest of the dissertation. It is composed of three unconnected parts: visual description, multimedia databases indexing, and machine learning.

The visual description part first accounts for the idea behind the description of images through low-level features. It then moves to present different existing types of low-level descriptors. These types are colour, texture, region, and salient points. It is pointed out that colour and texture are the most commonly used descriptors in image retrieval applications. Additionally, it is noted that region descriptors need the image to be segmented and are, for this reason, less interesting although they provide local descriptions of images. Finally, salient points are presented as methods providing local descriptions while avoiding the pitfall of segmentation.

The multimedia databases part first describes the difference between traditional and multimedia databases. It turns out that databases containing visual features require multidimensional access methods. This requirement signifies that conventional indexing methods, for example based on hashing, are not directly usable. Two types of multidimensional access methods are then presented. The first type of methods, called point access methods, is used to index multidimensional points while the second type of methods, called spatial access methods, is used to index multidimensional points that additionally possess a spatial extension.

Finally, the machine learning part first defines the classification problem. The text then gives an overview of statistical machine learning. It turns out that classification methods are evaluated using the expected risk, or the amount of error made when classifying novel patterns. In order to select a good classifier, the expected risk needs thus to be estimated, for example using cross-validation techniques. Additionally, most classification techniques are first designed for two classes and then extended to several classes. Finally, the last section of this part gives an overview of a popular classification technique, namely support vector classifier.

A State of the Art on Image Duplicate Image Detection

3

The problem of duplicate image detection originates from different — and often unrelated — fields. As a result different problem definitions and solutions exist. We first loosely define the duplicate detection problems in section 3.1. Two quite dissimilar solutions, namely watermarking and content-based duplicate detection, are then compared in section 3.2. Finally in section 3.3, existing content-based duplicate detection techniques are presented and analysed.

3.1 What is duplicate detection?

The definition of duplicate detection is now given. Duplicate detection is a task that aims at detecting the duplicates of an original image. Consequently, it is first necessary to define what a duplicate is. In short, a duplicate is a transformed version of an original artwork that keeps a similar visual value. In other words, ‘being a duplicate’ is a pairwise equivalence relationship that links the original to any of its variations through a transformation operation, for example, compression, brightness changes or cropping. By extension, if an image A is a duplicate of another image B and yet another image C is duplicate of image B , then image C is in turn a duplicate of image A .

Finally, the task of duplicate detection can be expressed as follows. Duplicate detection aims at detecting all the duplicates of a particular image among a collection of images. Or in a simplified form, duplicate detection’s goal is to determine whether two given images are duplicates of each other or unrelated to each other. This is a naive definition that contextualises this state of the art whereas a more formal one is given in chapter 4.

3.2 Two duplicate detection philosophies

Two very dissimilar duplicate detection philosophies exist, namely watermarking-based methods and content-based approaches. The watermarking approach consists in embedding a signature within the original image before the dissemination of the work. Duplicates of the original artwork can subsequently be detected by checking the signature's presence within images. On the other hand, the content-based approach relies, as suggested by its name, on the analysis of the image's content in order to extract relevant visual features. Duplicates are then identified when their features are close to those of the original image. In the next two subsections, these two philosophies are presented in more details and their advantages and drawbacks are analysed.

3.2.1 Watermarking-based duplicate detection

Historically, image duplicate detection has been mainly performed using watermarking techniques. The idea behind watermarking is rather simple: the content's copyright owner incorporates, in a robust and imperceptible manner, a secret signature within the image prior to its dissemination. The hidden signature serves two goals. Firstly, it permits the identification of the content owner in litigious cases. Secondly, it permits to detect copies of the content, for instance by browsing the Internet, and subsequently to determine whether a copy is legally or illegally used. Many books and surveys are available on watermarking as this field of signal processing becomes more mature [for example Barnett, 1999; Cox *et al.*, 2001; Cox and Miller, 2002; Hartung and Kutter, 1999].

Recently watermarking, as a mean to protect content, underwent strong criticisms. Herley started a debate on the shortcomings of watermarking with a controversial paper entitled "Why watermarking is nonsense [Herley, 2002]." The crux of Herley's argumentation is that protecting all objects in a small neighbourhood of the marked object, as performed in most published watermarking algorithms, is necessary but not sufficient. He continues by arguing that a useful watermarking algorithm needs to protect all valuable variations and not merely those that are close to the marked object. Other authors continued to add to this debate, for example [Barni, 2003a,b; Moulin, 2003] emphasised that watermarking is still a young field of signal processing, and that no method has yet been able to protect the content from all possible attacks. However, Moulin partly dismissed the "watermarking is nonsense" statement by noticing that it may be quite difficult to deliberately find a valuable transformation of the marked object that escapes detection. Additionally, he remarked that watermarking has been quite useful in low-security related applications, for instance in cable TV or message embedding, and that new methods may yet further the performance of watermarking algorithms. Finally, Barni quite interestingly asked "Why should we hide information within the data, when we could more easily use headers, or other means, to reach the same goals [Barni, 2003a]?"

It is the author opinion that watermarking has important shortcomings, as described in the following. While watermarking can be useful in certain situations, it cannot be regarded as a mean to protect the content in the long term unless the watermark does indeed protect all valuable variations of the marked object. For example, let us imagine that a photographer embeds

a watermark into one of its most valuable image and then sells the marked image to different clients. Subsequently, let us further imagine that a client finds a valuable transformation of the image that escapes detection. Now, this client of dubious ethics is empowered to redistribute the photographer's work in all impunity since it is no longer possible to detect this modified copies' copies by means of the watermark. In other words, once the mark has been removed from one object while keeping the object value, watermarking becomes useless as a mean to protect this object. Valuable unmarked copies of the object exist and, consequently, there is no more hindrance to illegally use this particular work. Finally, while it might be indeed quite hard to create a valuable copy that escapes detection, what prevents the use of the corresponding transformation on other works watermarked with the same algorithm? In short, watermarking is not a flexible duplicate detection approach in the sense that the mark is unchangeable and, thus, cannot be adapted in case of failure. On a different note, watermarking requires to embed a signature before distribution, which is not always practical, for example in the case of illegal images monitoring as presented in section 3.2.3, nor even tolerated because some artists might be reluctant to accept any kind of modifications to their works [Kalker *et al.*, 2001].

3.2.2 Content-based duplicate detection

As said before, content-based approaches rely on image analyses rather than message embedding. Most of the existing content-based approaches are based on the creation of an image summary, called hash or digest. The hashes are subsequently used to compare between images using a conventional L_1 distance. In the following, these methods are termed robust hashing. To the best of the author's knowledge, few content-based methods are unrelated to hashing. However, these particular works are of special interest since this thesis also aims at performing content-based duplicate detection without relying on hashing. In the following, the methods unrelated to hashing are termed fingerprinting. Fingerprinting techniques are basically of two kinds, as detailed in section 3.3.1, either several hashes are generated for the description of an image or the distance used to compare the hashes is not based on the conventional L_1 distance. In the first case, the similarity between images is not given by the distance between their hashes but rather by the number of hashes that match. In the second case, the distance metric is often adapted to the specificity of the pair of compared images. Note, however, that the usage of the term fingerprinting is peculiar to this thesis. Indeed, in the signal processing literature, fingerprinting often refers to any content-based duplicate detection technique or even to a particular application of watermarking, where the embedded message is used to store the identity of the digital content's buyer.

In general, content-based duplicate detection approaches are more flexible than watermarking techniques, and do not impact on the content. However, they also have their shortcomings. Indeed, while content-based techniques can be adapted faster than watermarking-based duplicate detection, for example to counter a novel duplicate generation algorithm, they are more prone to collisions, or in other words, to false detections. For example, typical watermarking algorithms achieve false detection rates in the order of one per million (or more). On the other hand, the best content-based algorithms only achieve false detection rates in the order of one per tens of thousand.

It is the author's opinion that watermarking techniques will stay ahead in term of false detection rate but that content-based techniques are going to close the gap. One of the main reasons lies in the two philosophies principal dissimilarity. Indeed, in watermarking, the embedded message is known and can be generated so as to avoid any ambiguity even in cases where images are very similar yet different in terms of contents. None of this is possible with content-based techniques. The only possibility to avoid any ambiguity is to on richer visual features and hope for better discriminative power. On the other hand, it is common practical knowledge that richer features often translate into features that are less robust and can thus change drastically when the image is modified.

3.2.3 Applications and applicability

Monitoring

Monitoring refers to the tracking of images for, among other purposes, royalty collection, statistic gathering, copyright infringement detection, and illegal material detection. This application is passive in the sense that it has no direct influence on the content; for example, it does not prevent an image to be displayed. In other words, the main function of this application is to observe and report [Kalker *et al.*, 2001].

Both watermarking and content based methods can be used to monitor image usage. For some applications, not only is it necessary to detect the content, but also it is needed to trace the distribution history. In this case, watermarking is the only solution since additional information has to be carried. On the other hand, legacy contents tracing is not possible with watermarking and, similarly, illegal images tracing cannot be solved by means of watermarking [Kalker *et al.*, 2001]. Indeed, watermarking requires embedding to be carried out before dissemination but, in the last case, the source is controlled by someone who benefits to remain anonymous.

An example of illegal material detection is given in [Penna *et al.*, 2005]. The police usually keep a collection of paedophilia-related images that were caught in the course of their investigations. They can then detect these known images by monitoring, for example, an Internet backbone or scanning the computer's content owned by a suspect. In this kind of applications, detection of known illicit content usage, is typical of content-based duplicate detection algorithms. On the other hand, there already exist systems [Fleck *et al.*, 1996; Wang *et al.*, 1997] that aim at detecting the presence of naked bodies within an image. They are usually based on low-level features such as skin detection or elongated objects detection. Consequently, they cannot make the difference between legal pornography and illegal pornography since this would require automatic image understanding at a level that today's technology cannot achieve.

Clustering

Clustering refers to regroup images that are duplicates when querying a database. Typically, if an image search engine, for instance Google or Yahoo image, is queried with popular keywords, such as Lenna or Britney Spears, many of the returned images are actually the same or slightly modified versions. This is typically illustrated in figure 3.1 where Google image is queried with the

keyword Lenna. Hence, it would be useful to group the actual duplicates under a single image, so as to not overwhelm the user with redundant information.

Both watermarking and content-based duplicate detection can be used to cluster the images returned by a query. However, watermarking usage for this application is quite awkward since it requires the images to be publicly watermarked for identification purposes. Again, the use of watermarking is clearly not possible for legacy content. On the other hand, this kind of application fits quite well to the content-based duplicate detection paradigm.

Version search

Version search refers to searching the right version of an image. For example, suppose that you only have a thumbnail version of a picture that you like. It would be hence quite interesting to be able to search an image database, for instance Google or Yahoo image, by querying it with the thumbnail and expect all the existing variations of this image.

Both watermarking and content-based duplicate detection can be used to perform search for image versions. Again, watermarking is less suited to the task than content-based duplicate detection for reasons similar to those given for clustering.

3.3 Content-based techniques

Content-based duplicate detection is still a relatively young field of signal processing since the first major publication dates back to the end of the nineties to the best of the author's knowledge. As a consequence, not many works have yet been published, and most of the published algorithms are not mature enough to properly assess their usability. Indeed, either the performance is relatively poor or the method's complexity is too high. Additionally, many reported works only perform cursory testing, for example using only a few images or a limited number of transformations. The remaining of the section presents the main contributions to content-based duplicate detection. The text is divided into two parts, namely fingerprinting and robust hashing as distinguished in section 3.2.2.

The performance of most content-based duplicate detection methods is assessed in terms of recall and precision, defined as follows

$$recall = \frac{\text{number of correctly detected duplicates}}{\text{total number of duplicates}}, \quad (3.1)$$















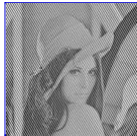



$$precision = \frac{\text{number of correctly detected duplicates}}{\text{total number of (correctly or wrongly) detected duplicates}}. \quad (3.2)$$

The transformations used to generate the test duplicates vary from one work to another. Many works use watermarking benchmarks but not all of them. Note that it has been argued that watermarking benchmarks might not be adapted to test content-based duplicate detection systems since they usually aim at producing duplicates whose embedded signatures are out-of-phase with that of the original. Additionally, the image collections used to estimate recall and precision are almost always different, be it in content or in size. Furthermore, the assessment methodology

Google [Web](#) [Images](#) [Groups](#) [News](#) [Scholar](#) [more >](#) [Sign in](#)

lenna [Advanced Image Search](#)
[Moderate SafeSearch is on](#) [Preferences](#)

Images Showing: All image sizes Results 1 - 18 of about 17,300 for lenna. (0.07 seconds)

					
Want to see the original Lenna? 400 x 225 pixels - 25k - jpg www.ee.cityu.edu.hk	lenna 158Kb 512 x 480 pixels - 159k - gif www.visgraf.impa.br	Original Lenna LENA 256 x 256 pixels - 43k - jpg www.mee.tcd.ie	lenna.jpg 512 x 512 pixels - 38k - jpg www.mathe.tu-freiberg.de	Index of afs sibp user kenta lenna-... 1024 x 1024 pixels - 383k - png stuff.mit.edu	Index of afs sibp user kenta lenna-... 1024 x 1024 pixels - 369k - png stuff.mit.edu [More results from stuff.mit.edu]
					
標準画像 lenna 輝度成分 512 x 480 pixels - 273k - gif www.mis.med.akita-u.ac.jp	Quantized Lenna Images 512 x 512 pixels - 159k - jpg web.mit.edu	click for larger view 800 x 600 pixels - 86k - jpg www.rootsweb.com	click for larger view 800 x 600 pixels - 71k - jpg www.rootsweb.com [More results from www.rootsweb.com]	Vista aerea di Lenna 352 x 290 pixels - 51k - jpg www.provinciaberghamasca.com	Cartina di Lenna 448 x 326 pixels - 62k - jpg www.provinciaberghamasca.com [More results from www.provinciaberghamasca.com]
					
lenna.png 512 x 512 pixels - 148k - png www.esat.kuleuven.be	Lenna 244 x 333 pixels - 28k - jpg vanilkijusty18.blog.onet.pl	lenna.medstr.jpg 512 x 512 pixels - 69k - jpg www.cvmt.dk	Panoramica sul centro di Lenna 802 x 605 pixels - 151k - jpg www.valbrenbanaweb.it	CUSSW Research Scientist: Lenna ... 200 x 220 pixels - 13k - jpg www.columbia.edu	Large faris-lenna.jpg 600 x 516 pixels - 59k - jpg lorelai.com

Result Page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [Next](#)

[Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

©2006 Google

Figure 3.1: Google image is queried with the keyword 'Lenna'. On the first page of returned by this query, ten out of sixteen images are actually variations of the original Lenna image (see figure 3.2a).

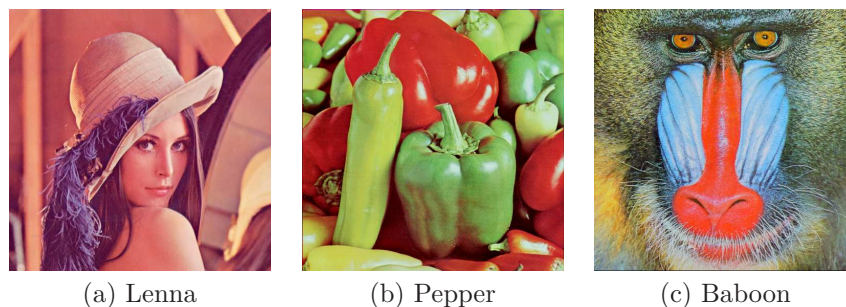


Figure 3.2: *Example of typical original images used for testing content-based methods.* These images can be downloaded from <http://sipi.usc.edu/database/>.

is also not constant since some works consider databases containing the original images while other considers that the originals are the queries to database containing suspect images. These discrepancies signify that it is quite difficult to objectively compare the performance of the existing algorithms. Finally, note that most works include the images shown in figure 3.2 in the set of original images.

3.3.1 Fingerprinting techniques

Fingerprinting relates to any technique that uses a summary of the image content but does not rely on a conventional distance metric to assess the similarity of two summaries. The boundary between fingerprinting and robust hashing, as previously defined in section 3.2.2, can be quite blurred. However, for simplicity sake, fingerprinting regroups methods that either generate several hashes for a single image or are based on non-conventional distance functions. In both cases, fingerprinting-based systems lead to more complex indexing techniques than hashing-based methods.

The general idea behind the fingerprinting techniques based on several hashes is now outlined while actual methods are described thereafter [Ke *et al.*, 2004; Lejsek *et al.*, 2006b; Lu and Hsu, 2005; Monga and Evans, 2004]. In these four approaches, each hash usually describes a particular region of the image. In other words, the description of the image is made richer. The number of regions, as well as their localisations and shapes, typically depends on the image content. It ranges from a handful of hashes to several thousands. Finally, two images are duplicates of each other if the number of matching hashes is above a certain threshold. Within the family of content-based duplicate detection approaches, these methods are by far those that obtain the best performance in terms of precision and recall. However, they often rely on complex features and require a great number of comparisons. To assess if two images are duplicate of each other, each hash of one image has indeed to be compared to each hash of the other image. For this reason, database indexing techniques play an important role in the computational efficiency of such approaches.

We now turn our attention to the general idea behind fingerprinting techniques based on non-conventional distance functions. Actual methods are described thereafter [Lefebvre *et al.*, 2003; Qamra *et al.*, 2005]. In these two approaches, the distance used to compare two feature vectors extracted from two images is not a metric-based function but rather a more complex function. More specifically, duplicates of an image do not necessarily lie within a hyper-sphere centred on that

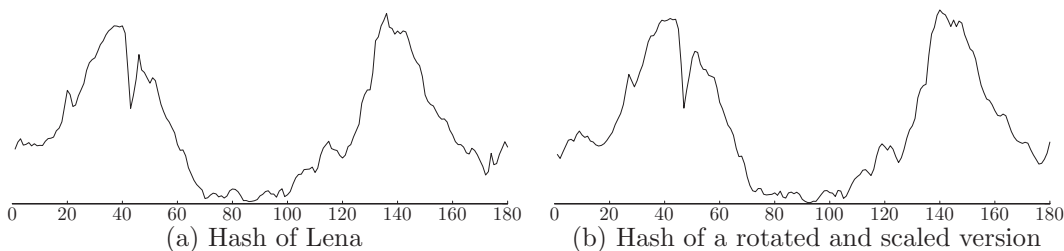


Figure 3.3: *Example of a hashes for the method proposed in [Lefèbvre et al., 2002].* The left figure corresponds to the hash extracted from the Lena image while the right figure corresponds to that extracted from a rotated and scaled version. The x -axis gives the rotation degrees, while the y -axis represents the amplitude of the medium point.

image. This observation has led to approaches that can have very good performance while using simpler features with respect to the fingerprinting techniques based on several hashes previously presented. The duplicate detection technique proposed in this thesis, see chapter 4 and chapter 5, is actually based on a non-conventional distance function.

Fingerprinting based on the Radon transform

The image fingerprinting technique developed by Lefèbvre *et al.* is based on the Radon transform of the image [Lefèbvre *et al.*, 2002; Lefèbvre *et al.*, 2003]. The algorithm first consists in computing the Radon transform of an image [Deans, 1983]. A medium point, invariant to similarity transform of the image, is then computed for each angular projection. The hash is finally obtained by concatenating together those invariant points. Examples of hashes are given in figure 3.3. Moreover, the type of modifications applied to an image can be detected by comparing the original hash to that derived from the modified image. Two images are determined to be duplicates of each other by first computing the cross-correlation between their hashes, the position of the maximum is then used to synchronise the two hashes. The distance between two images is finally given by the mean square error (MSE) between the two hashes. With respect to the classification used in this thesis, this approach corresponds to a fingerprinting technique based on non-conventional distance function since it requires the computation of a cross-correlation function between the summaries.

The paper [Lefèbvre *et al.*, 2003] also presents some interesting results on collision and detection robustness. A collection of 40 images taken from the USC-SIPI database* is used. Each image is then modified according to the following eight transformations: 3×3 Gaussian filtering, 3×3 averaging filtering, JPEG compression with a quality of 25% and 15%, scaling with a factor of 0.8 and 1.2, and rotating by 1° and 2° . Then, the distance between the original and each duplicate is computed, resulting in a total of 320 values. It results that 312 out of 320 distances are below 10^{-3} . Additionally, the distance between each of the 780 possible original image pairs are also computed. It results that all 780 distances are above 10^{-3} . This corresponds to a recall of $0.975 = 312/320$ and a precision of $1 = 1 - 0/780$. However, the size of the test set is too small to draw any definite conclusion. Additionally, the range of transformations that can be detected seems quite poor.

*see <http://sipi.usc.edu/database/> for more information.

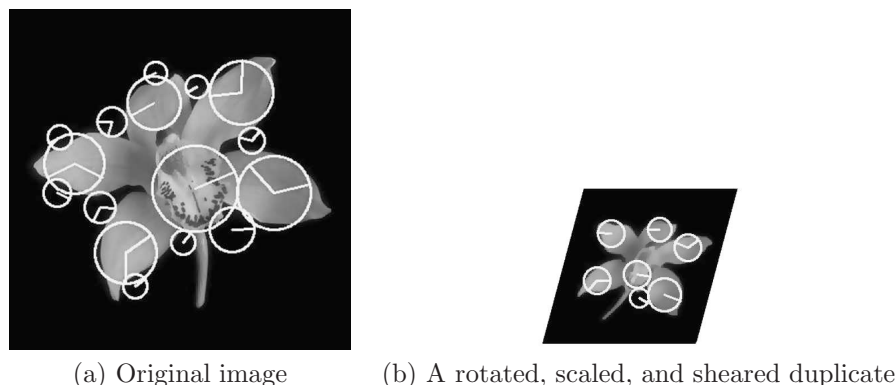


Figure 3.4: *Example of key points (KPs) in a pair of duplicate images [Ke et al., 2004].* The KPs are shown as white circles with embedded lines denoting dominant orientations and circle size denoting scale. Many of the KPs are found at the same relative positions. Note that the KPs corresponding to smaller scales are not represented (that is, most KPs are absent).

Fingerprinting based on key points

Ke *et al.* propose a fingerprinting method based on the extraction of features, referred to as key points (KPs), which are stable in a scale-space representation [Ke *et al.*, 2004]. An image is typically represented by thousands of KPs. Test images are then classified as duplicates or non-duplicates using local sensitive hashing to match their KPs to those of the original image. More specifically, no distance is directly computed but it is rather the number of matching KPs that quantifies if two images are duplicates of each other. With respect to the classification used in this thesis, this approach corresponds to a fingerprinting technique based on multiple hashes since an image is represented by thousands of local summaries.

This fingerprinting technique is mainly based on the robustness of the key points, which are popular local descriptors presented in [Lowe, 2004]. The KPs detector consists in four main steps, namely, scale-space maxima detection, KPs localisations, and orientations assignment. The scale-space maxima detection is efficiently implemented by constructing a Gaussian pyramid. The pyramid is subsequently used to detect the local maxima (termed KPs) in a sequence of difference-of-Gaussian images. In the second step, the KPs localisations are refined and the points that are found to be unstable are eliminated. In the third stage, the dominant orientation of each KP is determined as a function of the orientations found in its surrounding patch. Finally, the last step consists in describing a local patch orientations histogram, normalised in such a way that it is invariant to scale changes and affine transformations. Each image is then represented by thousands of KPs, and hence extra care has to be taken when indexing the KPs. An example of KPs localisations and dominant orientations is depicted in figure 3.4.

While this approach achieves very good performance, in terms of tradeoff between precision and recall, it requires a computationally complex features extraction step as well as many matching since each KP of an image has to be tested against all KPs of other images. A collection of 6261 images is used to test the system. 150 images are randomly selected from the collection and, for each image, 40 duplicates among twelve categories are generated. These categories are colourising,

contrast changes, cropping, despeckling, downsampling (no antialiasing filtering), flipping, colour depth reduction, outer frame addition, right-angle rotation, scaling (with antialiasing filtering), saturation and intensity changes. Section 4.3.1 gives more details about the used transformations. This results in a total of 12 111 images that are used to create a database where an average of 1100 KPs per image are extracted. Then, the 150 original images are used to query the database and the 40 most similar images are tallied to determine the number of false positives and consequently that of false negatives. The performance is finally synthesised in a single precision versus recall working point: a recall of 0.9985 corresponds to a precision of 1. An additional experiment is carried out, in which more difficult transformations are considered. This time, 10 duplicates are generated for each original image. They result from cropping the image by 50%, 70%, and 90%, shearing the image along the x -axis by 5° , 10° , and 15° , changing the intensity by 50% and 150%, and severe increase/decrease of the contrast. A total of 7611 images are thus used to create a database. In this additional experiment, a recall of 0.984 corresponds to a precision of 0.9986.

While the performance obtained by this method is the best for a content-based approach to date, the technique relies on a complex descriptor. More precisely, several seconds are needed on an actual computer to analyse an image and the description of a single image consists of thousands of 150-entry vectors. This means that, depending on the requirements of the duplicate detection system in terms of the number of tested images per second, the computational infrastructure can be very costly.

There exist other works based on feature points, for example those of [Lejsek *et al.*, 2006b; Monga and Evans, 2004]. Contrarily to the work of Ke *et al.*, [Monga and Evans, 2004] converts the set of feature points into a single binary hash. Additionally, the detection of the feature points is much simpler since it is based on the Harris' corner detector [Harris and Stephens, 1988]. On the other hand, the work of Lejsek *et al.* is very similar to that of Ke *et al.*. The only noticeable difference is that their own descriptor [Lejsek *et al.*, 2006a] is used instead of Lowe's KPs descriptor. The used descriptor is slightly more efficient than that of Lowe: less KPs are computed while still achieving better matching results.

Fingerprinting based on a mesh representation of the image

The fingerprinting method developed by Lu and Hsu is based on tiling the image with non-overlapping triangles and then generating a hash per triangle [Hsu and Lu, 2004; Lu and Hsu, 2005; Lu *et al.*, 2004]. The technique can be decomposed into two main steps. In the first step, the image is represented by a set of right-angled triangles. In the second step, each right-angled triangle is converted into a binary hash. These two steps are described in the next paragraph. With respect to the classification used in this thesis, this approach corresponds to a fingerprinting technique based on multiples hashes since an image is represented by as many local summaries as there are triangles in the mesh.

To create the set of right-angled triangles, the Harris corner detector [Harris and Stephens, 1988] is first applied to a downsampled version of the image. The justification behind using downsampling is twofold; firstly it avoids the detection of unstable corners, contained in the high-frequency band, and secondly it reduces the number of detected corners. Subsequently, the



Figure 3.5: *Example of image meshing [Lu and Hsu, 2005] for robust hashing.* Each triangle of the mesh is then warped to a right-angled triangle and gives to a hash.

Delaunay triangulation's algorithm [Lee and Schachter, 1980] is used to transform the set of corners into a triangular mesh; an example is depicted in figure 3.5. Each triangle is then normalised or, in other words, warped into a right-angled triangle of constant size. Now, each right-angled triangle is converted into a binary sequence of fixed-length as follows. The normalised triangle and its flipped version are superposed to create a 32×32 block. Then, the 2D discrete cosine transform (DCT) is applied to each 4×4 sub-block and the first AC coefficient is kept; this means that a triangle is represented by a total of 64 AC coefficients. The justification behind the selection of this particular coefficient is that higher-frequency coefficients are subject to noise and that the DC coefficient is not very discriminative. Then, the AC sequence is converted into a binary sequence by assigning a one to the 32 largest coefficient, and a zero to the 32 smallest coefficients. Finally, two images are duplicates of each other if the Hamming distance between, at least, N pairs of hashes is smaller than a certain threshold.

The performance of the method is interesting in terms of tradeoff between precision and recall. More precisely, a collection of 20 000 images and ten traditional images, such as Lena or Baboon, are used to create a database of original images. Then, the ten traditional images are modified according to the watermarking benchmark StirMark benchmark version 3.1, see section 4.3.1 for more information, and the original and the resulting 890 copies are used to query the database. The performance is finally synthesised in a precision versus recall table. For example, recalls of 0.82 and 0.945 correspond to precisions of 0.82 and 0.009, respectively. On an interesting side-note, the exact same method can be also used to generate authentication hashes.

Fingerprinting based on perceptual distance function

The fingerprinting technique developed by Qamra *et al.* is based on the computation of a perceptual distance function (DPF) [Li *et al.*, 2002; Qamra *et al.*, 2005]. Note that the abbreviation PDF is not used so as to avoid any confusion with Probability Density Function. More precisely, a DPF is generated for each pair of original and unknown image and measures the similarity between the two. The general idea of the approach is to activate different features for different image pairs. Hence, only the most similar features are taken into account when computing the distance. With respect to the classification used in this thesis, this approach corresponds to a fingerprinting

technique based on non-conventional distance function since it takes only the most similar entries of the summaries to compute the distance.

Let images be represented by p -dimensional feature vectors, and define the i -th distance Δd_i between two images as the absolute difference between the i -th feature. Then the basic perceptual distance function (DPF) is defined as the r -th root of the sum of the m smallest i -th distances $(\Delta d_i)^r$ where r and m are two parameters [Li *et al.*, 2002]. The justification behind the use of DPF is grounded in the science of cognitive psychology where it is shown that humans infer similarity between objects from their similarities rather than from their dissimilarities [Medin *et al.*, 1993; Tversky, 1977]. The number m of features used to compute the distance is selected as the one that achieves the best, on average, result on a training set. The proposed DPF achieves interesting results but is limited by the fact that m is fixed. Indeed, the similarities of different pairs of objects may depend on a different number of features. To overcome this restriction Qamra *et al.* propose three complementary methods for adaptively selecting m [Qamra *et al.*, 2005]. The first method, called thresholding, selects all i -th distances below a fixed threshold. The second method samples the DPF according to different values of m and averages them. The third method adds a weight to each feature; the weight is set as the inverse of the feature's standard deviation among similar images. Note that these three methods are complementary and can be used together.

The performance of the method is interesting in terms of tradeoff between precision and recall. A collection of 20 000 images is used to test the system. Among them, 500 images are randomly selected and modified according to 40 duplicates, the same transformations than for Ke *et al.*'s work are used, among twelve categories. These categories are colourising, contrast changes, cropping, despeckling, downsampling (no antialiasing filtering), flipping, colour depth reduction, outer frame addition, right-angle rotation, scaling (with antialiasing filtering), saturation and intensity changes. Section 4.3.1 gives more details about the used transformations. A total of 40 000 images are thus indexed in a database. Subsequently, the 500 seed images are used to query the database, and the 40 most similar images are tallied to determine the number of false positives and consequently that of false negatives. The performance is finally synthesised in a precision versus recall curve. For example, recalls of 0.9 and 0.8 correspond to precisions of 0.67 and 0.93, respectively.

The cognitive psychology explanation is interesting but a more mundane reason, not cited by the authors, exists for the algorithm's good performance. Indeed, although some features are robust against certain types of image transformations, they can vary drastically for other transformations. Subsequently, by using a distance function that takes into account only the most similar features, one insures that the most robust features are always used. While it results in good general performances, it also signifies that the recall rates fall very quickly for high precision rates. Indeed, this is symptomatic of the metric used that implies that the system is unable to distinguish between similar yet unrelated images.

Fingerprinting based on image thumbnails

The fingerprinting technique developed by Wang *et al.* is based on thumbnail versions of the image [Wang *et al.*, 2006]. More specifically, images are divided into n by n blocks and the average intensity within each block is computed. Different values of n are used to represent the image with

more details, and the resulting thumbnails are concatenated into a single vector. The dimension of the vector is then reduced by making use of principal component analysis [Jackson, 1991] and selecting the K largest eigenvalues. Finally, a binary string is produced by assigning an one to entries larger than the vector's average and a zero otherwise. Two images are duplicates if the most significant bits of their hashes are equal and if there is less than a certain number of bits that differ for the least significant bits. With respect to the classification used in this thesis, this approach corresponds to a fingerprinting technique based on non-conventional distance function since two distances are actually computed.

The performance of the method is difficult to assess since it is based on a peculiar metric and considers only a limited number of transformations. For instance, this study consider only the following transformations, scaling, colour to greyscale conversion, and compression. In the experiment, images are selected from the Internet. They contain the answer to four queries: 'Angelina Jolie', 'Anime', 'Britney Spears', and 'Cartoon'. The first images returned by a query are then grouped in a so-called scope. Then, each pair of images within a scope is labelled whether they are duplicates of each other, according to the above transformations, or unrelated. Finally, conventional precision and recall metrics are applied on the pair of images and summarised within a table for different scope's sizes. It results in a precision around 0.35 and a recall around 0.96 for a scope's size of 100. However, the performance degrades as the group size increase, for instance, a scope size of 1000 corresponds to a precision of 0.28 and a recall of 0.93. Additionally, images that are connected through a chain of pairwise duplicate relationship are grouped into a single group. Grouping achieves a recall of 0.55 and a precision of 0.96 for a scope's size of 100.

3.3.2 Robust hashing techniques

Hashing relates to techniques that use a single summary of the image content — often called a digest or hash value. In duplicate detection based on robust hashing, the distance between digests is used to determine the corresponding images relationship. More precisely, two images are duplicates of each other if the distance between their hashes is smaller than a certain threshold. A typical distance is based on the L_1 -norm. For example, the L_1 distance between two binary strings, often called Hamming distance, gives the number of bits that differ while the normalised version scales this distance between zero, all bits are equal, and one, all bits are different.

The concept of hashing is very similar to that of cryptographic hash functions, which maps data strings to a small and constant number of bits. Cryptographic hash functions are often, and successfully, used to authenticate messages [Stinson, 2002]. They are not, however, directly usable for multimedia content because images can undergo quite severe modifications without altering their perceptual values. Indeed, cryptographic functions are designed so that the alteration of a single bit of the message results in a totally different hash. This deficiency has led several research teams to develop the notion of robust hashing.

Note also that robust hashing for duplicate detection is tightly linked to robust hashing for authentication. Indeed, many of the hashing methods presented thereafter have also a security component that aims at securing the produced hash through randomisation. This permits to use the hash in the same fashion than cryptographic hash functions. However, this feature is not really

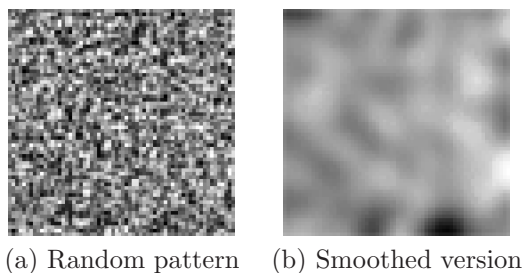


Figure 3.6: *Example of a random pattern and its smoothed version [Fridrich, 1999].* The left figure shows a 64×64 random pattern while the right figure depicts one of its smoothed version.

relevant to duplicate detection as studied in this thesis.

Hashing based on random projections

The hashing technique developed by Fridrich is based on projecting the image onto random patterns [Fridrich, 1999, 2000]. To achieve this, the author proposes two steps. In the first step, the image is projected on N randomly generated patterns with zero mean. In the second step, the projections' values are converted into a binary sequence.

To create the N random patterns, an initial pattern is first generated using a random generator, and the other patterns are obtained by filtering the initial pattern with different low-pass filters. An example of a random pattern, as well as one of its smoothed version is depicted in figure 3.6. Subsequently, the image is projected on each pattern. If the absolute value of the projection is above a certain threshold, a one is assigned to the pattern and a zero otherwise. The ones and zeros are finally concatenated together to form the hash. Note that the threshold is adaptively adjusted so as to obtain approximately an equal number of zeros and ones.

In [Fridrich, 2000] two approaches are proposed to make the aforementioned method robust against rotation and scaling. The first one uses the Fourier-Mellin transformation [Zwicke and Kiss, 1983]. On the other hand, the second approach is based on patterns with a circular symmetry that have their centres mapped to the centre of gravity of the image.

Some basic tests are performed in order to show the robustness of the scheme. The method seems to be quite robust, StirMark benchmark version 3.1, on the few images that are tested. However since the aim of this work is to generate a watermark correlated to the image content, no study has been made regarding its discriminative power.

Hashing based on random image tiling

The hashing technique developed by Venkatesan *et al.* is based on a random rectangular tiling of the image [Venkatesan and Jakubowski, 2000; Venkatesan *et al.*, 2000]. In the initial work of [Venkatesan and Jakubowski, 2000], a tiling framework, based upon four steps, is presented. In the first step, the image is divided into possibly overlapping regions. In the second step, each region is summarised with a value. In the third step, each value is randomly rounded to either the nearest larger or the nearest smaller integers. In the fourth step, the values are aggregated into

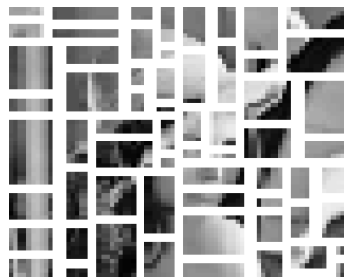


Figure 3.7: *Example of a random rectangle tiling on the coarsest wavelet sub-band [Venkatesan et al., 2000]. 64 rectangles are used on the Lena image.*

an intermediate hash, and finally an error correcting code is used to compress the hash. In other words, the intermediate hash is considered as a noise contaminated code.

An actual tiling technique is presented in [Venkatesan *et al.*, 2000]. More precisely, the image is first transformed in the wavelet domain, then each sub-band is decomposed into non-overlapping rectangles, for an example of decomposition see figure 3.7, and each rectangle is summarised with a statistic, namely mean for the coarse sub-band and variance for the others sub-bands. The statistics are then randomly rounded to form 3-bit values. The aggregated string of 3-bit values is finally decoded using a Reed-Muller error-correcting decoder. Two images are duplicates of each other if the normalised Hamming distance between their hashes is lower than a certain threshold.

To test the algorithm, a database of 100 images, containing among other images Lena and Baboon, is used. Duplicates are obtained by applying the StirMark benchmark version 3.1 benchmark. No quantitative results are given but it is stated that the scheme is robust, in other words the Hamming distance is close to zero, for the following transformations: rotations up to 2° , cropping up to 10% of image area, scaling by up to 10%, random deletion of up to 5 lines, shifting by up to 5%, JPEG compression using quality factor as low as 10%, 4×4 median filtering. Then, the probability of collision is tested by comparing Baboon's hash to the 99 remaining hashes. It is found that the corresponding Hamming distances range between 0.35 and 0.55. this signifies that for the transformations mentioned above, the recall is near 1 and the precision is 1. However, the considered transformations are mild, and hence it does not give any indication of the method's performance on the complete StirMark benchmark version 3.1 benchmark.

In [Mihçak and Venkatesan, 2001], an iterative region growing on each tile replaces the wavelet-based description of the tiles. The iterative region growing aims at producing a binary low-resolution version of the image where only geometrically strong components are present. To achieve this, a iterative median filter is used to produce an image where each pixel corresponds to the median value of a given rectangular region centred on the pixel. Note that the size of the rectangle depends on the geometry of the surrounding region. The results seem to indicate that this scheme is more robust than the wavelet-based description.

Additional work exists on the topic of intermediate hash compression, or in other words reducing the size of the hash while adding robustness. For example, Johnson and Ramchandran use the distributed coding paradigm to compress the intermediate hash by making use of a Wyner-Ziv encoder [Johnson and Ramchandran, 2003]. Additionally, Monga *et al.* aims at ensuring that

perceptually identical images are compressed to the same hash [Monga *et al.*, 2004, 2006]. To reach this goal, the authors propose to cluster the space of intermediate hashes into perceptually close regions. They first built a cost function in an arbitrary metric space such that its minimisation yields the ideal clustering. Additionally, they show that this ideal clustering problem is an *NP*-complete problem and propose two heuristic approaches to approximate the minimisation. Monga *et al.* give experimental results and compare them to those obtained for [Venkatesan *et al.*, 2000]. Their scheme is more flexible and obtains better performance than simple error-correcting compression, but at the price of a higher complexity.

Hashing based on the discrete cosine transform

The image hashing technique developed by Kim is based on the DCT of a low resolution version of the image [Kim, 2003]. The first step of the approach consists in computing a 8×8 version of the image. To achieve this, the image is subdivided into 64 non-overlapping and equal-sized blocks, and the average intensity of the pixels within each block is computed. The justification of this resizing relies on providing an invariance of the fingerprint to local changes. In the second step, the 2D DCT is computed on the 8×8 image. The AC coefficient of the DCT are then ranked according to their magnitudes. The result is a hash given by a permutation of the first 63 integers. Finally, two images are declared similar when the distance between their corresponding fingerprints (given by the L_1 -norm) is below a certain threshold.

The performance of the method is unsurprisingly good for local transformations but quite poor for geometric transformations. A collection of 40 000 images is used to create a database. Additionally, 5 images and their 11 duplicates are added to this database and thus results in a total of 40 055 images. The transformations used to create the duplicates are different for each original but are mainly of non-geometric nature. For additional information on the used transformations, the reader is referred to [Kim, 2003]. The system performance is then evaluated by querying the created database with each of the original images. The performance is finally synthesised in a precision versus recall curve. For example, recalls of 0.92 and 1 correspond to precisions of 0.86 and 0.06, respectively.

Hashing based on the Radon transform

The robust image hashing technique developed by Seo *et al.* is based on the Radon transform of the image [Seo *et al.*, 2003, 2004]. The idea behind the approach is as follows. In a first step, the Radon transform [Deans, 1983] is modified so as to make it invariant to affine transformations of the image. In a second step, the affine invariant transformation is converted into a binary hash. These two steps are described in the next paragraph.

The auto-correlation makes the Radon transform invariant to translation while the log-mapping and the Fourier transform bring the scale and rotation invariance. In the second step, a 20×20 binary fingerprint is computed. To achieve this, the 21×21 low-frequency coefficients of the Fourier transform are selected. The justification of this choice relies on the tradeoff existing between the robustness and the discriminatory power of the chosen feature. Indeed, practice has shown that, in general, low-frequency features are more robust while high-frequency features are more

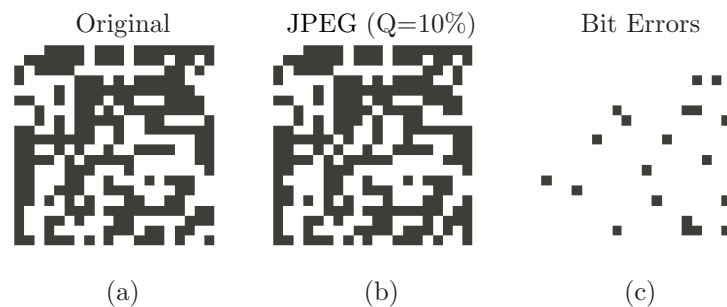


Figure 3.8: *Example of fingerprints* [Seo et al., 2004]. (a) Fingerprint of the original Lena image, (b) Fingerprint of the compressed Lena Image with a 10% quality factor, and (c) the difference between a and b showing the error in black.

discriminatory. Then, a 2×2 two-dimensional filter, designed to detect sign changes, is applied to the selected coefficients and the result is converted to a binary hash. The justification of the binary conversion is again empirical as it is indeed experimentally verified that the difference between affine invariant features is very robust against many kind of transformations. Note that two intermediate binary hashes are obtained, one for the amplitude of the Fourier transform and the other one for the phase. Finally, the two hashes are merged using a bitwise exclusive or function. The justification behind this merging is that it is experimentally verified to improve the pairwise independence, thus lessening the risk of collisions.

Two images are duplicates of each other if the Hamming distance between their hashes is below a certain threshold. For example, figure 3.8a shows an example of the hash extracted from the Lena image. Additionally, figure 3.8b shows the hash extracted from a compressed version of the Lena image while figure 3.8c illustrates the difference between the two hashes, corresponding to a Hamming distance of 0.05. In this case, these two images are duplicates of each other only if the threshold is larger than 0.05.

The performance of the method is globally interesting but relatively poor in terms of the tradeoff between precision and recall. A collection of 1000 images is used to test the system. A database of original images is then constructed using the robust hashing method. Then, the following eleven transformations are used to create the test images: JPEG compression (quality 10%), Gaussian filtering, sharpening filtering, 4×4 median filtering, 45° and 90° rotations, 0.5 and 0.15 scalings, 2% cropping, 17 columns and 4 rows removal, random bending. The transformations are used to estimate the system's recall rate. On the other hand, the system's precision is estimated by inputting the originals themselves and then logging how many originals are returned. Note that an ideal system should return only a single original. In [Seo et al., 2004], two working points are reported for two different values of the threshold: 0, or no error, and 10/200. In the first case, 878 out of 11 000 test images are not correctly detected, which corresponds to a recall of $0.92 = 1 - 878/11000$. On the other hand, an average of 2.396 original images are returned per query, which corresponds to a precision of $0.38 = (1000 \times 0.92)/(1000 \times 2.396)$. In the second case, 20 test images are not correctly detected, which corresponds to a recall of $0.998 = 1 - 20/11000$. On the other hand, an average of 51.5 original images are returned per query, which corresponds to a precision of $0.02 = 1000 \times 0.998/(1000 \times 51.5)$.

Note that considering only pairwise relationship is, in the author's opinion, limitative and sometime misleading. Indeed, two real duplicates that are quite different, and result in very different hashes, will never be considered to be duplicates of each other unless a chain of duplicate images links them. In other words, this approach seems to work well when many duplicates of the same image are considered at the same time.

3.3.3 Standardisation efforts

The moving picture experts group (MPEG) is a working group of ISO/IEC charged with the development of video and audio encoding standards. One of the resulting standards, MPEG-7, is a formal system for describing multimedia content. MPEG members are currently studying the feasibility of incorporating a visual descriptor into the standard MPEG-7 that is specifically designed to serve as visual identifier. In other words, they are proposing a standardised feature that should performs well for the duplicate detection task [MPEG12816, 2006; MPEG13152, 2006; MPEG13579, 2006]. This duplicate detection task is taken very seriously within MPEG since test conditions [MPEG12841, 2006] and image management database tools [MPEG13861, 2006] are being modified so as to accommodate this new search task.

The proposed features is based on feature points and is very similar to [Ke *et al.*, 2004; Lejsek *et al.*, 2006b; Monga and Evans, 2004]. Actually, this descriptor is a simplified version, for complexity reasons, of Lowe's detector [Lowe, 2004]. For instance, the localisation of feature points is based on the Harris corner detector [Harris and Stephens, 1988], and the description of the feature points is based on the local gradient histogram but for a region with a fixed size rather than a size that depends on the region's content. This is ongoing work, and the visual identifier that is going to be standardised will certainly improve.

In the MPEG's study, duplicates are generated according to the following transformations: brightness changes, aspect ratio changes, colour to grey-level conversion, JPEG compression, colour-depth reduction, cropping, histogram equalisation, blurring, rotation, scaling, translation, and flipping. Each transformation is then parameterised according to three severity level: heavy, medium, and light modifications. Currently, the proposed descriptor is robust against most of these transformations except rotation, scaling and flipping. In any case, it performs better, for the duplicate detection task, than the edge detector already present in the MPEG-7 standard but is more complex.

3.4 Chapter summary

In this chapter, the state of the art for image duplicate detection is presented. We first distinguish between two philosophies, namely watermarking and content-based, and describe the advantages and drawbacks of each of them. Basically, watermarking is less flexible than content-based method because it requires modifying the image for incorporating the signature. More precisely, the embedding's requirement entails that watermarking is adapted only if one has total control over the original artwork and means that a watermarked image can be detected only as long as a mean

Table 3.1: *Synthesis of state of the art duplicate detection methods.* This table synthesises some state of the art methods. The marks +, ++, +++ refer to, respectively, so-so, good and excellent while – denotes a drawback. The appearance order of the methods are the same as in section 3.3.

method	type	memory	complexity	performance
[Lefebvre <i>et al.</i> , 2003]	fingerprinting	+	+	–
[Ke <i>et al.</i> , 2004]	fingerprinting	–	–	+++
[Hsu and Lu, 2004]	fingerprinting	+	+	++
[Qamra <i>et al.</i> , 2005]	fingerprinting	+	++	++
[Wang <i>et al.</i> , 2006]	fingerprinting	+	++	–
[Fridrich, 2000]	hashing	+++	+	+
[Venkatesan <i>et al.</i> , 2000]	hashing	++	+	+
[Kim, 2003]	hashing	++	+++	+
[Seo <i>et al.</i> , 2004]	hashing	+++	+	+

to remove the signature is not discovered. On the other hand, content-based duplicate detection is more flexible but not yet as mature as watermarking in terms of precision and recall rates.

We then presented several existing content-based techniques. These methods are classified into two sub-categories, namely robust hashing and fingerprinting. Robust hashing approach consists in summarising the image with a digest, often binary, and then use a simple L_1 distance to determine if two images are duplicates of each other or unrelated. On the other hand, fingerprinting refers to method that cannot be classified as robust hashing, according to the previous definition. It turns out that most content-based duplicate detection techniques are of the robust hashing type. However, they often rely on simpler features than fingerprinting techniques and, consequently, do not perform as well. Still, the produced hash can be easily used to index images while this is not always the case with fingerprinting techniques.

It is also noted that content-based duplicate detection is still a recent field of signal processing. As such, there is not yet standardised methods to assess the performance of content-based duplicate detection techniques. Additionally, there is still a lot of different opinions on the exact definition of the problem or even on what a duplicate is. Still, it is an active research domain that is quickly growing. For instance, it is the object of standardisation proposal within the MPEG-7 framework, at least for features that could be specifically used for duplicate detection. The road is still long as many related problems have to be first solved, such as defining a standard way of testing duplicate detection system.

We now synthesise the existing content-based duplicate detection techniques. The synthesis can be found in table 3.1, where the different methods are synthesised in terms of memory, complexity and performance. Memory refers to the amount of memory required to store the description of an original or other information necessary to detect its duplicates. Complexity refers to the computational complexity necessary to describe the test image and to compare the description with that of an original image. In other words, complexity does not relate to the time necessary to train the system. Performance concerns the tradeoff between falsely detected unrelated images and falsely rejected duplicates. Finally, note that the marks given are the qualitative appreciations of the author obtained through the available published information.

Part II

Dissertation

Peut-être cet ouvrage est-il trop long: toute plaisanterie doit être courte, et même le sérieux devrait bien être court aussi. Lettres Philosophiques, Voltaire (1694 — 1778)

A Framework for Content-based Image Duplicate Detection

4

In the first part of this chapter, we define a generic framework for content-based duplicate detection systems. The proposed framework begins with a simple mathematical model of the duplicates of an image. The model defines the subspace spanned by the duplicates of an original and is presented in section 4.1. This model permits to gain some understanding on how duplicates are organised, and gives useful indications of how to design an efficient duplicate detection system. The second element of the framework is a generic system to detect the duplicates of original images; it is accounted for in section 4.2. Two cases are analysed. The first case concerns a system that detects the duplicates of a single image while the second case deals with a more general system that simultaneously detects the duplicates of several images. The next element of the framework is the assessment methodology, reported in section 4.3, in which we detail the test images and the error metrics used to assess the performance of the proposed duplicate detection system.

In the second part of this chapter, we give an overview of the proposed duplicate detection system in section 4.4. Additionally, this section also presents common components of the proposed duplicate detection system. More precisely, it includes the preprocessing operations applied to the image, and the subsequent low-level visual features extraction procedure, used in chapter 5 and chapter 6.

4.1 Model of the duplicates of an image

The general idea behind the approach proposed in this thesis is to estimate the region of the image space in which the duplicates of a particular image lie. Duplicates can then be easily detected by asserting whether a test image lies inside or outside this particular region. For example, the region determined by all the resized versions of an image by a resizing factor going from 0.1 to 5 is, under certain assumptions, a continuous smooth curve embedded within the image space. The curve

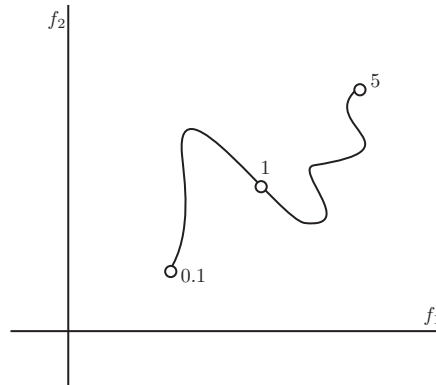


Figure 4.1: *Curve defined by an image and its duplicates.* The figure shows a two-dimensional feature space that exemplifies the duplicates obtained from the modification of an original image by a smooth transformation, for example resizing by a factor going from 0.1 to 5, in this case a resizing factor of one corresponds to the original image.

starts at factor 0.1, goes through the original for the resizing factor equals to one, and finishes at factor 5. Consequently, transformations of the original image by other operations result in as many curves going through the original, and the duplicates of a particular image lie in the region defined by the union of these curves.

It is relatively difficult to imagine a curve embedded within the image space since this space has a large number of dimensions. On the other hand, a relatively limited number of visual features can be used to describe an image. In this case, the curves embedded within the image space, a very large number of dimensions, are mapped into curves embedded within the feature space, relatively low number of dimensions. This idea is illustrated in figure 4.1 for two visual features and for the resizing operation. In the following, the model of the duplicates of an image is given for the image space but can be easily extended to the feature space. We next formalise this idea and extend it to duplicates of the duplicates.

Let us assume that images are smooth bi-dimensional functions and consequently that the image space is the space of the smooth functions. We further assume that the considered image transformations are smooth functionals defined on this space. Then, the set containing the duplicates of the original image \mathbf{I} can be defined as follows

$$\mathcal{D}(\mathbf{I}) = \{f_i(\mathbf{I}, p) : p \in \mathcal{C}_i, i = 1, 2, \dots, N\}, \quad (4.1)$$

where the $f_i(\mathbf{I}, p)$ are the N considered transformation functionals, p stands for each functional's parameter, and the set \mathcal{C}_i give the correspond parameterisations or, in other words, the possible values of the parameters. To give a better intuitive feeling, lets consider that $f_1(\mathbf{I}, p)$ corresponds to the resizing transformation functional or operation. In this case, p stands for the scaling factor and \mathcal{C}_1 corresponds to the range of allowed scaling factors; for instance \mathcal{C}_1 is given by the interval $[0.1, 5]$ for the example presented previously. The original image \mathbf{I} is implicitly part of $\mathcal{D}(\mathbf{I})$ because we assume that for any transformation $f(\cdot, \cdot)$ there exists an invariant parameterisation p such that $\mathbf{I} = f(\mathbf{I}, p)$. For example, in the case of the resizing operation it implies that the corresponding

parameterisation, \mathcal{C}_1 , contains the real number one, which creates a duplicate image equals to the original image.

Additionally, duplicates of the duplicates can be considered in turn to be duplicates of the original, in which case, other curves going through each duplicate are also included. The duplicate set $\mathcal{E}_n(\mathbf{I})$ for up to n -level of compositions can be recursively defined by

$$\mathcal{E}_n(\mathbf{I}) = \{\mathcal{D}(\mathbf{J}) : \mathbf{J} \in \mathcal{E}_{n-1}(\mathbf{I})\}, \quad (4.2)$$

$$\mathcal{E}_1(\mathbf{I}) = \mathcal{D}(\mathbf{I}). \quad (4.3)$$

Note that $\mathcal{E}_n(\mathbf{I}) \supseteq \mathcal{E}_{n-1}(\mathbf{I}) \supseteq \dots \supseteq \mathcal{E}_1(\mathbf{I})$ because of the existence of the invariant parameterisation.

The set $\mathcal{E}_n(\mathbf{I})$ is a complex object to apprehend. To analyse $\mathcal{E}_n(\mathbf{I})$, we first introduce a simplification. Indeed, let us now consider that duplicates resulting from n -level of composition can be expressed by a single functional $g_n(\mathbf{I}, \mathbf{p})$. The first variable \mathbf{I} is the original image, and the other variable \mathbf{p} is a vector of parameters that controls the duplicate aspect, for example \mathbf{p}_1 can be the scaling factor and \mathbf{p}_2 the rotation angle. Such a functional can be recursively constructed by using the previously introduced transformation functionals $f_i(\mathbf{I}, p)$. More precisely

$$g_n(\mathbf{I}, \mathbf{p}) = f_n(g_{n-1}(\mathbf{I}, \mathbf{p}), \mathbf{p}_n), \quad (4.4)$$

$$g_1(\mathbf{I}, \mathbf{p}) = f_1(\mathbf{I}, 1). \quad (4.5)$$

Note that the order of operations can be modified by permuting the indices i of the transformations $f_i(\mathbf{I}, p)$. In this simplified case, the set of duplicates for n -level of composition is given by

$$\mathcal{F}(\mathbf{I}, n) = \{g_n(\mathbf{I}, \mathbf{p}) : \mathbf{p} \in \mathcal{C}_1 \times \mathcal{C}_2 \times \dots \times \mathcal{C}_n\}. \quad (4.6)$$

The duplicate set $\mathcal{F}(\mathbf{I}, n)$ is thus defined by a bounded smooth high-dimensional surface, or smooth manifold, embedded within the image space. The manifold intrinsic dimensionality is upper bounded by n , the number of considered compositions, since the manifold is created by a function controlled by $n + 1$ parameters and one of them is the original image.

Additionally, it is possible to link the manifold $\mathcal{F}(\mathbf{I}, n)$ with the more complex object $\mathcal{E}_n(\mathbf{I})$ defined above. Indeed, lets now consider that the number of compositions n is equal to the number of transformations N . Since different sets $\mathcal{F}(\mathbf{I}, n)$ can be constructed for different orders of operations, we get

$$\mathcal{E}_N(\mathbf{I}) = \bigcup_{\nu} \mathcal{F}(\mathbf{I}, \nu), \quad (4.7)$$

where ν is a permutation of the first N positive integers, \bigcup_{ν} signifies the union on all possible permutations, and $\mathcal{F}(\mathbf{I}, \nu)$ stands for the duplicate set as defined in equation (4.6) but with the order of operations modified according to the permutation ν . This result implies that $\mathcal{E}_N(\mathbf{I})$ is given by the union of the $N!$ smooth manifolds.

We now analyse the effect of varying the order of operations on the duplicate set $\mathcal{E}_N(\mathbf{I})$. To achieve this, we assume that varying the order of operations changes the resulting duplicate but

not drastically. Let $d(\cdot, \cdot)$ be a distance function on the image space, and define

$$\xi(\mathbf{I}) = \max_{\mathbf{p}} \max_{\nu_1, \nu_2} d(g_N(\mathbf{I}, \mathbf{p}, \nu_1), g_N(\mathbf{I}, \mathbf{p}, \nu_2)) \quad (4.8)$$

where $\nu_{1,2}$ are permutations of the first N positive integers, and $g_N(\mathbf{I}, \mathbf{p}, \nu)$ stands for the single transformation functional defined above but with the order of operations modified according to the permutation ν . Then the maximisation on ν_1 and ν_2 gives the maximal possible distance between duplicates for a given parameterisation \mathbf{p} . Finally, the value of $\xi(\mathbf{I})$ gives the largest such distance among all possible parameterisations. Now, let us define the high-dimensional volume $\mathcal{V}(\mathbf{I}, \nu)$ based on the manifold generated by an arbitrary order of operations ν

$$\mathcal{V}(\mathbf{I}, \nu) = \{\mathbf{J} : d(\mathbf{J}, \mathbf{K}) \leq \xi(\mathbf{I}), \mathbf{K} \in \mathcal{F}(\mathbf{I}, \nu)\}. \quad (4.9)$$

For any order of operations, defined by the permutation μ , we then have $\mathcal{V}(\mathbf{I}, \nu) \supset \mathcal{F}(\mathbf{I}, \mu)$. This result implies that the duplicates of an image can be enclosed within a high-dimensional volume that has a thickness $2\xi(\mathbf{I})$. The thickness is proportional to the influence of the order of operations. For instance, if varying the order of operations does not change the resulting duplicate then the duplicates of an image lie on a smooth manifold whose intrinsic number of dimensions is upper bounded by N , the number of considered transformations.

In the real world many assumptions do not hold. For instance, images are not smooth signals but rather sampled and spatially bounded signals. Additionally, transformations might not be smooth; typical examples of non-smooth transformations are joint picture experts group (JPEG) compression or cropping. Nevertheless, the model developed in this section remains useful as it gives clues as to how to develop an efficient duplicates detection system. When referring to this model in the following, we often call it the subspace spanned by the duplicates of an original.

4.2 Generic duplicate detection system

In this section, a generic duplicate detection system is proposed. The generic system consists in a system that simultaneously detects the duplicates of multiple original images. However, a simplified version of the general system, where only a single original image is considered, is first presented in section 4.2.1. The multiple original duplicate detection system is then presented in section 4.2.2. Not only is the duplicate detection system proposed in this thesis based on the general system, but also the simplified version is one of the key components of the proposed system. Finally, the simplified and general systems serve as basis to develop adequate methods to evaluate their performance, as done in section 4.3.

4.2.1 Generic duplicate detection — single original image system

The detection of the duplicates of a single original can be modelled as follows. We consider a system tuned to the detection of the duplicates of a specific original image \mathbf{O} . This duplicate detection system can be viewed as a binary classifier that maps the test image \mathbf{T} into one of two classes. More precisely, the label +1 corresponds to the class “the test image \mathbf{T} is a duplicate of

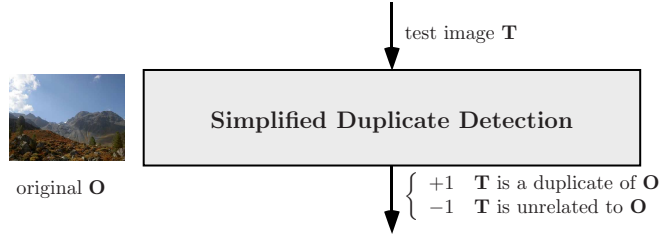


Figure 4.2: *Simplified duplicates detection system (single original).*

the original image \mathbf{O} ” while the label -1 stands for the class “ \mathbf{T} is unrelated to \mathbf{O} .” Such a system can be summarised to a binary function $d_{\mathbf{O}}^1(\cdot, u) \in \{-1, +1\}$ where u is a parameter controlling the system’s selectiveness. More precisely, $d_{\mathbf{O}}^1(\mathbf{T}, u)$ is equal to $+1$ if the test image \mathbf{T} is estimated to be a duplicate of \mathbf{O} and to -1 if \mathbf{T} and \mathbf{O} are considered unrelated. The detection of the duplicates of a single original is illustrated in figure 4.2.

The system’s mechanics can be defined as follows. The main idea is to estimate the probability $\Pr\{\mathbf{T} \sim \mathbf{O}\}$ that a test image \mathbf{T} is a duplicate of the original image \mathbf{O} . A decision can then be obtained by comparing the estimated probability $\Pr\{\mathbf{T} \sim \mathbf{O}\}$ to a fixed threshold $u \in [0, 1]$. If the probability is larger than the threshold then the test image is considered to be a duplicate of the original image; otherwise both images are regarded as unrelated. Finally, $d_{\mathbf{O}}^1(\cdot, u)$ is formally given by

$$d_{\mathbf{O}}^1(\mathbf{T}, u) = 2 \cdot \left(\mathbb{I}_{\{x: x > u\}}(\Pr\{\mathbf{T} \sim \mathbf{O}\}) - \frac{1}{2} \right) \quad (4.10)$$

where $u \in [0, 1]$ is a threshold, \mathbf{O} is the original image, \mathbf{T} is the test image, $\Pr\{\mathbf{T} \sim \mathbf{O}\}$ is the estimated probability that \mathbf{T} is a duplicate of \mathbf{O} , and $\mathbb{I}_{\mathcal{A}}(x)$ is the indicator function. Recall that $\mathbb{I}_{\mathcal{A}}(x)$ is equal to one if $x \in \mathcal{A}$ and to zero otherwise.

4.2.2 Generic duplicate detection — multiple original image system

The detection of duplicates of multiple original images can be modelled as follows. We now consider a system tuned to the simultaneous detection of the duplicates of any original among a set of specific original images \mathcal{O} . Each element \cdot of \mathcal{O} , thereafter denoted $\mathcal{O}(\cdot)$, corresponds to a specific original image. This duplicate detection system can be viewed as a multi-class classifier that maps the test image \mathbf{T} into one of $N + 1$ classes, where N is equal to $|\mathcal{O}|$, the number of original images. More precisely, N classes, labelled $+1, +2, \dots, +|\mathcal{O}|$, correspond to the case “the test image \mathbf{T} is a duplicate of the corresponding original images.” On the other hand, the remaining class, labelled -1 , stands for the case “ \mathbf{T} is unrelated to any image among \mathcal{O} .” Such a system boils down to the integer-valued function $d_{\mathcal{O}}^N(\cdot, u) \in \{-1, +1, \dots, +|\mathcal{O}|\}$ where u is a parameter controlling the system’s selectiveness. More precisely, $d_{\mathcal{O}}^N(\mathbf{T}, u)$ is equal to a positive integer i if the test image \mathbf{T} is estimated to be a duplicate of the original image $\mathcal{O}(i)$ and to -1 if \mathbf{T} is considered unrelated to any of the images contained in \mathcal{O} . The detection of the duplicates of multiple originals is illustrated in figure 4.3.

The system’s mechanics can be defined as follows. The main idea is to estimate the set

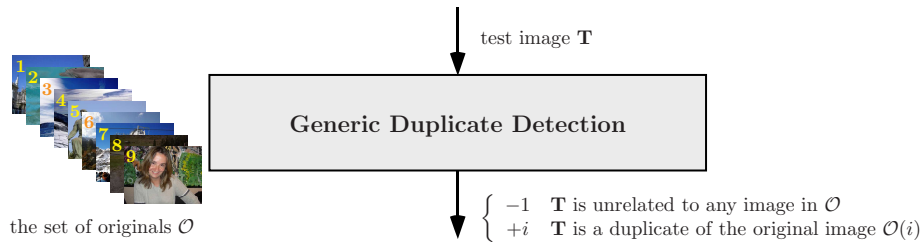


Figure 4.3: *Generic duplicates detection system (multiple original images).*

of probabilities $\{\Pr\{\mathbf{T} \sim \mathcal{O}(i)\}\}_{i=1}^N$ where each element $\Pr\{\mathbf{T} \sim \mathcal{O}(i)\}$ is an estimation of the probability that \mathbf{T} is a duplicate of the corresponding original image $\mathcal{O}(i)$. In other words, the probabilities can be estimated using the single original duplicate detectors presented in the previous section. A decision can then be obtained by comparing the largest probability contained in the aforementioned set to a fixed threshold u . If the probability is larger than the threshold then the test image \mathbf{T} is considered a duplicate of the corresponding original image while otherwise \mathbf{T} is regarded as unrelated to any of the original images. Finally, $d_{\mathcal{O}}^N(\cdot, u)$ is formally given by

$$m = \arg \max_{i=1, \dots, N} p_{\mathbf{T} \sim \mathcal{O}(i)}, \quad (4.11)$$

$$d_{\mathcal{O}}^N(\mathbf{T}, u) = m \cdot I_{\{x: x > u\}}(\Pr\{\mathbf{T} \sim \mathcal{O}(m)\}) - I_{\{x: x \leq u\}}(\Pr\{\mathbf{T} \sim \mathcal{O}(m)\}), \quad (4.12)$$

where $u \in [0, 1]$ is a threshold, \mathcal{O} is the set of original images, \mathbf{T} is the test image, $\Pr\{\mathbf{T} \sim \mathcal{O}(i)\}$ are the estimated probability that \mathbf{T} is a duplicate of the corresponding original image $\mathcal{O}(i)$, and $I_{\mathcal{A}}(x)$ is the indicator function. If the largest estimated probability is smaller than the threshold, the first term in the right hand side of equation (4.12) is then equal to zero and the second term is equal to minus one. On the other hand, if the largest estimated probability is larger than the threshold, the second term in the right hand side of equation (4.12) is then equal to zero and the first term is equal to the label of the estimated original of the test image.

4.3 Performance evaluation methods

In this section, we present the images and the metrics used to assess the performance of the duplicate detection algorithms proposed in the following chapters. The test images are introduced in section 4.3.1. Two metrics are then defined in section 4.3.2. One metric is used to assess the performance of the single original image duplicate detection system while the other one is used for the multiple original images duplicate detection system.

4.3.1 Test images

To assess the performance of the proposed systems, the same image collections as in [Ke *et al.*, 2004] are used. The first collection contains 18785 photographs including, but not limited to, landscapes, animals, constructions, and people. The image sizes and aspect ratios are variable, for example 900×600 , 678×435 , or 640×480 pixels. They are mostly colour images, except

Table 4.1: *Duplicate images test set Qamra*. This test set contains the duplicates proposed in [Qamra *et al.*, 2005] and used in [Ke *et al.*, 2004] as well. It simulates transformations often encountered when publishing images on the Internet.

categories	#, parameterisations
Colourising	3, Tint the red, green, or blue channel by 10%
Contrast changes	2, Increase or decrease the contrast ^a
Cropping	4, Crop by 5, 10, 20 and 30%
Despeckling	1, Apply ImageMagick’s despeckling operation
Downsampling	6, Downsample by 10, 20, 30, 40, 50, 70 and 90% ^b
Flipping	1, Flip along the horizontal axis
Colour depth reduction	1, Reduce the colour palette to 256 colours
Outer frame	4, Add an outer frame of 10% the image size
Rotation	3, Rotate by 90, 180 and 270°
Scaling	6, Scale up by 2, 4, 8 times, and down by 2, 4, 8 times ^c
Saturation changes	6, Change the values of the saturation channel by 70, 80, 80, 90, 110, 120 and 130%
Intensity changes	4, Change the intensity channel by 80, 90, 110 and 120%

^ausing ImageMagick’s [Still, 2005] default parameter

^bwithout antialiasing filtering

^cwith antialiasing filtering

for about one thousand images that are grey-levels. The second collection contains photographs of 9000 paintings. The use of collections with varied contents permits to assess the performance of the duplicate detection algorithms in a variety of situations. For instance, the first collection contains photographs covering a wide-range of scenes while the second collection contains very similar images in terms of colours and textures.

The collections are randomly split into two mutually exclusive subsets \mathcal{O} and \mathcal{F} . The set \mathcal{O} represents the originals and contains 200 images, and the set \mathcal{F} are images that are used to estimate the false positives error rate of the system.

The test duplicates are generated by applying two sets of transformations on the original images. The first set of transformations, denoted *Qamra*, is the same as that used in [Ke *et al.*, 2004; Qamra *et al.*, 2005]. It represents transformations often encountered when publishing images on the Internet. There are twelve categories of transformations, as shown in table 4.1. An example for each of them is depicted in figure 4.4. The second set of transformations, denoted *StirMark*, is based on the duplicates generated to assess the robustness of watermarking methods, namely *StirMark* benchmark version 3.1 [Petitcolas and Kutter, 2001]. It mainly concerns geometric transformations. There are fourteen categories of transformations, as shown in table 4.2. An example for each of them is depicted in figure 4.5. The number of duplicates per original image is 40 for the test set *Qamra* and to 88 for the test set *StirMark*.

A complete test set consists of two components: a set of images and a set of labels. The set of images, denoted \mathcal{T} , is given by the union of the unrelated images \mathcal{F} and either one of the duplicates sets *Qamra* or *StirMark*. The set of labels, denoted \mathcal{L} , associates each image in \mathcal{T} with a label; namely, 0 for the unrelated images, and different positive numbers ($+1, +2, \dots, +|\mathcal{O}|$) for the duplicates of each original image.

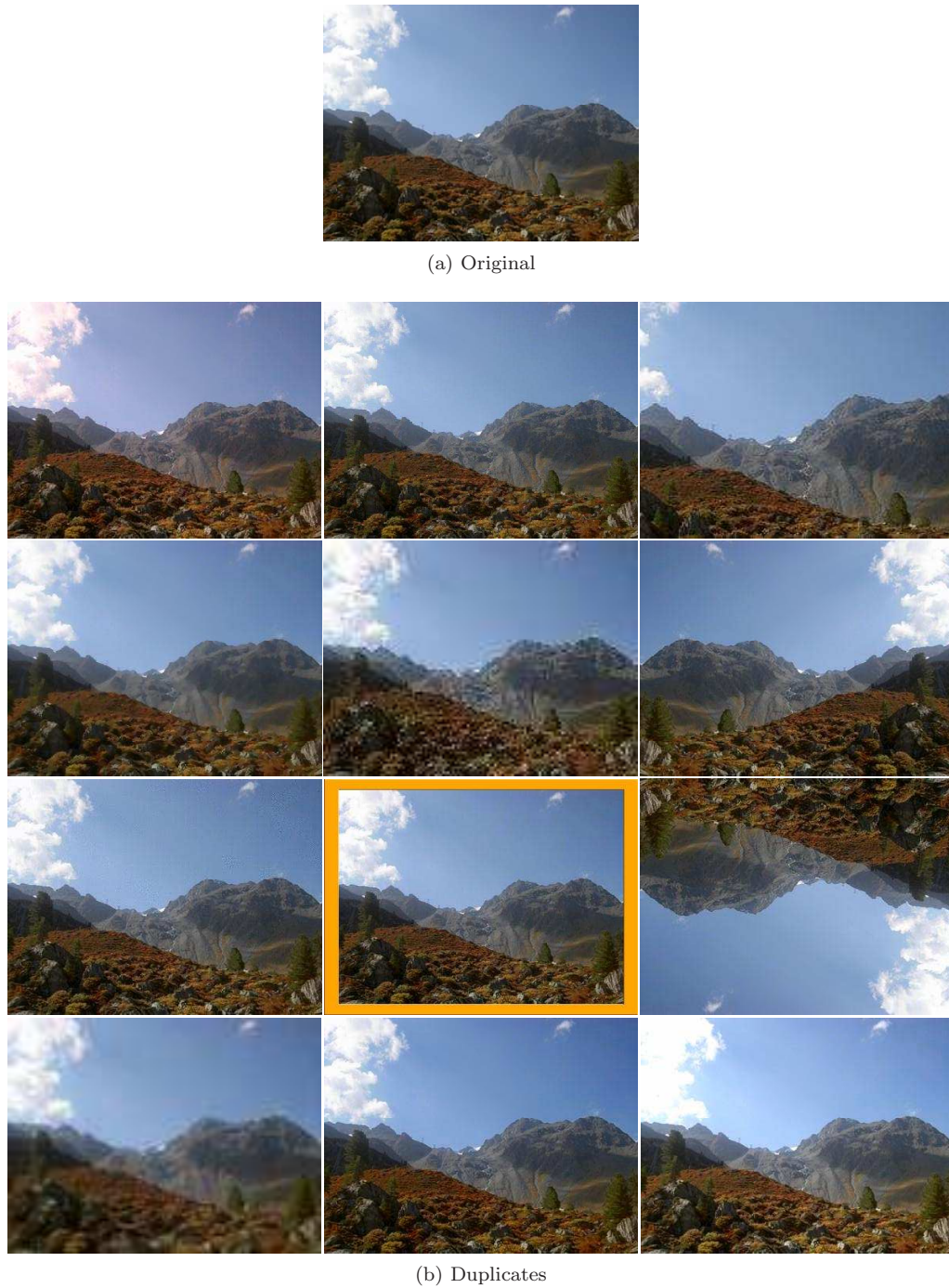
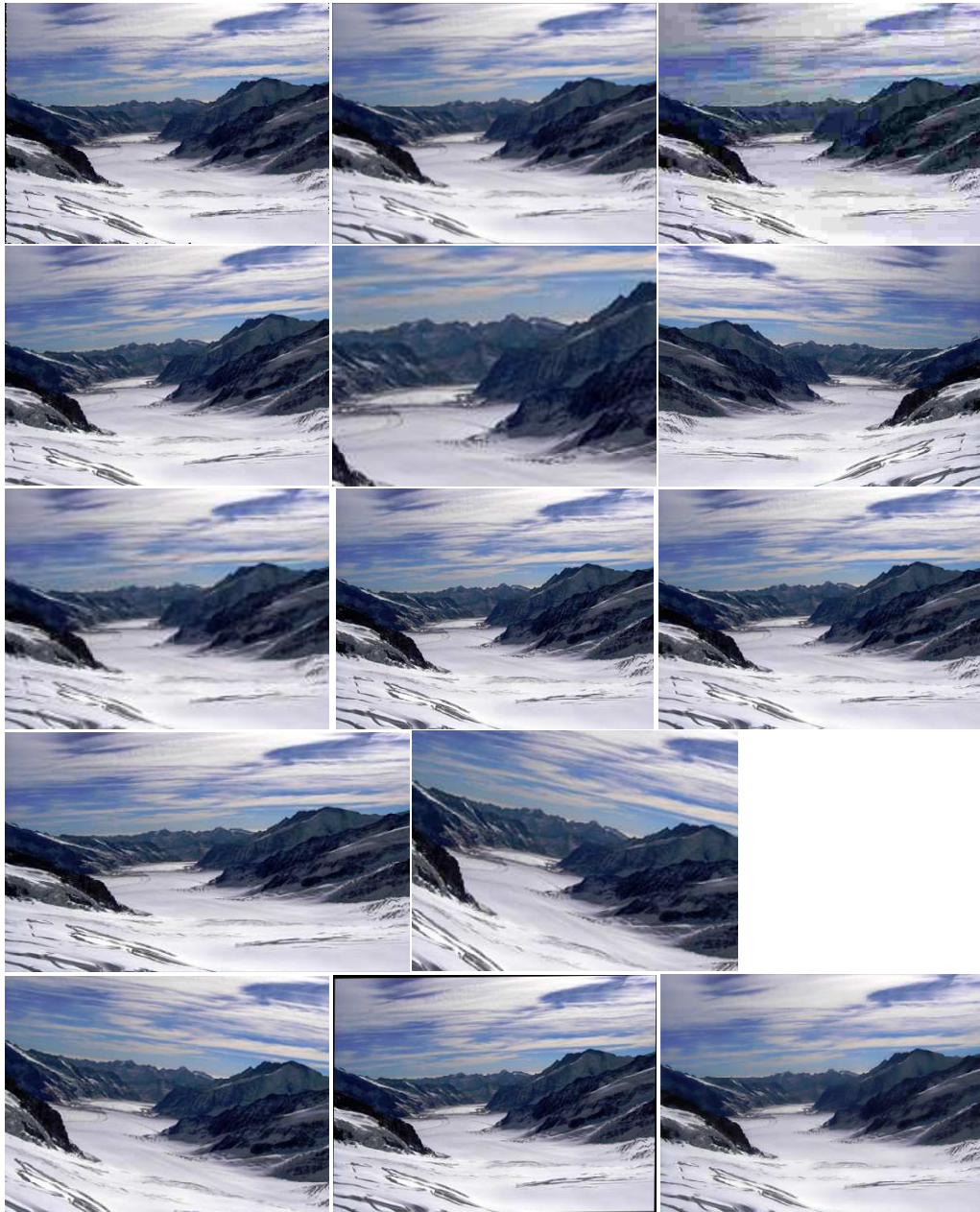


Figure 4.4: *Examples of test duplicates generated by the benchmark Qamra.* There is one duplicate example per category, the order used (left-right, top-down) is the same as in the table 4.1. Every images are resized so to have an equal height. For information, the photograph was taken on the highs of Nendaz — Switzerland.



(a) Original



(b) Duplicates

Figure 4.5: *Examples of test duplicates generated by the benchmark StirMark.* There is one duplicate example per category, the order used (left-right, top-down) is the same than in table 4.2. Every images are resized so to have an equal height. For information, the photograph was taken from the Jungfrauoch — Switzerland.

Table 4.2: *Duplicate images test set StirMark*. This test set contains the duplicates proposed in [Petitcolas and Kutter, 2001] and used in the assessment of most watermarking algorithms. It simulates transformations often encountered when copying images.

categories	#, parameterisations
Median filtering	3, filter of size 2×2 , 3×3 and 4×4
Gaussian filtering	1, approximate filter of size 3×3
JPEG compression	12, JPEG compression with quality factors 90, 80, 70, 60, 50, 40, 35, 30, 25, 20, 15 and 10
Shearing	6, shearing in (X, Y) directions by (0, 1), (0, 5) (1, 0), (5, 0), (1, 1) and (5%, 5%)
Cropping	9, centred cropping by 1, 2, 5, 10, 15, 20, 25, 50 and 75%
Flipping	1, horizontal flip
Scaling	6, scaling by factors 0.5, 0.75, 0.9, 1.1, 1.5 and 2
Line removal	5, removal of (n columns, m rows): (1, 1), (1, 5), (5, 1), (5, 17) and (17, 5)
Random bending	1, ‘StirMark’ random geometric distortions
Aspect ratio	8, change aspect ratio of X(Y) by a factor 0.8, 0.9, 1.1 and 1.2
Rotation	16, rotations by -2, -1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1, 2, 5, 10, 15, 30, 45 and 90°
Rotation/scaling	16, same as above but followed by scaling
Linear transform	3, general linear geometric transformation $\mathbf{c}' = \mathbf{T}\mathbf{c}$ ^a
FMLR	3, frequency mode Laplacian removal attack

^avalues of $\mathbf{T} : \begin{pmatrix} 1.010 & 0.013 \\ 0.009 & 1.011 \end{pmatrix}, \begin{pmatrix} 1.007 & 0.010 \\ 0.010 & 1.012 \end{pmatrix}, \text{ and } \begin{pmatrix} 1.013 & 0.008 \\ 0.011 & 1.008 \end{pmatrix}$

4.3.2 Performance metrics

There are different ways to assess the performance of a duplicate detection system. For example, one can measure the tradeoff between false positives and false negatives error rates, or between precision and recall. For more information about precision and recall, the reader is referred to chapter 3. In the following, we use the paradigm of false positive versus false negative rates because the duplicate detection problem is considered, in this thesis, from a classification point of view and not from a retrieval point of view.

As mentioned previously, the performance of a duplicate detection system can be evaluated through the tradeoff between the false positives and false negatives error rates. A false positive error is a test image that is estimated to be a duplicate of an original but is not. Conversely, a false negative error is a test image that is a duplicate of an original but is not detected as such. In the following, the true class label of a generic test image is denoted by c_t and its estimated class label by c_e . While false positive errors can occur whenever the test image is estimated to be a duplicate, $c_e > 0$, false negative errors happen only when the test image is really a duplicate, $c_t > 0$. In both cases, errors signify that the estimated label and the true label differ. More precisely, given a true class c_t and an estimated class c_e , an error happens if the function $e(c_t, c_e)$, given thereafter, is equal to one.

$$e(c_t, c_e) = 1 - \mathbb{I}_{\{c_t\}}(c_e), \quad (4.13)$$

where $\mathbb{I}_{\mathcal{A}}(x)$ is the indicator function. As said before, a false positive error happens if there is an error and if the estimated label is larger than zero. Accordingly, given a true class c_t and an

estimated class c_e , a false positive error happens if

$$\text{fp}(c_t, c_e) = e(c_t, c_e) \cdot \mathbf{I}_{\mathbb{N}^*}(c_e), \quad (4.14)$$

is equal to one. Similarly, a false negative error means that

$$\text{fn}(c_t, c_e) = e(c_t, c_e) \cdot \mathbf{I}_{\mathbb{N}^*}(c_t), \quad (4.15)$$

is equal to one. Finally, no error signifies that there are no false positive nor false negative errors or, in other words both $\text{fp}(c_t, c_e)$ and $\text{fn}(c_t, c_e)$ are equal to zero.

In the following, we define the false positives and false negatives error rates to be the probability that the corresponding error happens given that the tested image can potentially produce that error. Now, the exact definitions of the error rates depend on whether the duplicate detector knows a single original or multiple originals. For example, a false positive error can potentially happen to any test image if the system knows multiple original images. Indeed, a real duplicate assigned the wrong original is a false positive error. In this case, the false positives error rate is given by $p_{\text{FP}} = \Pr\{c_e \neq c_t, c_e > 0\}$. On the other hand, a false positive error can only happen for unrelated test images if the system knows a single original image. In this case, the false positives error rate is given by $p_{\text{FP}} = \Pr\{c_e \neq c_t | c_t < 0\}$.

Nonetheless, for both types of system, a false negative error is only possible if the test image is a duplicate and the corresponding error rate is given by $p_{\text{FN}} = \Pr\{c_e \neq c_t | c_t > 0\}$. To give a better intuitive understanding, lets consider a p_{FP} equals to 0.05. In this case, five out of one hundred test images are, on average, wrongly detected as duplicates or, in other words, are assigned to the wrong originals. Similarly, a p_{FN} equals to 0.08 means that, on average, eight out of one hundred true duplicates are not detected.

There exists a tradeoff between the false positives and false negatives error rates. Indeed, different values of p_{FP} and p_{FN} are obtained by varying the parameters of the duplicate detection system, for example the threshold u given in section 4.2. The receiver operating characteristic (ROC) curve [Fawcett, 2003] is often used to represent the tradeoff between error types. In this representation the true positive rate, one minus the false negatives error rate, is plotted as a function of the false positives error rate. In this thesis, we use a variant of the ROC curve called detection error tradeoff (DET) curve [Martin *et al.*, 1997].

Contrary to ROC curves, the DET curves represent the false negatives error rate as a function of the false positives error rate. Since both axes correspond to error measurements, they can both make use of a logarithmic scale. The interpretation of DET curves is analogous to that of ROC curves: a classifier X is more accurate than a classifier Y when its DET curve is below that of Y. The exact construction of the DET curve depends on the duplicate detection system used, and is given in the next two subsections.

Performance metrics — single original image system

For the single original duplicate detection model, the false positives and false negatives error rates are equal to $p_{\text{FP}} = \Pr\{c_e \neq c_t | c_t < 0\}$ and $p_{\text{FN}} = \Pr\{c_e \neq c_t | c_t > 0\}$, respectively. They develop as follows

$$p_{\text{FP}} = \Pr\{c_e = +1 | c_t = -1\}, \quad (4.16)$$

$$p_{\text{FN}} = \Pr\{c_e = -1 | c_t = +1\}, \quad (4.17)$$

since in the binary detectors only know two classes.

Hence, an estimate of the false positives error rate, for a selectiveness threshold value of u and a given original image $\mathcal{O}(n)$, is given by

$$\hat{p}_{\text{FP}}(\mathcal{O}(n), u) = \frac{1}{\sum_{j=1}^{|\mathcal{L}|} \mathbb{I}_{\{-1\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{s.t. } \mathcal{L}(i)=-1}}^{|\mathcal{T}|} \text{fp}\left(-1, d_{\mathcal{O}(n)}^1(\mathcal{T}(i), u)\right), \quad (4.18)$$

where \mathcal{T} is the set of test images defined in section 4.3.1, \mathcal{L} is the corresponding set of labels, $d_{\mathcal{O}(n)}^1(\mathcal{T}(i), u)$ is the function (defined in section 4.2.1) that estimates the class label of the test image $\mathcal{T}(i)$ with respect to original image $\mathcal{O}(n)$, $\text{fp}(\cdot, \cdot)$ is the function that indicates a false positive error and is defined in the previous section. Similarly, the estimates for the false negatives error rate, for a selectiveness threshold value of u and a given original image $\mathcal{O}(n)$, is given by

$$\hat{p}_{\text{FN}}(\mathcal{O}(n), u) = \frac{1}{\sum_{j=1}^{|\mathcal{L}|} \mathbb{I}_{\{n\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{s.t. } \mathcal{L}(i)=n}}^{|\mathcal{T}|} \text{fn}\left(+1, d_{\mathcal{O}(n)}^1(\mathcal{T}(i), u)\right). \quad (4.19)$$

To assess the performance of a system that detects the duplicates of a single original, we make use of several detectors. Each detector is tuned to a specific image $\mathcal{O}(n)$. The algorithm's tradeoff between false positives and false negatives error rates is summarised into a single DET curve constructed as follows. For each original image $\mathcal{O}(n)$, a DET curve is produced by gathering the estimated probabilities $\hat{p}_{\text{FP}}(n, u)$ and $\hat{p}_{\text{FN}}(n, u)$ for different values of the threshold u .

All the curves are finally synthesised into a single DET curve, denoted $\overline{\text{DET}}$, by using vertical averaging, algorithm 5 in [Fawcett, 2003]. In the vertical averaging procedure, a false negatives error rate is obtained by averaging the false negatives error rates given by the different DET curves at the same false positives error rate. This implies that a working point on the $\overline{\text{DET}}$ curve corresponds to thresholds that are, possibly, different for each detector. In practice, a lookup table can be used to determine the correct threshold values in function of the chosen working point.

Using vertical averaging on the DET curves permits to have an estimates of the optimal performance of the ensemble of binary detectors regardless of the method used to combine them.

Performance metrics — multiple original images system

For the multiple originals duplicate detection model, the false positives and false negatives error rates are equal to $p_{\text{FP}} = \Pr\{c_e \neq c_t, c_e > 0\}$ and $p_{\text{FN}} = \Pr\{c_e \neq c_t | c_t > 0\}$, respectively. Using the prior probabilities $p_i = \Pr\{c_t = i\}$, the false positives error rate becomes

$$p_{\text{FP}} = \sum_{i=-1}^{|\mathcal{O}|} \Pr\{c_e \neq c_t, c_e > 0 | c_t = i\} p_i, \quad (4.20)$$

$$= \sum_{i=1}^{|\mathcal{O}|} \Pr\{c_e \neq c_t, c_e > 0 | c_t = i\} p_i + \Pr\{c_e > 0 | c_t = -1\} p_{-1}. \quad (4.21)$$

Conversely, the false negative develops as follows

$$p_{\text{FN}} = \frac{1}{\sum_{i=1}^{|\mathcal{O}|} p_i} \sum_{i=-1}^{|\mathcal{O}|} \Pr\{c_e \neq c_t | c_t = i\} p_i, \quad (4.22)$$

$$= \frac{1}{1 - p_{-1}} \sum_{i=1}^{|\mathcal{O}|} \Pr\{c_e \neq c_t | c_t = i\} p_i, \quad (4.23)$$

$$= \frac{1}{1 - p_{-1}} \sum_{i=1}^{|\mathcal{O}|} (\Pr\{c_e \neq c_t, c_e > 0 | c_t = i\} + \Pr\{c_e = -1 | c_t = i\}) p_i \quad (4.24)$$

where equation (4.23) derives from the fact that the events $c_t = +1$ to $c_t = |\mathcal{O}|$ form a partition of the event $c_t \neq -1$, similarly the events $c_e = -1$ and $c_e \neq -1$ form a partition of the outcome space. Using Bayes's theorem, this leads to equation (4.24). Let us define the following quantities

$$p_{D \leftrightarrow D} \equiv \frac{1}{1 - p_{-1}} \sum_{i=1}^{|\mathcal{O}|} \Pr\{c_e \neq c_t, c_e > 0 | c_t = i\} p_i, \quad (4.25)$$

$$p_{D \rightarrow U} \equiv \frac{1}{1 - p_{-1}} \sum_{i=1}^{|\mathcal{O}|} \Pr\{c_e = -1 | c_t = i\} p_i, \quad (4.26)$$

$$p_{U \rightarrow D} \equiv \Pr\{c_e \neq -1 | c_t = -1\}. \quad (4.27)$$

They can be interpreted as follows. $p_{D \leftrightarrow D}$ gives the probability that a wrong original is assigned to a duplicate. Similarly, $p_{D \rightarrow U}$ gives the probability that an actual duplicate is estimated as unrelated to any original images. Finally, $p_{U \rightarrow D}$ gives the probability that an unrelated image is estimated as a duplicate of some original. Then the false positives and false negatives error rates can be expressed in terms of $p_{D \leftrightarrow D}$, $p_{D \rightarrow 0}$, and $p_{U \rightarrow D}$

$$p_{\text{FP}} = p_{D \leftrightarrow D}(1 - p_{-1}) + p_{U \rightarrow D} p_{-1} \quad (4.28)$$

$$p_{\text{FN}} = p_{D \leftrightarrow D} + p_{D \rightarrow U}. \quad (4.29)$$

We can now give an estimator for the probabilities of errors. First note that the priors are usually unknown. On the other hand, we can assume that the test image is more frequently an

unrelated image than a duplicate of an original; thus $p_{-1} \gg \sum_{i=1}^{|\mathcal{O}|} p_i$. It implies that $p_{-1} \approx 1$, and conversely that $1 - p_{-1}$ is very small. Let us define $\alpha \equiv \frac{p_{-1}}{1-p_{-1}} \approx (1 - p_{-1})^{-1} \gg 1$. Then, an approximation for the probability of false positive is the following

$$p_{\text{FP}} \approx p_{D \leftrightarrow D} \alpha^{-1} + p_{U \rightarrow D}. \quad (4.30)$$

Since $p_{D \leftrightarrow D} \alpha^{-1} \leq \alpha^{-1}$, the false positives error rate is dominated by $p_{U \rightarrow D}$ when $p_{U \rightarrow D} \gg \alpha^{-1}$. Furthermore, note that for a decent duplicate detection system $p_{D \leftrightarrow D}$ is likely to be extremely small since the system has an extensive knowledge about all its original images. In this particular case, a further approximation is carried out: $p_{\text{FP}} \approx p_{U \rightarrow D}$.

We further assume that the priors are the same for all the positive c_t , namely $p_t = (1 - p_{-1}) / |\mathcal{O}|$. In other words, a test image has the same likelihood to be a duplicate of an original or another. In this case $p_{D \leftrightarrow D}$ and $p_{D \rightarrow U}$ simplify to

$$p_{D \leftrightarrow D} = \frac{1}{|\mathcal{O}|} \sum_{i=1}^{|\mathcal{O}|} \Pr \{c_e \neq c_t, c_e > 0 | c_t = i\}, \quad (4.31)$$

$$p_{D \rightarrow U} = \frac{1}{|\mathcal{O}|} \sum_{i=1}^{|\mathcal{O}|} \Pr \{c_e = -1 | c_t = i\}, \quad (4.32)$$

and estimates of the error rates, for a selectiveness threshold value of u , are then given by

$$\hat{p}_{D \leftrightarrow D}(u) = \frac{1}{|\mathcal{O}| \cdot \sum_{j=1}^{|\mathcal{L}|} \mathbb{I}_{\{x:x>0\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{s.t. } \mathcal{L}(i)>0}}^{|\mathcal{T}|} \text{fp}(\mathcal{L}(i), d_{\mathcal{O}}^{\text{N}}(\mathcal{T}(i), u)), \quad (4.33)$$

$$\hat{p}_{D \rightarrow U}(u) = \frac{1}{|\mathcal{O}| \cdot \sum_{j=1}^{|\mathcal{L}|} \mathbb{I}_{\{x:x>0\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{s.t. } \mathcal{L}(i)>0}}^{|\mathcal{T}|} \text{fn}(\mathcal{L}(i), d_{\mathcal{O}}^{\text{N}}(\mathcal{T}(i), u)), \quad (4.34)$$

$$\hat{p}_{U \rightarrow D}(u) = \frac{1}{\sum_{j=1}^{|\mathcal{L}|} \mathbb{I}_{\{-1\}}(\mathcal{L}(j))} \sum_{\substack{i=1 \\ \text{s.t. } \mathcal{L}(i)=-1}}^{|\mathcal{T}|} \text{fp}(\mathcal{L}(i), d_{\mathcal{O}}^{\text{N}}(\mathcal{T}(i), u)), \quad (4.35)$$

where \mathcal{T} is the set of test images defined in section 4.3.1, \mathcal{L} is the corresponding set of labels, $d_{\mathcal{O}}^{\text{N}}(\mathcal{T}(i), u)$ is the function (defined in section 4.2.2) that estimates the class label of the test image $\mathcal{T}(i)$ with respect to the set of original images \mathcal{O} , $\text{fp}(\cdot, \cdot)$ and $\text{fn}(\cdot, \cdot)$ are the functions that indicate false positive and false negative errors as defined previously in this section.

To assess the performance of an algorithm that detects the duplicates of multiples originals, we make use of a single system tuned to the set of original images. The DET curve is constructed by gathering the probabilities \hat{p}_{FP} and \hat{p}_{FN} estimated using the probabilities $\hat{p}_{D \leftrightarrow D}(u)$, $\hat{p}_{D \rightarrow U}(u)$, and $\hat{p}_{U \rightarrow D}(u)$ computed for different values of threshold u .

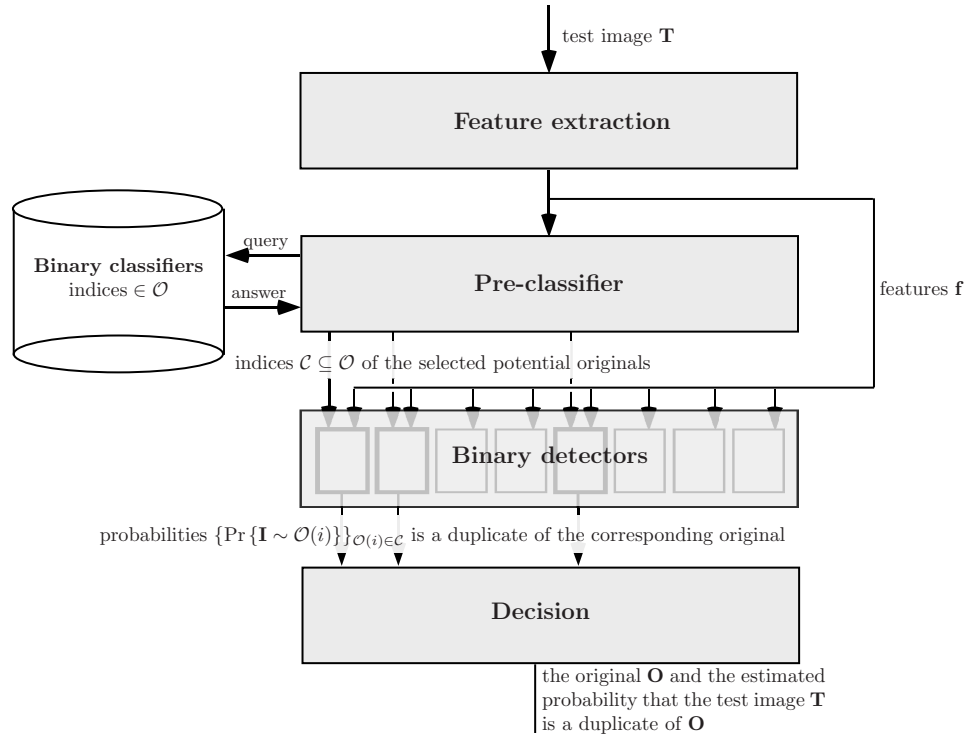


Figure 4.6: *Block diagram of the duplicate detection approach.* The four main steps are detailed thereafter. The feature extraction step extracts relevant visual features \mathbf{f} from the test image \mathbf{T} . The pre-classifier step selects potential original images $\mathcal{C} \subseteq \mathcal{O}$ from an indexing structure. The latter indexes the corresponding duplicate manifolds and uses indices among the set \mathcal{O} . For each most probable original image $\mathcal{O}(i) \in \mathcal{C}$, the binary detectors estimate the probabilities $\Pr\{\mathbf{I} \sim \mathcal{O}(i)\}$ that the test image \mathbf{T} is a duplicate of the original image $\mathcal{O}(i)$. Finally, the decision step either decides that the test image \mathbf{T} is unrelated to any of the selected potential original images in \mathcal{C} or that $\mathbf{O} \in \mathcal{C}$ is the most probable original.

4.4 Approach overview and common components

The approach proposed in this thesis relies on the observation made in section 4.1. More precisely, the duplicates of an image lie, under certain conditions, on a smooth manifold. We further assume that the manifolds defined by different original images are different in location as well as in shape. This additional assumption leads to an approach where the manifolds are explicitly estimated for every original image. Following this idea, an efficient method that estimates the probability that a test image lies on this manifold is first proposed. In the following, we call this single original image duplicate detector a binary detector. Then, this approach is extended to the case where many original images are available and that the test image must be asserted to be a duplicate of one of them or unrelated to any of them. This extended duplicate detection approach is divided into four main steps, which are illustrated with a block diagram in figure 4.6. In section 4.4.2, the feature extraction step is introduced. The remaining steps are detailed in the two next chapters. More precisely, the binary detectors step is detailed in the chapter 5 while the pre-classifier step and the decision are elaborated in chapter 6.

As said previously, the feature extraction step is detailed in this section. It is a common component used in both chapter 5 and chapter 6. The feature extraction step is composed of two parts. In the first part, the test image \mathbf{T} is preprocessed as detailed in section 4.4.1. In the second parts, visual information is extracted from the preprocessed image as presented in section 4.4.2.

4.4.1 Image preprocessing

This section describes the preprocessing operations that are applied to an image before feature extraction. The preprocessing operations have two goals. First, the image is described in a colour space that permits to easily extract meaningful information. And second, a weak form of robustness to transformations such as resizing, framing, or changes in intensity, is introduced. In the following, we suppose that an image of height I and width J is given by three $I \times J$ matrices \mathbf{R} , \mathbf{G} , and \mathbf{B} corresponding to the Red, Green, and Blue channels, respectively. The matrices are indexed as follows. $\mathbf{R}_{(i,j)}$ corresponds to the element given on the i -th line of the j -th column of \mathbf{R} . Similarly, $\mathbf{R}_{(i,\cdot)}$ corresponds to the i -th line, and $\mathbf{R}_{(\cdot,j)}$ to the j -th column.

Before anything else, we introduce a weak robustness to framing by removing nearly constant lines and columns. The removal of lines and columns is performed in an iterative way; a single iteration is described in the following. We first compute, for each line i , the standard deviation $\sigma_{(R,i)} = \text{std}(\mathbf{R}_{(i,\cdot)})$ on the red channel, and similarly $\sigma_{(G,i)}$ and $\sigma_{(B,i)}$ on the green and blue channels. Then, the averaged standard deviation σ_i is computed for each line i and each colour channel. Finally, a line i is removed only if the corresponding averaged standard deviation σ_i is smaller than $s \cdot \sum_i \sigma_i / \tilde{J}$ where s is a parameter that controls the sensitivity of the line removal algorithm, and \tilde{J} is the current number of columns. For the first iteration, \tilde{J} equals the number of columns in the image. Typically, s is set to 0.1 in the following. Similarly, nearly ‘constant’ columns are then removed. The processus is finally iterated as long as there exist lines or columns to be removed.

Then, a weak form of scale invariance is introduced by resizing the image. More precisely, the image is resized such that it contains approximately 2^{14} pixels, this number corresponds to a square image of 128×128 pixels, while keeping its original aspect ratio. Apart from the weak form of scale invariance, the size of preprocessed image is mostly constant regardless of the test image size, it also permits to speed up feature extraction by reducing the number of pixels to process.

The scaled image is then represented in a modified hue saturation intensity (HSI) space: the logarithmic Hue, Saturation, and equalised-Intensity space. More specifically, the logarithmic Hue \mathbf{H}^{\log} is defined as follows [Finlayson and Schaefer, 2001]

$$\mathbf{H}^{\log} = \frac{\log \mathbf{R} - \log \mathbf{G}}{\log \mathbf{R} + \log \mathbf{G} - 2 \log \mathbf{B}}, \quad (4.36)$$

where \mathbf{R} , \mathbf{G} and \mathbf{B} are the red, green and blue channels, and the operations are performed element-wise. The logarithmic Hue has the advantage to be invariant to gamma and brightness changes. The Saturation \mathbf{S} is the same as for classical HSI [Gonzalez and Woods, 2002, chapter 6], and is given by

$$\mathbf{S} = 1 - \frac{3 \cdot \min(\mathbf{R}, \mathbf{G}, \mathbf{B})}{\mathbf{R} + \mathbf{G} + \mathbf{B}}, \quad (4.37)$$

Table 4.3: *Used features overview.* This table lists the used features, the statistic types, and the number of extracted values.

name	feature type	number of features
Gabor	mean of the squared coefficients	30
Gabor	standard deviation of the squared coefficients	30
Colour	class histogram	10
Colour	mean of each class	24
Colour	standard deviation of each class	24
Colour	spatial distribution of each class	20
Grey-level	class histogram	8
Grey-level	spatial distribution of each class	16
		total = 162

where the operations are applied element-wise. By construction, the Saturation is quite invariant to changes in illumination. Finally, the equalised Intensity \mathbf{I}^{equ} is given by

$$\mathbf{I}^{equ} = \text{equ} \left(\frac{\mathbf{R} + \mathbf{G} + \mathbf{B}}{3} \right), \quad (4.38)$$

where $\text{equ}(\cdot)$ is the global histogram equalisation operator [Gonzalez and Woods, 2002, section 3.3], and the addition and division operations are performed element-wise. The equalisation permits to make the Intensity mostly invariant to changes of gamma and brightness as shown in [Maret *et al.*, 2006a].

4.4.2 Features

This section introduces the features used for the experiments carried out through the remaining chapters. Note that a more thorough discussion on general visual feature extraction can be found in chapter 2. The features are extracted after the image has been processed as described in section 4.4.1.

The features used in this thesis are of three types: texture, colour and grey-level statistics. They are similar to those used in [Qamra *et al.*, 2005], in which they are found to give good results in image duplicate detection applications. The main differences are the added 24 grey-level features and the absence of ‘local’ statistics. The added grey-level features capture a characteristic missed by the colour features, namely the distribution of the intensity level, and bring increased performance as demonstrated in chapter 5 and chapter 6. Table 4.3 summarises the 162 used features. Each group of features is then detailed in the following subsections.

Texture

The texture features are composed of the first and second order statistics of each sub-band of the Gabor transform. The latter is performed as in [Manjunath and Ma, 1996] on the equalised intensity channel \mathbf{I}^{equ} . To be self-contained, the construction of the set of Gabor filters is now summarised. The two-dimensional Fourier transform of the mother Gabor filter $G(u, v)$ can be

written as follows

$$G(u, v) = \exp -0.5 \left(\frac{(u - W)^2}{\sigma_u^2} + \frac{v^2}{\sigma_v^2} \right), \quad (4.39)$$

where u and v are the horizontal and vertical frequencies, respectively. The parameter W controls the central horizontal frequency while the parameters σ_u and σ_v control the filter horizontal and vertical widths, respectively. A set of oriented Gabor filters $\{G_{mn}(u, v)\}_{mn}$ can then be obtained by performing a change of variables on the mother filter

$$G_{mn}(u, v) = a^{-m} \cdot G(a^{-m}u \cos \theta + a^{-m}v \sin \theta, -a^{-m}u \sin \theta + a^{-m}v \cos \theta), \quad (4.40)$$

where $\theta = n\pi/K$, K is the total number of orientations, a is a factor larger than one, and n and m are the orientation and scale indices, respectively. The set of Gabor filters is quite redundant because the filters overlap. In [Manjunath and Ma, 1996], this redundancy is reduced by choosing the filters parameters (a , σ_u , σ_v , and W) so that the half-peak magnitude contours of adjacent filters touch but do not overlap. These parameters are controlled by four meta-parameters: the upper centre frequency of interest U_h , the lower centre frequency of interest U_l , the number of orientations K and the number of scales S . The parameters are then given by

$$a = (U_h/U_l)^{-\frac{1}{S-1}}, \quad (4.41)$$

$$\sigma_u = \frac{(a-1)U_h}{(a+1)\sqrt{2 \ln 2}}, \quad (4.42)$$

$$\sigma_v = \tan\left(\frac{\pi}{2K}\right) (U_h - 2 \ln(\sigma_u^2/U_h)) \left(2 \ln 2 - \frac{(2 \ln 2)^2 \sigma_u^2}{U_h^2}\right)^{-0.5}, \quad (4.43)$$

and $W = U_h$, $n = 0, 1, \dots, K-1$, and $m = 0, 1, \dots, S-1$. Additionally, the filters sensitivity to global intensity changes is eliminated by adding constants to the Gabor filters so that the means of their real parts are equal to zero.

In this thesis, the parameters actually used are $U_h = 0.75$ for the upper centre frequency, $U_l = 0.05$ for the lower centre frequency, five scales $S = 5$ and six orientations $K = 6$. This results in a total of 30 filters. Then, the image is filtered using these 30 filters, resulting in 30 sub-band images. The obtained 30 sub-band images are finally summarised to the estimates of the means and the standard deviations of their squared coefficients. This results in a total of 30 mean and 30 variance estimates.

Colour

The colour features are computed in the HSI colour space. Each pixel in the image is classified into one of ten colour classes depending on its position in this space. The classes are the achromatic colours ($S = 0$) black, grey and white, and the chromatic colours ($S > 0$) red, orange, yellow, green, cyan, blue and purple. The equalised intensity is used to classify a pixel into one of the three achromatic classes. The logarithmic Hue is used to classify a pixel in one of the seven chromatic classes.

This is similar to the culture colour approach proposed in [Smith and Chang, 1995] and used

in the duplicate detection system presented in [Qamra *et al.*, 2005]. In this study, pink and brown are also considered, whereas in our case they are classified as red or orange. Brown and pink have similar values for the Hue channel as red or orange, but differ in the Intensity and/or Saturation channels. Operations such as saturation or intensity changes are common in image processing; they modify the Intensity 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.

A colour classes histogram is first computed. It gives the proportion of each colour class in the image. It is normalised such that it sums to one, and comprises 10 values. Channel statistics are then computed. For each colour class, chromatic or achromatic, mean and variance estimates of the equalised Intensity channel are computed. On the other hand, mean and variance estimates of Saturation and logarithmic Hue channels are computed only for the chromatic colour classes. This results in a total of 24 mean and 24 variance estimates. The shape of the spatial distribution of each colour class is finally computed. This is achieved by computing two shapes characteristics for each colour class, namely spreadness and elongation [Hu, 1962; Leu, 1991]. The first characteristic measures the compactness of the spatial distribution of a colour class. The second gives the ratio between the shape length and width. Note that even if pixels assigned to a colour form totally disconnected components, this feature still captures useful information, namely the spatial distribution of these components. This results in a total of 10 spreadness and 10 elongation measures.

Grey-Level

The grey-level features are based on the equalised Intensity channel of the HSI model. The dynamic range of the image is linearly partitioned into eight bins corresponding to as many classes. Each pixel of the image falls into one of these bins.

The use of grey-level feature is important because the colour features can be unsuited in some cases. For instance, it can happen when the reference or the test images are grey-level, or when conversion to grey-level is one of the considered operations in the duplicate detection system.

A grey-level classes histogram is first computed. It gives the proportion of the image's pixels for eight intensity ranges. It is normalised such that it sums to one, and comprises 8 values. Similarly to colour, the shape of the spatial distribution of each grey-level class is finally computed. This results in a total of 8 spreadness and 8 elongation measures.

4.5 Chapter summary

In this chapter, a framework for image duplicate detection is presented. The duplicate detection framework first consists in a model of duplicates. This model permits to explore the characteristics of the subspace spanned by the duplicates of an image; for example it is found that, under certain assumptions, this subspace is a manifold embedded within the image space. The second element of the framework is a generic duplicate detection system. Through this generic system, we develop our view of duplicate detection, namely, the classification of a test image into one of $K + 1$ classes.

K classes correspond to the K original images known to the system, or in other words “the test image is a duplicate of one of the known originals,” while the remaining class stands for “the test image is unrelated to any of the known original images.” Finally, the last element of the framework concerns the evaluation methodology of a duplicate detection system based on the classification approach.

In this chapter, an overview of the duplicate detection system proposed in this thesis is given. The system is composed of four steps: feature extraction, pre-classification, binary detectors and decision. Feature extraction consists in describing images by means of relevant visual statistics. The pre-classifier aims at selecting a limited number of originals among the K original images; an original is selected if the test image is potentially one of its duplicates. The binary detectors are a set of several binary classifiers; each binary classifier determines the probability that the test image is a duplicate of the corresponding original image. Note that only the binary classifiers corresponding to originals selected by the pre-classifier are used. In the last step, the decision simply consists in selecting the most probable original.

The feature extraction step is entirely described within this chapter while the binary detectors and the pre-classifier are the objects of the two following chapters.

A Single Original Duplicate Detection System

5

In this chapter, we detail our approach to image duplicate detection of a single original image. The approach is partially based on previous works [Maret *et al.*, 2005a, 2006a, 2005b]. The main idea behind the proposed system is to adapt duplicate detection to a specific original image. The system is then able to classify test images as duplicates of the original image or as unrelated images. An overview of the system is first given in section 5.1. The training example, as well as the performance metric, used to build the detectors are given in section 5.2. Then, the system is thoroughly described in section 5.3. The system performance and analysis are then detailed in section 5.4. Finally, possible research directions are proposed in section 5.5.

5.1 System overview

We now present an overview of the proposed single original image duplicate detection system, or binary detector. The system consists of four steps as shown in figure 5.1, each of them outlined thereafter. Before going further, notice that the method can be decomposed into two distinct parts. The first one, consisting of the step shown in the upper part of figure 5.1, is independent from the original image. Conversely the second one, comprising the steps shown in the lower part of figure 5.1, depends on the original image; training is therefore needed. The used training examples and metric are presented in section 5.2 while additional details about step-dependant training procedures are given, along with the thorough description of each step, in section 5.3.

Feature extraction The goal of the feature extraction step is to map images into a common space, where comparisons are more efficient. For this purpose global statistics, such as colour channels and textures, are extracted from the test image. The test image \mathbf{T} is first preprocessed as described in section 4.4.1. Then features are extracted for the preprocessed image. The list of

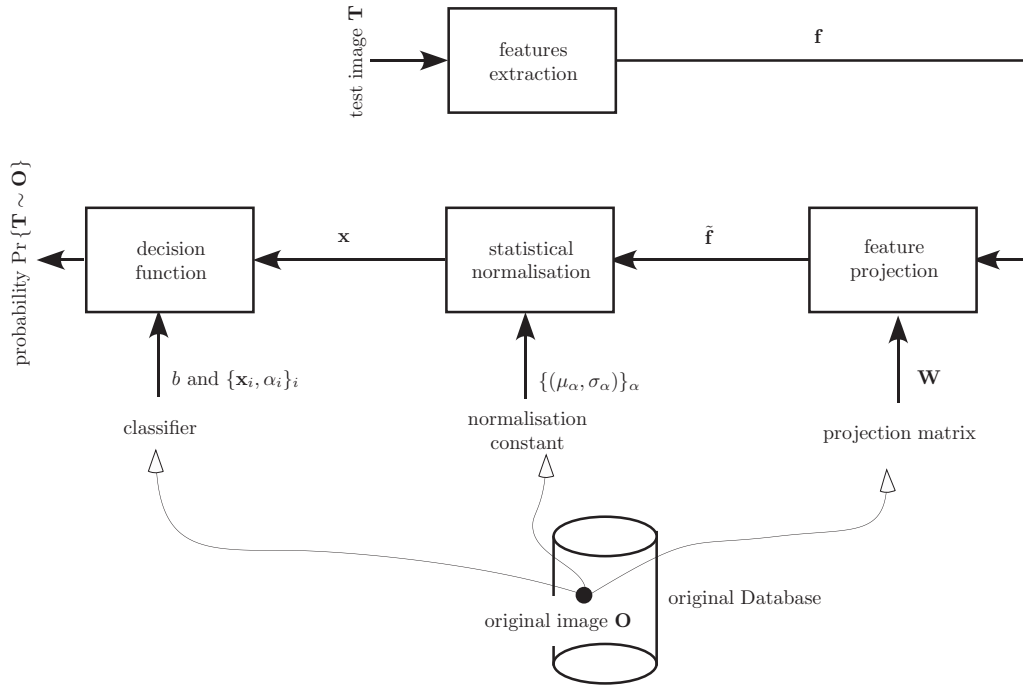


Figure 5.1: *Block diagram for a binary detector.* A test image is given to the system, which determines if it is a duplicate of the original image \mathbf{O} for which the detector is built. The method can be decomposed into two distinct parts: a step that is independent from the original image, upper part of the figure, and steps that depend on the original image, lower part of the figure).

features is described into more details in section 4.4.2. This second step results in a feature vector \mathbf{f} containing D elements.

Feature projection In the second step, the features are linearly transformed so as to obtain a better separation between duplicates of the original image and unrelated images. More precisely, the projected features are given by $\tilde{\mathbf{f}} = \mathbf{W} \cdot \mathbf{f}$ where \mathbf{W} is a $D \times D$ projection matrix. The projection can depend on the original image or be common to every original. In the following, \mathbf{W} is found through a simple principal component analysis (PCA) algorithm and is common to every original.

Statistical normalisation In the third step, the elements of $\tilde{\mathbf{f}}$ are normalised with respect to the statistical distribution of the duplicates. Accordingly, this step's parameterisation depends on the original image. The goal of this step is to give the same importance to each feature, independently of their value range. This results in a normalised vector \mathbf{x} containing D elements.

Decision function In the last step, a non-linear decision function is used to determine the probability $\Pr\{\mathbf{T} \sim \mathbf{O}\}$ that the test image \mathbf{T} , represented by the pattern \mathbf{x} , is a duplicate of the original image \mathbf{O} . Clearly, this step is parameterised according to the original image.

5.2 Remarks on training

As mentioned earlier, the last three steps shown in figure 5.1 need to be parameterised according to the original image and, consequently, require training. In this section, we present the training procedure, which is performed in cascade. Firstly, the projection matrix \mathbf{W} is computed, and the projected features are then normalised. And finally, the decision function is trained using the normalised features.

The remaining of this section is composed of two parts. The first part presents the examples used to train the system while the second part accounts for the metric used to assess the training performances.

Training examples

Examples of duplicate images, positive examples, can be generated artificially. Indeed, the original image can be modified using different operations, resulting in several duplicates. Furthermore, it is possible to have a richer set of training examples by nesting two or more operations to form a new operator known as a composition. However, in this thesis we explore a training method that does not require operations' composition. This is advantageous because it limits the number of training examples. Indeed, the number of training examples generated by using composition grows factorially as the number of nesting levels increases.

In this work, the duplicates are generated by the operations listed in table 5.1. Note that a single training set is used to create detectors that work well with both Qamra and StirMark benchmarks, see section 4.3 for more details on the benchmarks. The choice of these particular training examples is now motivated. We first determine the transformations that, on average, result in feature vectors farther from that of the original. The order of the transformations, as in the previous sense, is experimentally found to be as follows

1. rotation and rotation/scaling;
2. JPEG compression and cropping;
3. saturation changes and intensity changes;
4. colourising;
5. aspect ratio changes and downsampling;
6. scaling and shearing;
7. linear transformation and frequency mode Laplacian removal.

We then assign a number of duplicates per transformation proportional to the corresponding average distance. For instance, more examples corresponding to duplicates generated through rotations are selected than these generated through scaling. The exact breakdown is, for one-hundred training examples, as follows: eleven rotations, eleven rotations and scaling, ten JPEG compression, ten cropping, eight saturation changes, eight intensity changes, six colourising per channel, five aspect ratio changes, five downsampling, four scaling, four shearing, three linear

transformations, and three frequency mode Laplacian removal. Finally, the range of each transformation’s parameterisation is evenly sampled. Note that additional care is taken so as to avoid parameterisations used in the benchmark. For example, ten duplicates should be based on the JPEG compression. Since the JPEG quality parameter ranges, for the test set StirMark, from 90 down to 10 (by steps of 10 above a quality of 40 and by steps of 5 below this mark), we then choose to use JPEG-compressed training examples with the following quality factors 98, 88, 78, . . . , 28, 18, 8. Finally, we added the six duplicates from the following non-parameterisable transformations to the training set

1. contrast changes (plus and minus);
2. despeckling;
3. colour depth reduction;
4. horizontal flipping;
5. grey-level conversion.

Note that this last transformation, grey-level conversion, is neither part of Qamra nor StirMark benchmarks. It is however added so that the detectors work well for grey-level test images. It thus results in the 106 positive examples reported in table 5.1.

Examples of unrelated images, negative examples, can be obtained by using a set of images that are known to be different from the original image. This set can also be enriched by applying operations on its elements. In this study, we only consider the grey-level conversion. It permits to enrich the training set with grey-level images in order to avoid relying too heavily on the colour features. To construct the set of negative examples, 250 images are selected from the image collection. It thus results in 500 negative examples.

Finally, it is important to note the optimal choice of the training examples is still an open issue and is the focus of future research. However, some possible directions are given in section 5.5.

Training metric

The F-score metric is used to assess the detection performance during the training phase. The F-score is defined as follows [Fawcett, 2003]

$$F(\text{TP}, \text{FP}, \text{P}) = \frac{\text{TP}}{\text{P}} \times \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (5.1)$$

where P is the the total number of positive instances, TP is the number of positive instances correctly classified, and FP is the number of negative instances wrongly classified. The first term in the right hand side of equation (5.1) corresponds to the recall. Conversely the second term represents the precision. F-score balances these two conflicting properties: precision increases as the number of false positives decreases, and recall decreases as the number of false negatives diminishes, usually meaning that the number of false positives increases. Equation equation (5.1)

Table 5.1: *Training duplicates*. These duplicates have been found to give rise to duplicate detectors that work well on the Qamra and StirMark benchmarks.

categories	#	parameterisations
Colourising	18	Tint the red, green, or blue channel from -11% to +11% by steps of 4%
Contrast changes	2	Increase or decrease the contrast ^a
Despeckling	1	Apply ImageMagick’s despeckling operation
Downsampling	5	Downsample by 3% to 83% by steps of 20%
Colour depth reduction	1	Reduce the colour palette to 256 colours
Saturation changes	8	Change the values of the saturation channel by -22% to +22% by steps of 6%
Intensity changes	8	Change the intensity with the same parameters than for saturation
JPEG compression	10	JPEG compression with quality factors from 8 to 98 by steps of 10
Shearing	4	shearing in (X°, Y°) directions with X and Y varying from 0° to 6° by steps of 3°
Cropping	10	centred cropping from 2% to 94% by steps of 10%
Flipping	1	horizontal flip
Scaling	4	scaling by factors from 0.45 to 0.95 by steps of 0.2
Aspect ratio	5	change aspect ratio of X(Y) by a factor from 0.75 to 1.25 by steps of 0.2
Rotation	11	rotations by angles from 4° to 92° by steps of 8°
Rotation/scaling	11	same as above but followed by scaling
Linear transform	3	general linear geometric transformation $\mathbf{c}' = \mathbf{T}\mathbf{c}$ ^b
FMLR	3	frequency mode Laplacian removal attack with parameters 0.02, 0.04, and 0.06
Grey-level conversion	1	
total	106	

^ausing ImageMagick’s [Still, 2005] default parameter

^bsame matrices as for testing but with entries multiplied by 0.99.

can be rewritten as

$$F_\rho(\hat{p}_{\text{FP}}, \hat{p}_{\text{FN}}) = (1 - \hat{p}_{\text{FN}}) \times \frac{(1 - \hat{p}_{\text{FN}})}{1 + \rho \cdot \hat{p}_{\text{FP}} - \hat{p}_{\text{FN}}}, \quad (5.2)$$

where $\hat{p}_{\text{FP}} = \text{FP}/N$ and $\hat{p}_{\text{FN}} = \text{FN}/P$ are the estimated false positives and false negatives error rates as defined in section 4.3. $\rho = N/P$ gives the ratio between the number of negative and positive instances. As for equation (5.1), the first term in the right hand side of equation (5.2) corresponds to the recall, and the second one to the precision. In the rest of the document, we use the formulation given by equation (5.2). One drawback of this metric lies in the ratio ρ between the number of negative and positive instances; it has to be known beforehand.

5.3 Binary detector

We now thoroughly describe the proposed single original duplicate detection system. In particular, the steps presented in the lower part of figure 5.1, namely feature projection, normalisation and decision function, are detailed along with the training procedures whenever required. On the other

hand, the step presented in the upper part of figure 5.1, namely feature extraction, has been already described in section 4.4.

5.3.1 Features projection

The idea behind this step is to project the features into a space that permits to separate well duplicates and non-duplicates. The transformation could be linear or non-linear and, additionally, it can be adapted to the original or independent of it. In [Maret *et al.*, 2006a], we use a projection step adapted to each original, namely ICA-fx [Kwak and Choi, 2003]. ICA-fx is a dimensionality reduction method based on independent component analysis [Hyvarinen and Oja, 2000], it adds the class information to the feature vector in order to elect the independent components best suited to the binary classification problem. Further experiments, run for this thesis, showed that better results are obtained by simply using a PCA on a large set of images to produce a projection matrix \mathbf{W} common to all detectors.

More precisely, the PCA algorithm projects the features by finding the directions along which the scatter of the cloud of points is maximised [Duda *et al.*, 2001]. In other words, PCA projections lead to a good representation of the data. By computing the PCA on features representing various images, we thus obtain a projection that separates well, in the sense given previously, the images. For this reason, the PCA is used on images unrelated to the original image in order to find a $D \times D$ projection matrix \mathbf{W} common to all detectors. And the projected features are given by

$$\tilde{\mathbf{f}} = \mathbf{W} \cdot \mathbf{f}. \quad (5.3)$$

Note that the matrix \mathbf{W} depends on the image collection characteristic but is independent from the original image. For this reason, it can be computed on a very large set of feature vectors, and the same matrix can be used for every detector.

5.3.2 Normalisation

The goal of normalisation is to ensure that the feature elements are commensurable or, in other words, that the range of the entries of the feature vectors are comparable. The projected features $\tilde{\mathbf{f}}$ are normalised using a statistical normalisation method [Smith and Natsev, 2002]. More precisely, let $\boldsymbol{\mu}_\alpha$ and $\boldsymbol{\sigma}_\alpha$ be the mean and standard deviation estimates of the α -th projected features over a subset of the training set. More precisely, the training subset consists in the duplicate examples of the original and, to avoid taking into account outliers, training examples for which any feature is an extremum over the training set are ignored. The normalised α -th feature \mathbf{x}_α is then given by

$$\mathbf{x}_\alpha = \frac{\tilde{\mathbf{f}}_\alpha - \boldsymbol{\mu}_\alpha}{k \cdot \boldsymbol{\sigma}_\alpha}, \quad (5.4)$$

where $\tilde{\mathbf{f}}_\alpha$ is the projected feature given in equation (5.3). By Tchebychev's theorem, at least a fraction $1 - 1/k^2$ of the $\tilde{\mathbf{f}}_\alpha$ are within the interval $[-1, 1]$. In the following k is set to 3 so that more than 90% of the features \mathbf{x}_α are within $[-1, 1]$.

5.3.3 Decision Function

The decision function needs to determine whether the vector \mathbf{x} corresponds to a duplicate of the original image. This is a binary classification problem, where the two classes correspond to duplicates and non-duplicates, respectively. The goal is to build, using a limited number of training examples, a classifier that generalises well to novel patterns. Many classification algorithms can be used for this purpose. In published works, we showed that support vector classifier (SVC) yields good performance for the duplicate detection problem. In these works [Maret *et al.*, 2005a,b], the SVC approach is compared to two particular approaches, namely support vector data description (SVDD) and orthotope. The SVDD approach [Tax and Duin, 2004] uses a one-class classifier, similar to SVC, while the orthotope approach [Maret *et al.*, 2005b,c] is an *ad hoc* method based on a high-dimensional rectangle that separates duplicates and unrelated images. The performance of the SVC is found to be superior to those two approaches. Possible reasons are as follows. The SVDD generates very tight boundaries and, hence, is more prone to over-training. For the same reason, the SVDD is more sensitive to the chosen training examples than the SVC. On the other hand, the orthotope approach provides only a very crude approximation and, hence, results in poorer performance than the SVC. In the following, we only use the SVC-based decision function.

The basic SVC [Burges, 1998; Schoelkopf *et al.*, 2000] is a binary classifier that separates two classes with a hyperplane. Furthermore, non-linear kernels allow mapping patterns into a space where they can be better discriminated by a hyperplane. More information about SVC can be found in figure 2.3.4. In the following we first give a quick overview of the ν -SVC before detailing the choice of the training procedure.

Overview of ν -SVC

We use the ν -parameterisation [Chen *et al.*, 2005; Schoelkopf *et al.*, 2000] of the SVC, and a radial basis function as kernel. The dual constrained optimisation problem is given in equation (5.5). In the dual form, the Lagrangian is maximised by optimising the Lagrangian multipliers α_i

$$\max_{\alpha} -\frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \ker(\mathbf{x}_i, \mathbf{x}_j), \quad (5.5)$$

subject to the constraints $\sum_{i=1}^m \alpha_i y_i = 0$, $\sum_{i=1}^m \alpha_i \geq \nu$, and $0 \leq \alpha_i \leq 1/m$, where m is the number of training examples, the \mathbf{x}_i are the training patterns, the y_i are corresponding training labels (-1 for the negative examples and $+1$ for the positive examples), and $\ker(\cdot, \cdot)$ is a kernel function satisfying the Mercer's conditions. In this work, we use a radial basis function (RBF) kernel given by

$$\ker(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \cdot |\mathbf{x}_i - \mathbf{x}_j|^2\right). \quad (5.6)$$

The particular choice of kernel is motivated by several considerations. Not only is the linear SVC a particular case of the RBF kernel, but also the sigmoid and the RBF kernels behave similarly for certain choices of parameters [Keerthi and Lin, 2003]. Additionally, the RBF kernel presents less numerical difficulties than, for instance, the polynomial kernel since the influence of a support vector decays exponentially with respect to its distance. Finally, the RBF kernel is governed by

only one parameter instead of two for the polynomial kernel.

The parameters of this classification technique are $\nu \in [0, 1]$ and $\gamma \in \mathbb{R}^+$. 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 [Chen *et al.*, 2005; Schoelkopf *et al.*, 2000]. The kernel parameter γ controls the complexity of the decision boundary. The constrained optimisation problem given in equation (5.5) can be solved by means of standard quadratic programming techniques.

Finally, the decision function indicates to which class the test pattern \mathbf{z} belongs. It is given by

$$f(\mathbf{z}) = \operatorname{sgn} \left(\sum_{i=1}^m y_i \alpha_i \ker(\mathbf{z}, \mathbf{x}_i) + b \right), \quad (5.7)$$

where the constant b is determined by the support vectors. More precisely, b is given by

$$b = y_k - \sum_{i=1}^m y_i \alpha_i \ker(\mathbf{x}_i, \mathbf{x}_k), \quad (5.8)$$

for all \mathbf{x}_k such that $0 < \alpha_k < 1/m$. The value $f = \sum_{i=1}^m y_i \alpha_i \ker(\mathbf{z}, \mathbf{x}_i) + b$ in equation (5.7) is called the margin, and give the distance to the decision boundary.

A support vector classifier predicts only class label but not the probability of being of that class. In the following, we briefly describe how the SVC is extended for probability estimates. More details are in [Platt, 2000; Wu *et al.*, 2004]. Given two classes of data, the goal is to estimate for any pattern \mathbf{x} the posterior probabilities p_{+1} and p_{-1} , namely

$$p_{+1} = \Pr \{y = +1 | \mathbf{x}\} \quad \text{and} \quad p_{-1} = \Pr \{y = -1 | \mathbf{x}\}. \quad (5.9)$$

One way of transforming the SVC output in probability consists in training directly a classifier using a kernel based on the maximum likelihood. A more appropriate method is proposed by Platt, where a sigmoid function maps the margins into probability estimates [Platt, 2000]. The advantage of this technique is that the posterior probabilities $\Pr \{y = +1 | \mathbf{x}\}$ and $\Pr \{y = -1 | \mathbf{x}\}$ are directly obtained and the class conditional probability need not be estimated. The sigmoid is given by

$$\Pr \{y = +1 | \mathbf{x}\} = \frac{1}{1 + \exp(a \cdot f + b)}, \quad (5.10)$$

where the parameters a and b are estimated by minimising the negative log-likelihood function using known training data and their margin values f . Labels and decision values are required to be independent, so k -fold cross-validation can be used to obtain the decision values [Chang and Lin, 2007].

Basic method to determine the SVC parameters

In the ν -SVC, the kernel parameter γ and the parameter ν are to be determined in order to minimise the generalisation error. The latter is the error obtained when testing novel patterns, patterns not used during training, with a trained decision function.

More precisely, we want to minimise the F-score $F_\rho(\hat{p}_{\text{FP}}, \hat{p}_{\text{FN}})$ where \hat{p}_{FP} and \hat{p}_{FN} are the

estimated generalisation error for false positives, novel non-duplicates classified as duplicates, \hat{p}_{FP} is the generalisation error for false negatives, novel duplicates classified as non-duplicates, and ρ is the ratio between the number of novel non-duplicates and duplicates. In the considered applications, there are usually many more non-duplicates than duplicates so that $\rho \gg 1$. Nevertheless, ρ remains *a priori* unknown. Moreover, \hat{p}_{FP} and \hat{p}_{FN} are also unknown and need to be estimated.

Cross-validation is a popular technique for estimating generalisation errors. 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 is 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 generalisation error. This method has been shown to yield a good estimation of the generalisation error [Duan *et al.*, 2003].

In the following, we use a normalised version of the radial basis function kernel where γ in equation (5.6) is replaced by γ/κ . The normalisation constant κ is set to the second decile of the distribution of the intra-duplicate distances within the training set. It ensures that the optimal value of γ is around one with high probability.

While ν has an intuitive signification, it is not clear what its optimal value is [Chen *et al.*, 2005; Steinwart, 2003]. It was shown that twice \bar{R} , a close upper bound on the expected optimal Bayes risk, is an asymptotically good estimate [Steinwart, 2003]. While no such bound can be easily determined *a priori*, this theorem induces an algorithm to find a good ν by starting with the classification error of a well-trained classifier as an approximation of the optimal Bayes risk [Steinwart, 2003].

In this thesis, a good *a priori* approximation of the optimal Bayes risk is unfortunately unavailable. Consequently, good parameters for γ and ν are estimated through a full grid search. The procedure is divided in two steps, namely coarse and fine grid searches. In each step, a tenfold cross-validation is carried out for each feasible pairs (ν, γ) . The pair for which the estimated F-score is the highest is then chosen. The tried pairs are as follows.

- Coarse search: (γ, ν) for $\nu = 0.1 \cdot k - 0.05, k = 1, \dots, 10$ and $\gamma = 5 \cdot 10^k, k = -3, \dots, 3$.
- Fine search: (γ, ν) for $\nu = \nu_1 + 0.01 \cdot k, k = -5, \dots, +5$ and $\gamma = 0.2 \cdot \sigma_1 \cdot k, k = 1, \dots, +10$.

Here, ν_1 and γ_1 denote the value determined in the first step.

Extended method to determine the SVC parameters

The major challenge behind finding the correct training parameters of the ν -SVC for duplicate detection is twofold. Firstly, overtraining is to be avoided. In other words, novel duplicates should be well classified by the system. Secondly, the decision boundary needs to encompass a volume as small as possible. In other words, the probability that a randomly chosen image falls within the boundary is to be as low as possible.

Now, the method given previously, in the subsection “basic method to determine the SVC parameters,” works well in general but is not entirely suited to the duplicate detection problem. Indeed, there is quite a high probability that the chosen negative examples lie, on average, far

from the duplicate region. In other words, the resulting decision boundary will encompass a larger volume than necessary. Additionally, we want the detection system to be able to detect a large range of duplicate that include, possibly, nested transformations, for examples a change in contrast followed by a low quality JPEG compression.

To take the above particularities into account, we proceed as follows. Recall that the positive training examples consists in the 106 patterns, given in table 5.1, and the negative training examples in 500 patterns. We, first, select the 106 negative examples that are nearer to the duplicate. To achieve this, a hyper-sphere covering all positives examples is computed [Elzinga and Hearn, 1972]. The hyper-sphere is parameterised by its centre and radius. Subsequently, the hyper-sphere sphere can be used to select negative training examples. More precisely, the 106 patterns closer to the hyper-sphere centre are kept while the others are discarded. Training of the SVC is then only performed on these 106 positive and 106 negative patterns.

The second step consists in generating synthetic patterns used to, on the one hand, minimise the volume enclosed by the decision boundary and, on the other hand, maximise the generalisation on novel duplicate images. More precisely, 394 synthetic negative examples are generated as random elements evenly distributed within the hyper-sphere [Tax and Duin, 2001]. By minimising the number of these examples falling within the decision boundary, one indirectly minimise the corresponding enclosed volume. Similarly, 394 synthetic positive examples are generated using linear interpolations of the real positive examples [Chawla *et al.*, 2002]. More precisely, the nearest neighbours of a pattern are linearly mixed, with random positive weights, to produce a synthetic pattern. By maximising the number of these examples falling within the decision boundary, one insures that the detector is not over-training since it works for examples that are slightly different than that used to train the classifier.

Finally, good SVC parameters are found using the same grid search than for the method given previously, in the subsection “basic method to determine the SVC parameters,” but with the following modifications. The k -fold cross-validation is performed on the 212 real patterns and results in estimates for \hat{p}_{FP}^{cv} and \hat{p}_{FN}^{cv} for each point on the grid. These estimates are then corrected as follows. First, a classifier is trained using the parameter corresponding to the current grid point and the 212 real patterns. Then, it is used to classify the 792 synthetic patterns and the classification errors are accounted for. It results in two new estimates $\hat{p}_{\text{FP}}^{synth}$ and $\hat{p}_{\text{FN}}^{synth}$. Finally, the error estimates used to compute the F-score at the corresponding grid point are given by

$$\hat{p}_{\text{FP}} = \lambda \cdot \hat{p}_{\text{FP}}^{cv} + (1 - \lambda) \cdot \hat{p}_{\text{FP}}^{synth}, \quad (5.11)$$

$$\hat{p}_{\text{FN}} = \lambda \cdot \hat{p}_{\text{FN}}^{cv} + (1 - \lambda) \cdot \hat{p}_{\text{FN}}^{synth}, \quad (5.12)$$

where λ is a constant giving more weight to the cross-validation estimates or to the synthetic estimates. In the following, we use $\lambda = 106/392$ which is simply the ratio between the number of real and that of synthetic examples.

5.4 Results

In this section, we present experimental results in order to evaluate the performance of the proposed single original image duplicate detector. The first experiment, presented in section 5.4.1, compares the performance of the proposed duplicate detection system with system based on standard metrics, in this case L_1 and L_2 . The second experiment, described in section 5.4.2, explores the influence of the F-score parameterisation. The third experiment, depicted in section 5.4.3, analyses the performances of the individual detector. The fourth experiment, accounted for in section 5.4.4, presents the storage space and the computational resource required by the proposed duplicate detection system. The final experiments, described in section 5.4.5, analyses the proposed system's performance with respect to two other state of the art methods.

5.4.1 Baseline

In this first experiment, we compare the performance of the proposed system with that of simpler methods. These systems are based on the standard L_1 and L_2 metrics. The goal of this test is to analyse the performance improvements by using complex boundary decisions instead of ellipsoids (L_2) or union of hyper-planes (L_1).

Baseline — L_n -based duplicate detection systems

These methods are based on computing the distance between the normalised feature vector of the original image and that of unknown image. More precisely, the feature vectors are normalised as presented in section 5.3.2 but using mean and variance vectors computed on the entire image collection. The distances, based on the L_1 and L_2 metrics, are then computed between the normalised feature vector of the original image and those corresponding to the test images. More specifically, the distance d between two vectors \mathbf{x} and \mathbf{y} is given by $\sqrt[n]{\sum_{\alpha} |\mathbf{x}_{\alpha} - \mathbf{y}_{\alpha}|^n}$, where $n = 1$ for L_1 and $n = 2$ for L_2 . The resulting distances are then converted in the $[0, 1]$ range as follows

$$\tilde{d} = e^{-d}. \quad (5.13)$$

The \tilde{d} take values close to one for test images whose features are similar to that of the original image. Conversely they take values near zero for test images dissimilar to the original. This mapping permits to compute the DET curves using the same algorithm than for probabilities. Note that the function mapping d to \tilde{d} is not so important. Indeed, as long as it is strictly monotonically decreasing, from one to zero, it results in the same DET curve.

Baseline — experimental setup

We now compare the proposed SVC-based system to systems using distances based on the L_1 and L_2 metrics. For this purposes, the system is parameterised as follows. The ratio between unrelated and duplicate examples, for computing the F-score, is set to $\rho = 10^4$. Additionally, two-hundred original images are used to create two-hundred independent duplicate detectors, as described in section 5.3. Test images, corresponding to duplicate and unrelated images, are then feed to each

duplicate detector. This procedure generates a single DET curve per original image. The resulting two-hundred curves are then synthesised into a single curve by using vertical averaging, refer to section 4.3.2 for more information. $L_{1,2}$ -based systems go through the same procedure, using the same original and test images, and their performance is similarly synthesised into two DET curve.

The performance is evaluated on two different image collections, MM270k — commercial image collection [Ke *et al.*, 2004] — (MM270k) and CGFA — virtual museum [Ke *et al.*, 2004] — (CGFA), which are described in more detail in section 4.3. The first collection contains 18 785 photographs while the second collections contains photographs of 9000 artworks (paintings and drawings). Then, two benchmarks, extensively described in section 4.3, are used to test each collection. They contain the same unrelated images but differ in the duplicates’ generation. The first benchmark, Qamra benchmark (Qamra), contains transformations mainly based on colour modifications. On the other hand, the second test set, StirMark benchmark version 3.1 (StirMark), contains transformation mainly based on geometric modifications.

Baseline — MM270k image collection

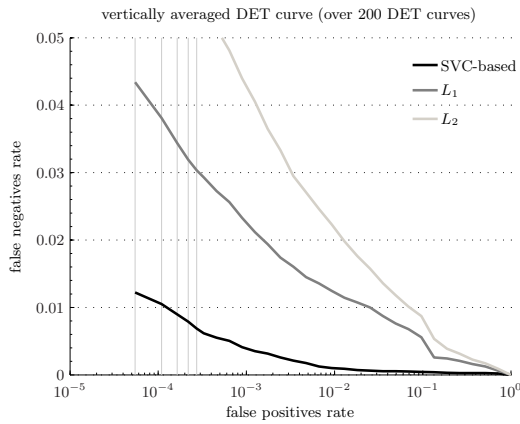
Figure 5.2 shows the performance of the $L_{1,2}$ -based system compared to that of the proposed system for the image collection MM270k and the two benchmarks. For Qamra benchmark, the proposed system displays about a factor two of improvements in terms of false negatives (FNs) error rates for a fixed false positives (FPs) error rate of 10^{-4} as shown in figure 5.2c. On the other hand, when using more difficult transformations, StirMark, the proposed system achieves better than five times less false negatives for a false positives rate of 10^{-4} as shown in figure 5.2d. Finally, figure 5.2a shows the performance when the test set contains the duplicates generated by both Qamra and StirMark. It can be observed that the system perform almost twice as good as L_1 when no false positive is detected. Additionally, figure 5.2b depicts the improvement, in term of false negatives decrease, of the proposed system with respect to its L_1 counterpart. Note that the improvement brought by the proposed duplicate detection system is always above 70%.

Baseline — CGFA image collection

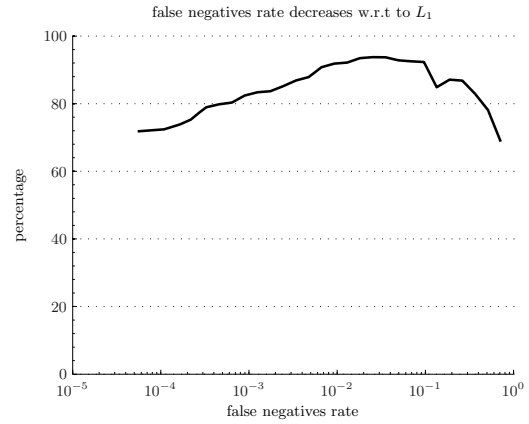
A similar trend can be observed for the more difficult, yet smaller, CGFA image collection, as depicted in figure 5.3. We consider the CGFA image collection more difficult than the MM270k image collection because, one, CGFA contains only paintings and, two, the same painter is often present with more than a work. Contrarily to our expectation, the detection of duplicates performs better on the CGFA collection than on the MM270k collection. The reason is that the MM270k image collection contains many photographs of the same scene but taken from a slightly different location or at a somewhat different time [Ke *et al.*, 2004]. This is illustrated in figure 5.7 and detailed explanation are given in section 5.4.3.

Baseline — L_1 versus L_2

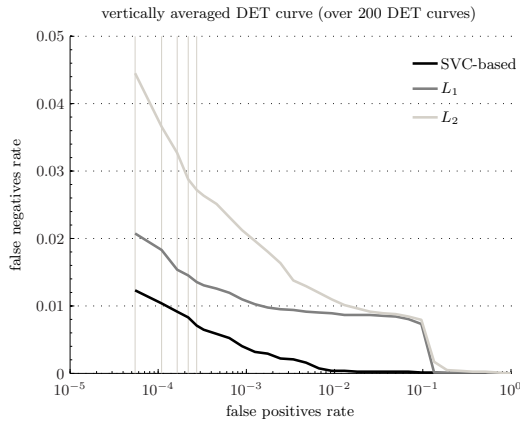
Finally, figure 5.2 and figure 5.3 both show that the L_1 metric always performs much better than the L_2 metric. While this phenomenon is not related to the proposed duplicate detection system, it deserves some explanations. This results was already observed in the context of image retrieval,



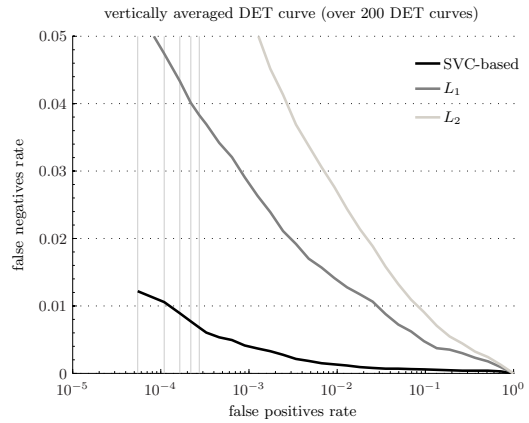
(a) Test set A — include the 128 duplicates generated by both Qamra and StirMark benchmarks. The false negatives error rates are, for no false positive error, 0.043, 0.081 and 0.117 for the SVC, L_1 and L_2 based systems, respectively.



(b) FN's rate decreases w.r.t to L_1 for test set A



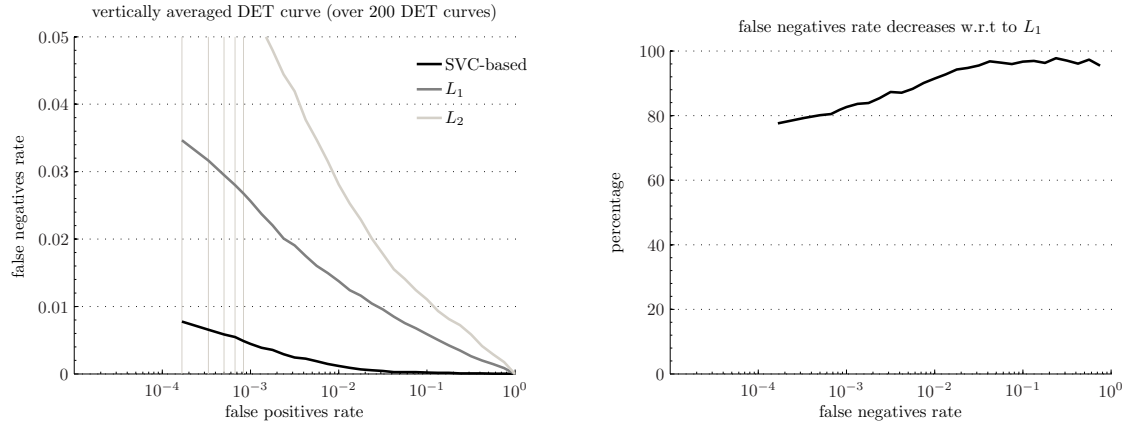
(c) Qamra — include only the 40 duplicates generated by Qamra benchmark benchmark. The FN's error rates are, for no false positive error, 0.041, 0.053 and 0.078 for the SVC, L_1 and L_2 based systems, respectively.



(d) StirMark — include only the 88 duplicates generated by StirMark benchmark version 3.1 benchmark. The FN's error rates are, for no false positive error, 0.044, 0.094 and 0.135 for the SVC, L_1 and L_2 based systems, respectively.

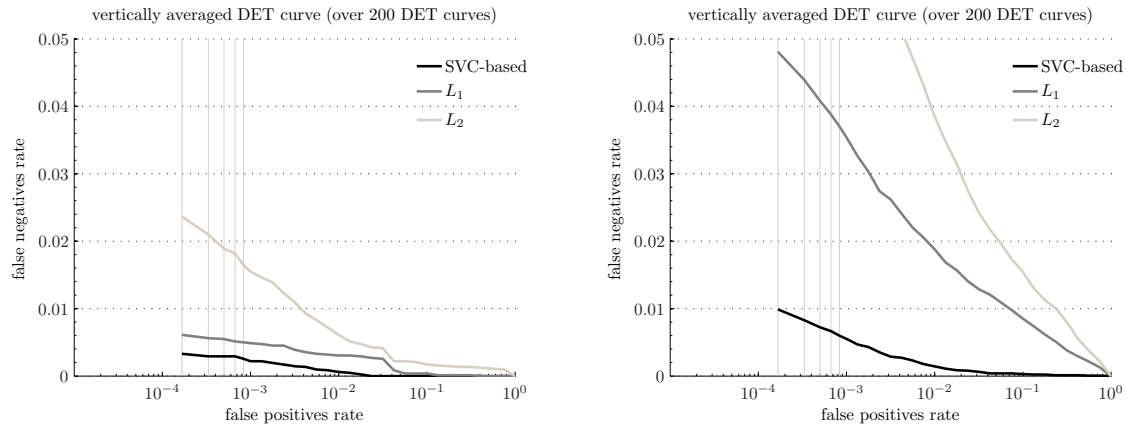
Figure 5.2: *Baselines for the collection MM270k (18 835 images)*. This figure shows the performances that the SVC, L_1 and L_2 duplicate detection systems obtain for the collection MM270k. The vertical lines represent five specific working points: one to five false positives are detected, respectively. Additionally, the working points corresponding to no recorded false positives are given in the sub-captions, instead of the figure, for the three systems.

for example see [Russell and Sinha, 2002]. Russell and Sinha simply concluded that the L_1 metric better captures the features of the human visual system. In the context of duplicate detection, however, a possible reason is as follows. First, notice that if the difference between \mathbf{x}_α and \mathbf{y}_α is below one, elevating the difference to the square results in a smaller value. And conversely, if the difference is above one, the difference to the square results in a larger value. Consequently, if the vector \mathbf{x} and \mathbf{y} represent duplicates, it suffices that a single entry α be much larger than one



(a) Test set A — include the 128 duplicates generated by both Qamra and StirMark benchmarks. The false negatives error rates are, for no false positive error, 0.011, 0.043 and 0.086 for the SVC, L_1 and L_2 based systems, respectively.

(b) FN's rate decreases w.r.t to L_1 for test set A



(c) Qamra— include only the 40 duplicates generated by Qamra benchmark benchmark. The FN's error rates are, for no false positive error, 0.006, 0.009 and 0.032 for the SVC, L_1 and L_2 based systems, respectively.

(d) StirMark — include only the 88 duplicates generated by StirMark benchmark version 3.1 benchmark. The FN's error rates are, for no false positive error, 0.014, 0.060 and 0.112 for the SVC, L_1 and L_2 based systems, respectively.

Figure 5.3: *Baselines for the collection CGFA (9600 images)*. This figure shows the performances that the SVC, L_1 and L_2 duplicate detection systems obtain for the collection CGFA. The vertical lines represent five specific working points: one to five false positives are detected, respectively. Additionally, the working points corresponding to no recorded false positives are given in the sub-captions, instead of the figure, for the three systems.

to obtain quite a large distance. Figure 5.4 shows the histogram of the differences, on all entries and for the 128 duplicates generated by the Qamra and StirMark benchmarks. While more than 90% of the differences are smaller than one, about 10% are outside this interval and very small percentage of the differences are quite large. In other words, 10% of the differences are somewhat arbitrary amplified while 90% of them are made smaller. On the other hand, this phenomenon does not occurs for the L_1 metric. This theory is also supported by the good performance obtained by DPF, see section 3.3.1 and section 5.4.5, where entries corresponding to larger differences are

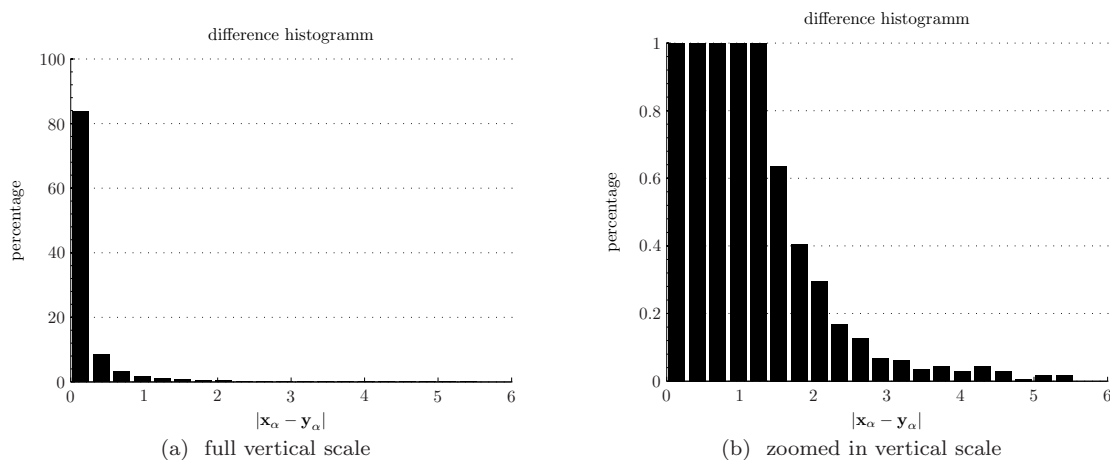


Figure 5.4: *Difference histogram*. This figure shows the histogram of the differences between the entries of vectors, corresponding to the features of duplicates, and the entries of the vector corresponding to the original.

not used to compute the distance. In short, the L_1 metric is more robust to outliers than the L_2 metric. This is a well-know result in other fields such as estimation theory.

Baseline — conclusions

These results show that the proposed system is interesting when difficult transformations are considered. Moreover, notice that the performances of the L_1 metric deteriorates rapidly as the working point moves to smaller false positives rates. This result is of particular importance because we argue that, depending on the application, the working point of a real-world duplicate detection system would be in the magnitude of the 10^{-6} , or even 10^{-7} , as million of images are to be checked. In this case, the proposed system is clearly a better choice. However, the scalability of the proposed system cannot be proven without further tests, which are part of the future works.

5.4.2 Influence of the F-score metric parameterisation

In this second experiment, we explore the effect of possible parameterisations of the F-score metric $F_\rho(\cdot)$. The value ρ gives the ratio between the number of expected non-duplicate instances and that of expected duplicate instances. In the considered applications, these numbers can hardly be determined *a priori*. However, we can safely assume that ρ is much larger than one because there are many more non-duplicates than duplicates.

The experiment is carried out only on the MM270k image collection and both Qamra and StirMark benchmarks are used. Figure 5.5a shows the DET curve for $\rho = \{10^0, 10^3, 10^5\}$. Additionally, figure 5.5b depicts the FN rates decrease brought by using $\rho = \{10^5, 10^3\}$ instead of $\rho = 10^0$. Note that the peaky nature of the curves above 10^{-3} FPs error rates is due to the low values of the FN error rates and also to the relatively small differences between the three curves in figure 5.5a.

Globally, different values of ρ influence only slightly on the results, namely the absolute

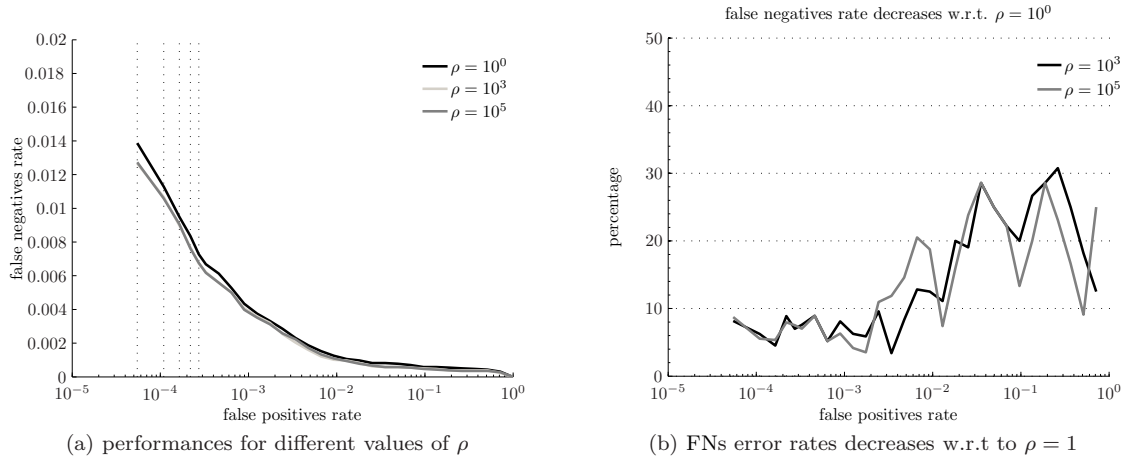


Figure 5.5: *Influence of the F-score parameterisation.* This figures depicts the influence of the F-score parameterisation. Different values of ρ , giving the ratio between the number of expected non-duplicate instances and that of expected duplicate instances, are used.

differences are less than 0.2% as shown in figure 5.5a. In the case of this particular test, set the correct value of ρ would be $160 = (18\,835 - 1)/118$ because each detector is tested with $18\,835 - 1$ unrelated images and 118 duplicates. However, high ρ values favour classifiers with very low false positives error rates while keeping reasonable false negatives error rates. Indeed, higher values of ρ signify that false positives errors are more penalised during the cross-validation procedure. This means that the larger ρ , the smaller the volume enclosed by the decision boundary becomes. Consequently, the probability that a negative example falls within this boundary is made smaller.

All in all, the influence of the F-score parameterisation is thus quite low. This is a positive fact because it means that even if the *a priori* estimate of ρ is quite off, the performance hit suffered by the system will not be very important. In the following, we choose to use an intermediate value for ρ , namely $\rho = 10^4$. With this choice, much larger than the correct value 160, the idea is to improve the performance for low false positives rates. Of course, it remains to be seen if this allegation holds true for very low FPs error rates, which necessitates further experimentations with much larger test sets.

5.4.3 Distribution of the false negatives error rates for no false positive error

We now analyse the distribution of the DET curves before vertical averaging. To achieve this, a specific working point is selected on the DET curves. More precisely, the FNs error rates are recorded when no false positive error is achieved by the detectors. As before, the experiments are carried out on the MM270k and CGFA image collections. The benchmark used is the largest one, namely both Qamra and StirMark are used to generate the duplicates.

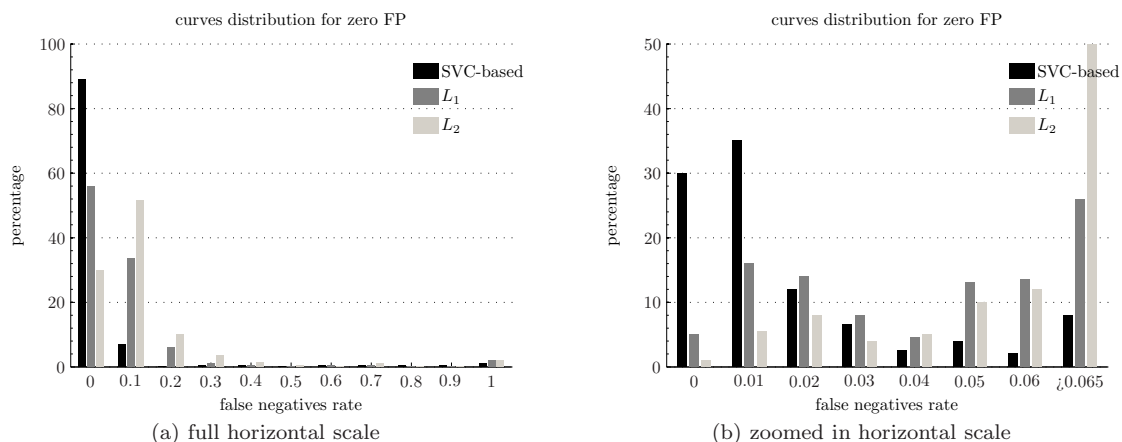


Figure 5.6: *FNs error rates distribution for MM270k image collection.* This figure depicts the FNs error rates distribution of the individual 200 detectors for no false positive error.

Distribution of the false negatives error rates — MM270k image collection

Figure 5.6 shows the FNs error rates histogram for no false positive error on the MM270k image collection and the Qamra and StirMark benchmarks. Additionally, similar histograms are given for the systems based on the L_1 and L_2 metrics. For example, figure 5.6a indicates that over ninety percent of the SVC-based detectors have FNs error rates around 0%, and five percent of the detectors have FNs error rates of 30% or above.

The ten detectors that have FNs error rates above 30% correspond to originals for which near-duplicates exist in the image collection, as already mentioned in section 5.4.1. This is illustrated in figure 5.7. Finally, figure 5.6b gives a more precise idea of the FNs error rates distribution. Indeed, about thirty percent of the SVC-based detectors achieves no false negative, or in other words no error at all, and about thirty-five percent reaches false negatives error rates around 1%. This signifies that the number of perfect detectors is six times higher for the SVC-based system than for the L_1 -based system.

Distribution of the false negatives error rates — CGFA image collection

Figure 5.8 shows the FNs error rates histograms for no false positive error on the CGFA image collection and the Qamra and StirMark benchmarks. For example, figure 5.8a indicates that over ninety-four of the SVC-based detectors have FNs error rates around 0%, and no detector have FNs error rates of 30% or above.

The performance obtained with the CGFA collection contrasts with the results obtained for the MM270k collection and indicates that the CGFA collection does not contain duplicates or near-duplicates. This is quite significant as the results' analysis is thus not blurred by them. For instance, the examinations of the number of perfect detectors based on the L_1 metric shows that the CGFA collection is more difficult than the MM270k collection. Indeed, while there are about five percent of perfect detectors based on the L_1 metric for the MM270k collection, there are

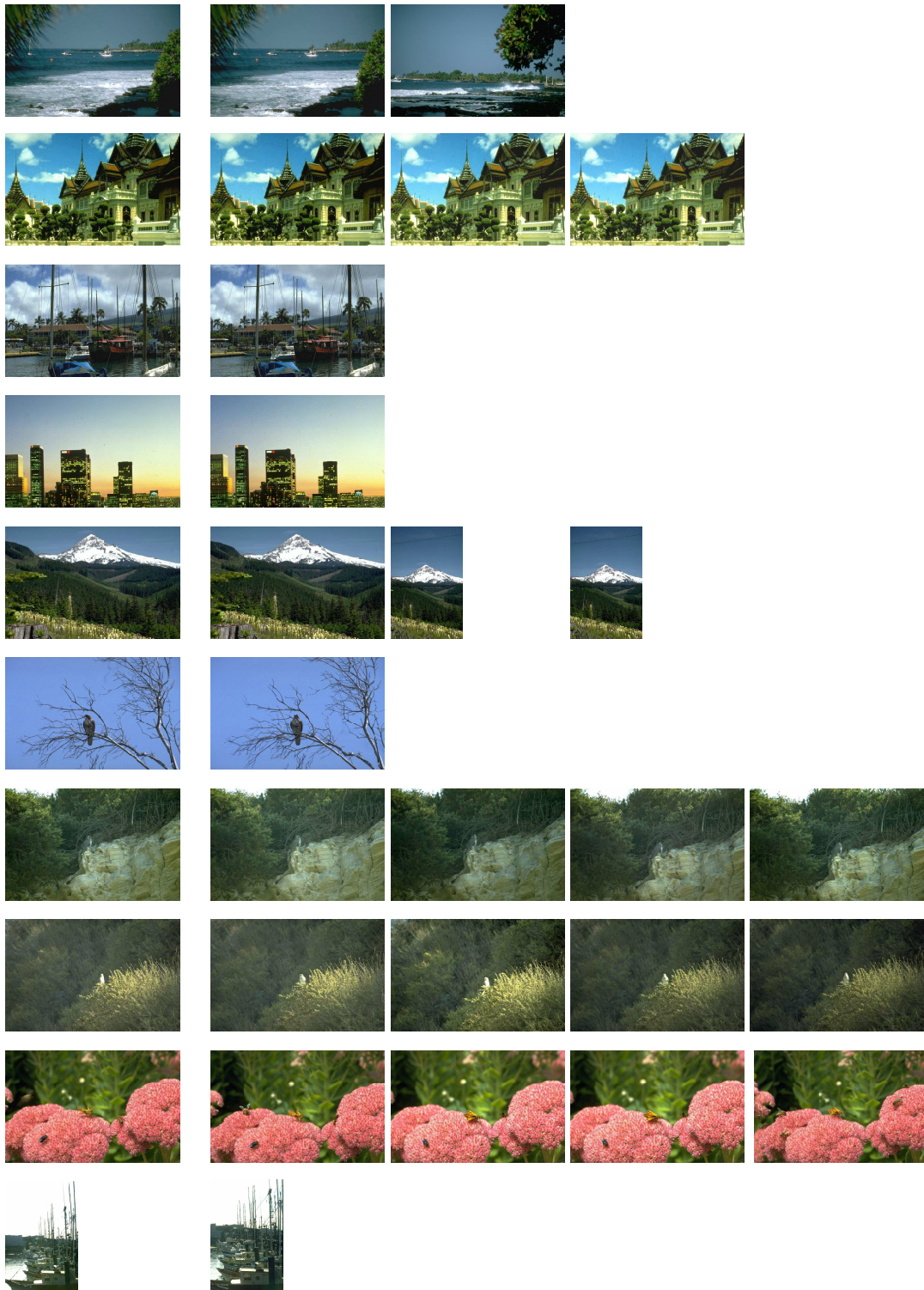


Figure 5.7: *Near-duplicate in MM270k*. This figure shows that the MM270k image collection contains many near-duplicate, photographs from the same scene taken at different locations and at different time. The first column on the left depicts ten images for which duplicate detectors have been trained. The remaining columns shows the images, taken from the MM270k collection, that the detectors find most likely to be duplicates of the respective original images.

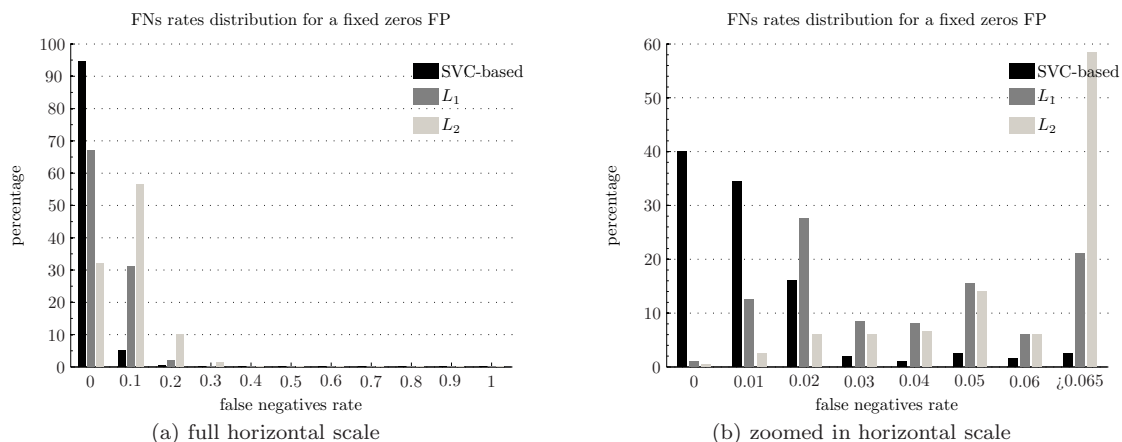


Figure 5.8: *FNs error rates distribution for CGFA image collection.* This figure depicts the FNs error rates distribution of the individual 200 detectors for no false positive error.

less than two percent of them for the CGFA collection. Additionally, the same observation can be made for detectors having FNs error rates around 1%. On the other hand, the percentage of perfect detectors based on the SVC is higher for the CGFA collection than for the MM270k. This result clearly demonstrates the adaptability of the proposed duplicate detection approach.

Now, the six duplicate detectors that give the highest FNs error rates for no false positive error are shown in figure 5.9. On a total of six, three detectors corresponds to grey-level original images while the other three are for colour images. Concerning the latter, it can be observed that the unrelated image with the highest probability of being a duplicate is very similar in terms of colour, tone, and contents. This is quite as expected since the features, describing the images, are based on the colour and on the texture contents. This also suggests that, depending on the desired performance, more sophisticated features are necessary. More on this topic is developed in section 5.5. Additionally, this highlights a typical limitation of content-based duplicate detection systems, whose performance are indeed bounded by the features used to describe the images as already pointed out in section 3.2.

We now turn our attention to the duplicates that correspond, for the same six detectors, to low detection probabilities. For the grey-level images, they are mainly of two types, namely colourising and downsampling to very low resolutions. On the other hand, the transformations corresponding to low detection probabilities on colour images are more varied. They include, for example, very low quality JPEG compression, extreme cropping, or downsampling to very low resolutions.

5.4.4 Requirements on storage and computational effort

The proposed duplicate detection method requirements are now analysed in terms of storage space and computational effort.

A number of parameters are needed to compare a test image to a given original. Namely, they are the PCA projection matrix, the normalisation constants, and the support vectors of the decision functions. In the following, we refer to the aforementioned elements as the description of

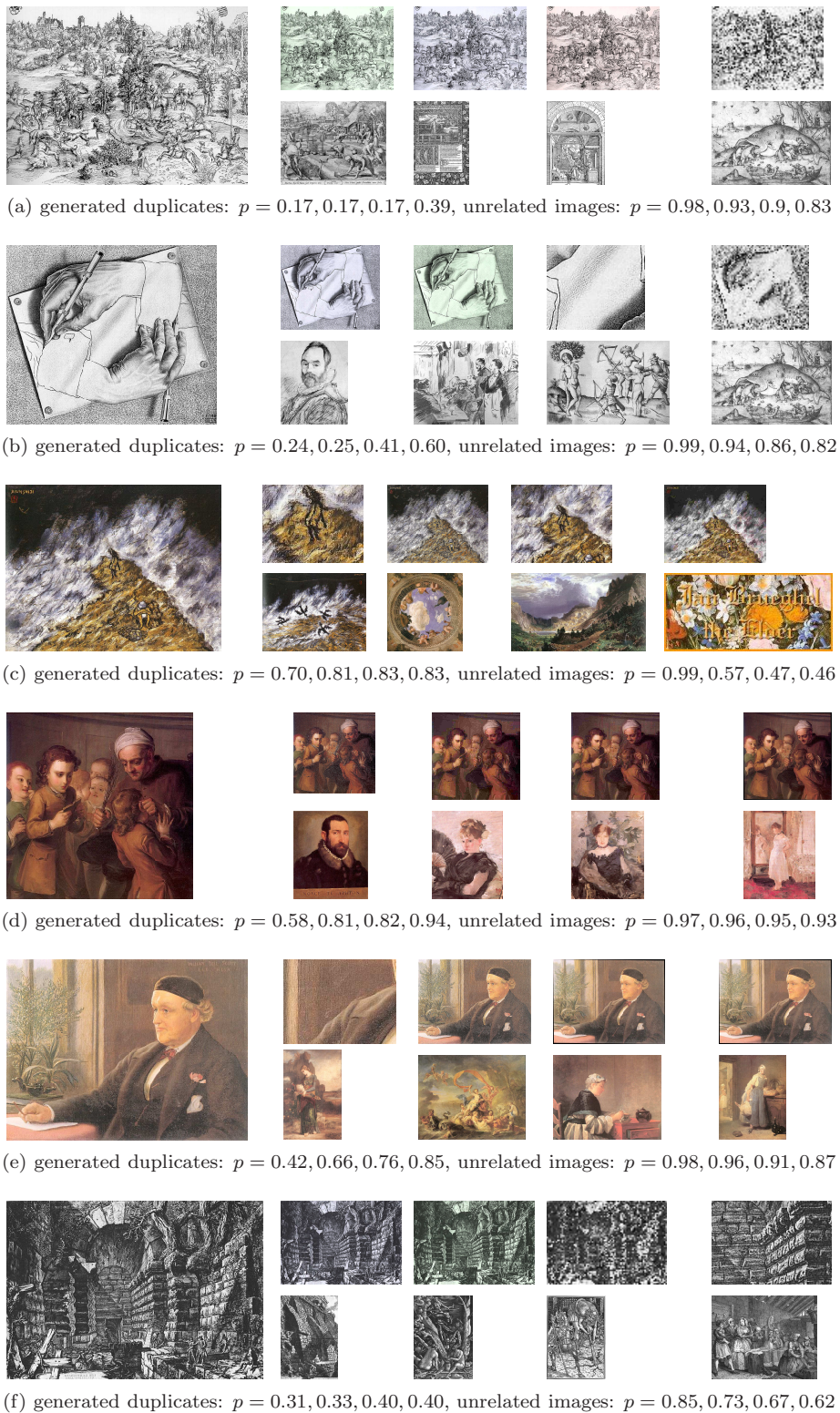


Figure 5.9: *Worst detectors for the CGFA collection.* This figure shows the six duplicate detectors that give the highest FN rates for zero false positive. For each sub-figure, the leftmost image corresponds the original, the top row represents the duplicates that obtained the lowest detection probabilities while the bottom row gives the unrelated images that achieved the highest detection probabilities.

Table 5.2: *Storage requirements estimation and average running time for testing.*

name	size, B	original image	name	time, s
PCA projection matrix	$162 \cdot 162 \cdot 2 = 52\,488$	independent	preprocessing	0.1
normalisation constants	$2 \cdot 162 \cdot 2 = 648$		feature extraction	0.5*
SVC, support vectors \mathbf{x}_i	$162 \cdot 130 \cdot 2 = 21\,060$	dependent	PCA projection	10×10^{-6}
SVC, $y_i \alpha_i$	$162 \cdot 2 = 324$		normalisation	60×10^{-6}
total	$74\,520 \leq 75\text{ kB}$		decision function	50×10^{-6}

(a) *Storage requirements estimation.* Real number are coded on 16 bits (two bytes). In our experiments, the average number of support vectors is found to be about 130.

(b) *Average running time for testing.* The experiments were carried out on a PC with a 2.8 GHz processor and 2 Go of memory.

the original image. The storage requirements are detailed in table 5.3a. On average, about 75 kB are needed to store each description. In other words, one megabyte can hold, on average, up to fourteen descriptions. This is a negligible amount of memory for today's computers.

Another important aspect is that of computational complexity of the method. The proposed method requires training for each original image. The training is computationally complex and it can, indeed, take up to ten minutes to train a detector on a PC with a 2.8 GHz processor and 2 Go of memory. Feature extraction from the synthetic duplicate examples and cross-validation to find good parameters of the SVC are the most complex parts of the training, and together take up to ninety percent of the running time. Since training can be done off-line, its computational complexity is less critical than that of testing.

The computational complexity of testing is estimated in table 5.3b. Note that except for the SVC part, the method is implemented in Matlab without any optimisation. This incurs longer running time. For instance, the feature extraction could be reduced to, at least, 0.1 seconds [Qamra *et al.*, 2005]. In the discussion that follows, we assume an optimised feature extraction step. The preprocessing and feature extraction steps are independent of the original image, and take about 0.2 seconds. On the other hand, the remaining steps depend on the original image, they take about 0.1×10^{-3} seconds per detector.

Let us consider the following scenario. A company is checking images circulating on the Internet to see whether they contain duplicates of original images for which it holds copyright. In this scenario, the company has to test an image with different detectors. When the number of owned original images is less than 200, most of the testing time is spent on preprocessing and extracting features from the test images. In that case, up to five test images can be processed per second and per computer. For a larger number of original images, most of the testing time is spent on the original image dependent steps. The number of test images that can be processed per second decreases linearly as the number of original images grows. Chapter 6 concentrates on pruning the original images, in order to avoid testing them all. That is, only the original images for which the test image can be potentially a duplicate are selected. Such methods can reduce the testing time, and have been applied with success in [Ke *et al.*, 2004; Qamra *et al.*, 2005].

5.4.5 Comparison with existing duplicate detection methods

We now compare the performance of the proposed method with that of existing duplicate detection systems. The choice of these systems is not easy since, as remarked in chapter 3, no standardised benchmark exists and the testing methodology is often not clearly given. For this reason, systems for which the set is clearly defined are chosen, namely [Ke *et al.*, 2004] and [Qamra *et al.*, 2005]. By clearly, we mean that either the image collection is given, or that the transformations used to define the duplicates are given, together with their parameters. For instance, Ke *et al.* made their image collections available, namely MM270k and CGFA, and use the transformations proposed in Qamra *et al.*, namely Qamra. Unfortunately, none of these two publications give results for the StirMark benchmark and, consequently, comparisons are only for the Qamra benchmark.

Comparison — from precision versus recall to FPs error rates versus FNs error rates

Both [Qamra *et al.*, 2005] and [Ke *et al.*, 2004] methods are set in the image retrieval framework and, therefore, give their result in terms of precision versus recall measurements. It is, however, possible to translate a precision-recall curve into a DET curve. Indeed, the first term in the right-hand side of equation (5.1) is equal to the recall, and permits to trivially compute the FNs error rate. Similarly, since the second term in the right-hand side of equation (5.1) is equal to the precision, it is also straightforward to determine the false positives rate given the ratio ρ and the previously computed false negatives rate.

Accordingly, the DET curve for the DPF method is obtained by inspecting the precision-recall curve reported in figure 5 of [Qamra *et al.*, 2005] and using $\rho = 40\,000/40$. However, Qamra *et al.* do not use the same image collection. Consequently, they are not subject to the near-duplicate problem encountered for the MM270k collection, as detailed in section 5.4.3. This means that their estimated performance are somewhat inflated for this particular collection.

Similarly, the point for KPs method is computed using the information reported in Table 1 and Table 2 of [Ke *et al.*, 2004] and $\rho = 18\,722/40$ for the MM270k collection and $\rho = 12\,000/40$ for the CGFA collection. Note that strangely enough Ke *et al.* do not give performance for the MM270k collection and the Qamra benchmark. They, however, give a result for this collection and another, non-standard, benchmark. From this, one can deduce the false positives error rate obtained on the MM270k collection. On their system, the number of false positives actually depends only on the threshold on the number of matching key points, which is the same for the two experiments. Finally, the false negatives error rate is approximated from the results obtained for the CGFA collection on the Qamra benchmark. Note that this likely inflates their performance since their system performs much better on the CGFA benchmark than on the MM270k one.

Comparison — results and analyse

Figure 5.10 compares the performance of the proposed duplicate detection system with state of the arts techniques reported in [Ke *et al.*, 2004; Qamra *et al.*, 2005]. The black line corresponds to the vertically averaged DET curve obtained with our system. The light grey line represents the estimated performance of a duplicate detection method based on perceptual distance function

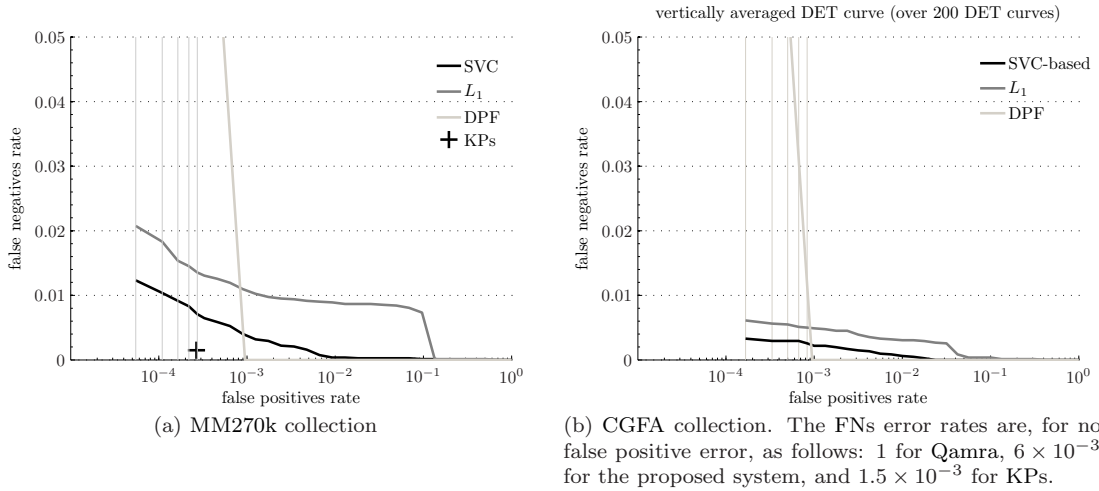


Figure 5.10: *Comparison with state of the art methods.* The proposed system is compared with two state of the art system, namely KPs [Ke *et al.*, 2004] and DPF [Qamra *et al.*, 2005]. The test are carried out on two different image collections, MM270k and CGFA, and the Qamra benchmark is used to generate the test duplicates. To keep a point of comparison, the performance of the L_1 system is also given.

(DPF) [Qamra *et al.*, 2005]. The cross indicates the performance of a duplicate detection system based on key points (KPs) [Ke *et al.*, 2004].

It can be seen that the proposed method achieves quite a good performance. For instance, on the CGFA collection, an average FNs error rate of 1×10^{-3} corresponds to a fixed false positive error rate of 1×10^{-3} . On the other hand, on the MM270k collection, an average FNs error rate of 2×10^{-3} corresponds to a fixed false positive error rate of 2×10^{-3} . This is not as good than on the CGFA collection because the MM270k collection contains near-duplicate as explained in section 5.4.3.

Now, comparing the performance of the DPF method with that of the proposed system two things can be observed. First, the DPF method achieves no FN error for false positives error rates above 1×10^{-3} . However, once below that point the performance degrades extremely rapidly. Recall that the DPF method consists in a metric where only the most similar entries in two vectors are used to computer the distance between them. While this improves the chance that duplicates are closer to the original, it similarly increases the probability that unrelated images become closer. This explains the sudden FNs error rates increase when the FPs error rates go below 1×10^{-3} . Second, while DPF performs somewhat better than the proposed system for false positive error rates above 1×10^{-3} , it is clearly outclassed below that threshold. Moreover, it should be noted that the features used in the current work are mainly a subset of those used in DPF: we use 162 features against 298 in the latter study, refer to section 4.4.2 for more details. This signifies that adapting the metric to each original image, as done in this thesis, brings tremendous increases in performance for the same image description.

On the other hand, the proposed method is outperformed by KPs. Indeed, on the CGFA collection, KPs achieves a FNs error rate of 1.5×10^{-3} for no false positive error. On the other

hand, the proposed method reaches, for the same test set, a false negatives error rate of 6×10^{-3} for also no false positive error. Additionally, the performance gap is slightly larger for the MM270k collection but these results are less significant since they are extrapolated for the KPs method and possibly the estimated performance of KPs is inflated. A possible explanation is as follows. In our method, most of the wrongly classified duplicates, false negatives errors, correspond to duplicates for which the illumination, or the intensity, has been changed to a great extent. The KPs method uses features invariant to this change but computationally more complex to extract, namely salient points [Lowe, 2004] and refer to section 2.1.4 and section 3.3.1 for more and information. Indeed, the feature extraction time of KPs is, depending on the image, between one and ten seconds per image [Ke *et al.*, 2004; Qamra *et al.*, 2005]. This is between five to fifty times, again depending on the image, slower than that for the proposed method. The fact that the extraction of key points is slower than the extraction of features, as used in the proposed system, can be of paramount importance for applications where many images have to be tested per seconds. Indeed, the proposed system require between five and fifty less computational resources. It would be interesting to build a duplicate detection system based on a fast and approximated version of Lowe method. For example Grabner *et al.* achieves a speedup in the order of eight to ten with respect to the original [Grabner *et al.*, 2006]. Their approximation is based on the very successful integral image algorithm [Crow, 1984; Viola and Jones, 2001]. However, since the approximation is quite severe it means that the resulting duplicate detector could perform quite poorly.

5.5 Exploratory works

In this section, we present possible research directions related to the topic of this chapter. Most of the proposal concerns performance improvement. Additionally, results are given whenever preliminary experiments have been run.

5.5.1 Optimal training examples

The choice of the training examples is an important topic that is not fully treated in this thesis. In this section we carry out an experiment, namely almost doubling the number of duplicate examples by more finely sampling the parameterisable transformations. The resulting duplicate training examples are given in table 5.3. Compare this with the examples given previously in table 5.1. This new training set contains 200 duplicates instead of 106 previously.

Figure 5.11 shows the performance for the previous training set, and that of the new one. Additionally, Figure 5.11b demonstrates that using the new training examples decreases by more than fifty percent the FNs error rates across the entire range of FPs error rates.

This preliminary experiment opens a very interesting research direction for duplicate detection, namely that of selecting the optimal set training examples. A possible solution stems from considering the duplicate model developed in section 4.1. Indeed, good training examples should specify as much as possible the subspace spanned by the duplicates. Consequently, a good training set is one that samples as evenly as possible this subspace. An explicit formulation for this sampling is certainly very difficult since the subspace is, in the simplest case, a manifold. However, a first

Table 5.3: *New duplicates examples for training.* This training set is the same than the one given in table 5.1 but the parameterisable transformations are more finely sampled. It contains 200 duplicates instead of 116 previously.

categories	#	parameterisations
Colourising	36	Tint the red, green, or blue channel from -11% to +11% by steps of 2%
Contrast changes	2	Increase or decrease the contrast ^a
Despeckling	1	Apply ImageMagick's despeckling operation
Downsampling	10	Downsample by 3% to 93% by steps of 10%
Colour depth reduction	1	Reduce the colour palette to 256 colours
Saturation changes	15	Change the values of the saturation channel by -22% to +22% by steps of 3%
Intensity changes	15	Change the intensity with the same parameters than for saturation
JPEG compression	19	JPEG compression with quality factors from 8 to 98 by steps of 5
Shearing	9	shearing in (X, Y) directions with X and Y varying from 0 to 6 by steps of 2
Cropping	19	centred cropping from 2% to 94% by steps of 5%
Flipping	1	horizontal flip
Scaling	8	scaling by factors from 0.45 to 0.95 by steps of 0.1
Aspect ratio	10	change aspect ratio of X(Y) by a factor from 0.75 to 1.25 by steps of 0.1
Rotation	22	rotations by angles from 4° to 92° by steps of 4°
Rotation/scaling	22	same as above but followed by scaling
Linear transform	6	general linear geometric transformation $\mathbf{c}' = \mathbf{T}\mathbf{c}$ ^b
FMLR	3	frequency mode Laplacian removal attack with parameters 0.02, 0.04, and 0.06
Grey-level conversion	1	
total	200	

^ausing ImageMagick's [Still, 2005] default parameter

^bsame matrices as for testing but with entries multiplied by 0.99 and 1.01.

approach can be to evenly sample the curve defined by a single transformation. This is much easier since the length of the curve can be easily approximated. Then another question is how to proceed when a transformation is controlled by more than a single parameter? Finally, another interesting question is whether the optimal training set depends on the image or if a single set, adequate for most original images, can be determined.

5.5.2 Combining classifiers

This proposal is based on the results displayed in figure 5.9. It can be observed that in some cases the unrelated images, assigned high probabilities by the detector, possess similar tones and colour than the corresponding original but in different quantities. Ideally this should not happen because all the necessary information is present in the features describing the images. However, the SVC does not capture this fact. This, most probably, happens because of the limited number of training examples, for instance, absence of unrelated training images behaving as mentioned above.

A possible way of improving this flaw is to incorporate more training examples. Another

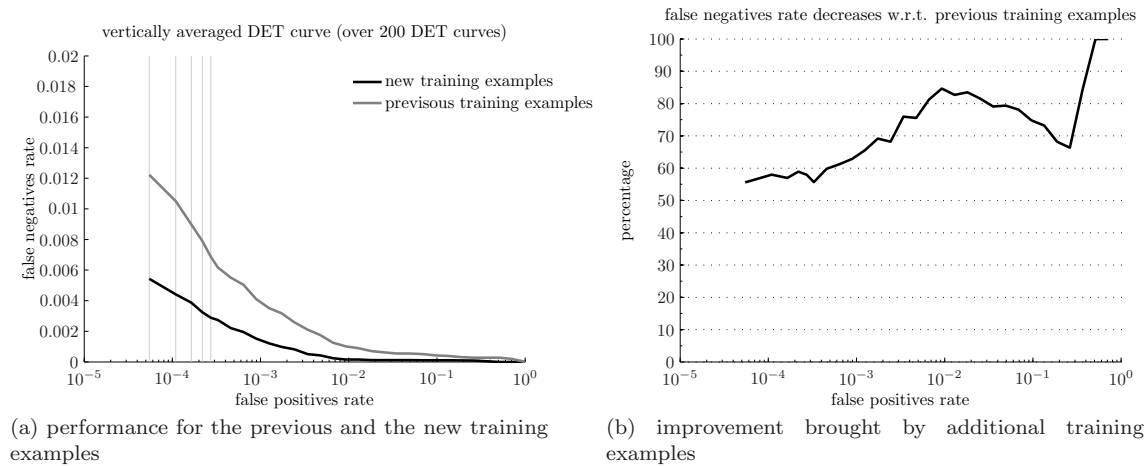


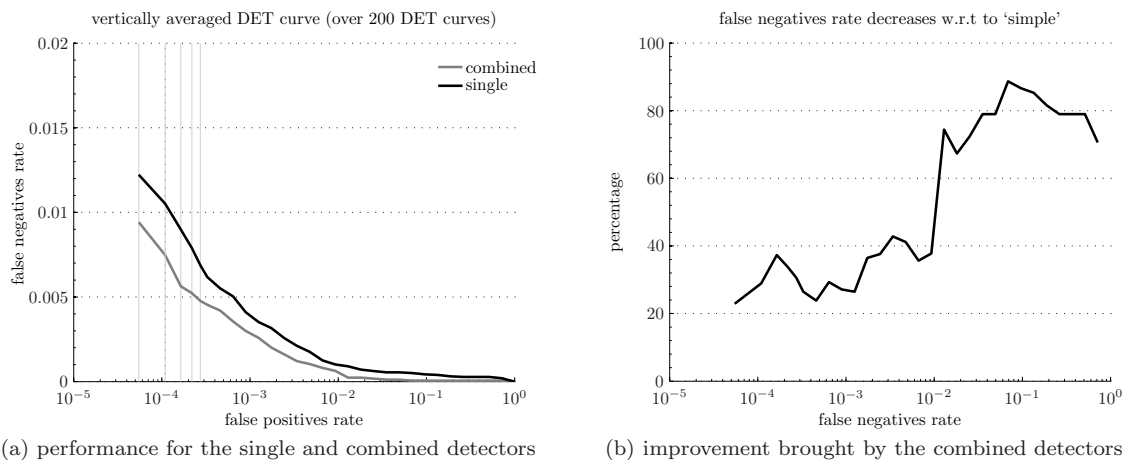
Figure 5.11: *Performance improvement brought by more training examples.* The training previous training examples are given in table 5.1 and the new training examples are given in table 5.3.

possible solution is to create another classifier, one that just takes into account the quantity of each colour in an image. We tried this approach, and extracted from the feature vectors, the entries that represent the quantity of each colour and grey-level contained within the image. Using the exact same procedure than for the entire features vector, a duplicate detector is so created. Alone, this duplicate detector performs quite poorly. When combined with the detector using the entire vector, however, the performance is greatly improved. To illustrate this point, let's consider the performance obtained for the MM270k image collection and for duplicates generated by the Qamra and StirMark benchmarks. They are depicted in figure 5.12. Both detectors output a detection probability, which are combined as follows

$$p_c = 1 - \frac{\sqrt{(1-p_1)^2 + w \cdot (1-p_2)^2}}{\sqrt{1+w}}, \quad (5.14)$$

where $p_{1,2}$ are the probabilities given by the single detectors and w is a positive number used to give more weights to one of the detectors. Note that this combination is in fact inversely proportional to the distance to the point $(1,1)$. In the experiments, $w = 1/2$, p_1 corresponds to the detector using all features while p_2 is for the simpler detector. The particular choice of w was motivated by the fact that p_1 results from a better performing detector than p_2 . Figure 5.12a shows the performance for the single detector, and that of the combination. Additionally, Figure 5.12b demonstrates that the combination decreases by about thirty percent the FN rates below 1×10^{-2} and up to eighty percent above this threshold. Indeed, for FP rates above 1×10^{-2} , the combined detector is able to achieve almost no FN errors.

This preliminary experiment opens a very interesting research direction for duplicate detection, namely that of classifiers combinations [Breiman, 1996]. Many questions have to be answered. For instance, what is the best way to combine different detectors? How to select the features subset? Is it necessary to have a detector that uses all the features or is it enough to combine only simple



(a) performance for the single and combined detectors

(b) improvement brought by the combined detectors

Figure 5.12: *Performance improvement brought by detector combination.* The single detector use all the available features, it is then combined with a simpler detector that uses only features related to the quantity of each colour, and grey-level, presents within an image.

detectors?

5.6 Chapter summary

In this chapter we presented a duplicate detection system based on a support vector classifier (SVC). The performance of the proposed system is then analysed and compared with state of the art methods. Finally, possible research directions are explored.

The proposed system is composed of the four steps outlined thereafter. In the first step, described in the previous chapter, global statistics are used to describe the image. In the second step, the features are linearly transformed so as to obtain a better separation between duplicates of the original image and unrelated images. In the third step, the elements of projected feature are normalised according to the statistical distribution of the duplicates. In the last step, a non-linear decision function, based on SVC, is used to determine the probability that the test image is a duplicate of the original image.

The performance of the proposed system is assessed, using standard benchmarks, and the result is analysed. It is found out that the proposed SVC-based duplicate detector greatly outperforms detectors using the same features but based on the L_1 metric. Additionally, the system is compared to existing state of the art methods. More precisely, it outperforms the DPF method, which uses more feature to describe the image. While slightly outperformed by the KPs method, the proposed method is five to ten times less computationally complex.

Finally, the performance of the proposed system can be greatly improved by using better training example, or by combining simpler classifiers. However, these two avenues of research necessitate further works.

Multiple Original Images Duplicate Detection System

6

In this chapter, we detail our approach to image duplicate detection of multiple original images. The approach is partially based on previous works [Maret *et al.*, 2006b,c]. The main idea behind the proposed system is to create a binary duplicate detector, as developed previously in chapter 5, and then to efficiently combine them together. The system is then able to classify a test image as duplicates of one of the original images or as an unrelated image. The main contribution of this chapter is the pre-classifier proposed to prune the images known to the system, which avoids to use every binary detector with every test image.

The approach is first motivated in section 6.1. Then, an overview of the system is given in section 6.2. Some remarks on training are given in section 6.3. Then, the pre-classifier's algorithm is thoroughly described in section 6.4 and the corresponding results are reported in section 6.5. The analysis of the entire system performances is given in section 6.6. Possible research directions are finally proposed in section 6.7.

6.1 Approach motivation

The system presented in this chapter aims at detecting duplicates of one of the many original images known to the system whereas the system presented previously, in chapter 5, knew only a single original. For more information on the differences between the two approaches, the reader is referred back to chapter 4 and chapter 5.

The main idea behind the proposed multiple original images duplicate detection system is to use a binary detectors, as developed in chapter 5, per original image. The combination of their results is then used to determine whether a test image is a duplicate of one of original images known to the system. Note that the number of original can be fairly large depending on the application, for example in the thousands or even millions. The problem is then as follows. When using a set

Algorithm 1 Multiple images duplicate detection**Require:** test image \mathbf{T} , trained pre-classifier and set of binary detectors**Ensure:** label l of most probable original, and the corresponding probability p

```

1: procedure DETERMINE_ORIGINAL( $\mathbf{T}$ )
2:    $\mathbf{f} = \text{FEATURE\_EXTRACTION}(\mathbf{T})$   $\triangleright$  step 1 — feature extraction
3:    $\mathcal{C} = \text{PRE\_CLASSIFIER}(\mathbf{f})$   $\triangleright$  step 2 — set of candidates
4:   for  $i = 1$  to  $|\mathcal{C}|$  do
5:      $m = \mathcal{C}_i$ 
6:      $p_i = \text{BINARY\_DETECTOR}_m(\mathbf{f})$   $\triangleright$  step 3 — binary detectors
7:   if  $|\mathcal{C}| > 0$  then
8:      $m = \arg \max_i p_i$   $\triangleright$  step 4 — most probable original
9:      $l = \mathcal{C}_m$ 
10:     $p = p_m$ 
11:    return  $(l, p)$ 
12:  else
13:    return  $(-1, 1)$ 

```

of binary detectors, each tuned to a specific original image, a test image need to be sequentially checked with each binary detector. Unfortunately, this procedure becomes quickly cumbersome as the number of original images grows. Therefore, we propose to use a pruning step based on an indexing structure, where the most likely original images are efficiently selected and the remaining originals are discarded. We call candidates the most likely original images. Ideally, the set of candidates contains a single element if the test image is indeed a duplicate and none otherwise. Nonetheless, a more realistic goal is to have a set whose size is a fixed fraction of the total number of original images.

6.2 System overview

We now give the gist of the proposed multiple original images duplicate detection system. The system consists of four steps as shown in figure 4.6, each of them is outlined thereafter. Algorithm 1 gives the pseudo-code of the system's mechanics. Recall that the system's goal is to determine whether a test image is a duplicate of one of the original images known to the system, each labelled from 1 to N . The algorithm thus returns the estimated label $l \in \{-1, +1, \dots, N\}$ and the corresponding probability p . An estimated label of -1 signifies that the test image has been discarded by the pre-classifier and means that the test image is considered unrelated to any of the originals. As already mentioned in section 4.2.2, a decision can then be obtained by comparing the estimated probability p to a fixed threshold u . If the probability is larger than the threshold then the estimated original is the one given by the label while otherwise the test image is considered unrelated to any of the original images.

Preprocessing and feature extraction The first step consists in preprocessing the image and then in extracting descriptive features from the preprocessed image. Preprocessing and feature extraction operations are both identical to those used for the binary detectors, and are thoroughly described in section 4.4.1 and section 4.4.2, respectively.

Pre-classifier The second step, in fact the main contribution of this chapter, aims at efficiently selecting a limited number of potential originals among all the original images known to the system. More precisely, we denote by \mathcal{C} the set of candidates. Since the set of original images is given by \mathcal{O} , \mathcal{C} is a subset of \mathcal{O} . Ideally, \mathcal{C} contains only few elements and includes the correct original if the test image is indeed a duplicate of one of the originals. The pre-classifier is built around an indexing structure. More precisely, an estimate of the subspace spanned by the duplicates, see section 4.1 for more details, is indexed for each original.

Binary detectors In the third step, the binary detectors developed in chapter 5 are used to order the elements within the set of candidates from the most probable to the least probable original. More precisely, the probabilities p_i that the test image is a duplicate of the originals \mathcal{C}_i are estimated. Finally, the elements of the set of candidates are sorted according to the corresponding probabilities.

Decision The last step selects the most probable original and also outputs the corresponding probability.

6.3 Remarks on training

The pre-classification step needs training, namely the estimated subspaces have to be indexed. The training is performed independently on each original. This means that new original images can be added without retraining the original images already indexed within the pre-classifier.

The training procedure requires only positive examples to index the estimate of the subspace spanned by the duplicates. However, both positive and negative examples are needed to evaluate the resulting indexation. To achieve this, the same training examples as for the binary detector are used. More precisely, the 200 positive examples are given in table 5.3 while the 500 negative examples are obtained by randomly selecting examples from the image collection. For more information, the reader is referred back to section 5.2 and section 5.5.1.

6.4 Pre-classifier

In this section, we detail the proposed pre-classifier and the corresponding training procedure. The pre-classifier's pseudo-code is given in algorithm 2. The pre-classifier algorithm returns the set candidates \mathcal{C} associated to the test image \mathbf{I} . Additionally, the pre-classifier algorithm is parameterisable so that it selects more or less examples. This is accomplished by modifying the value denoted δ . More precisely, for δ equals to zero the procedure selects as few candidates as possible while it selects more candidates for larger values.

The pre-classifier is subdivided into three steps, namely feature extraction, feature projection, and search. The feature extraction step is the same as for the binary detectors and was already presented in section 4.4.2. The second step consists in reducing the number of features, since 162 are too many for an efficient indexation scheme, and is presented in section 6.4.1. Finally, section 6.4.2 presents an indexation scheme based on a specific indexing structure, namely R-Trees.

Algorithm 2 Finds the potential originals of a test image

Require: Originals to be indexed in the R-trees R_{tree} with algorithm 3 or algorithm 4

```

1: procedure PRE_CLASSIFY( $\mathbf{I}, \delta$ )
2:    $\mathbf{f} = \text{FEATURE\_EXTRACTION}(\mathbf{I})$ 
3:    $\tilde{\mathbf{f}} = \mathbf{W}_d \cdot \mathbf{f}$ 
4:    $\mathcal{C} = \text{SEARCH}(R_{tree}, \tilde{\mathbf{f}} \pm \delta)$ 
5:   return  $\mathcal{C}$ 

```

6.4.1 Feature projection and normalisation for indexing

Many features are needed in order to have enough information to discriminate between duplicates and non-duplicates. Nonetheless, 162 features are too many for building an efficient indexing structure. For this reason, the dimensionality of the feature vector is reduced to d by making use of PCA. Recall that the PCA algorithm finds the directions along which the scatter, or variance, of the cloud of points is maximised [Duda *et al.*, 2001]. The construction of the projection matrix \mathbf{W}_d is as follows.

1 — $\tilde{\mathbf{W}}$: The PCA algorithm is applied to a training set containing the features of original images, for more details the reader is referred back to section 5.3.1, and results in a projection matrix $\tilde{\mathbf{W}}$. Then the projected features are given by $\tilde{\mathbf{W}} \cdot \mathbf{f}$.

2 — $\tilde{\mathbf{W}}_d$: The PCA produces a 162×162 projection matrix. In other words, the number of dimensions of the projected features equals that of the extracted features. To reduce the number of dimensions from 162 to d , the rows of the matrix $\tilde{\mathbf{W}}$ are first ranked from the the direction having the largest variance to the direction having the smallest variance. Subsequently, the d first rows of the projection matrix are selected, resulting in a $d \times 162$ matrix. This results in a projection matrix $\tilde{\mathbf{W}}_d$. This choice is motivated as follows. The direction for which the scatter is maximal corresponds also to the direction along which the average distance between the points is maximal. By selecting the d largest scatters, we indirectly select the d directions that, independently, best separate the points.

3 — \mathbf{W}_d : Finally, the projected features are normalised so that the variance along each projection direction is equal to one. This normalisation can be directly incorporated in the projection matrix by scaling each row accordingly.

We experimentally found out that PCA gives better results than ICA-fx for this purpose. Recall that ICA-fx [Kwak and Choi, 2003] is a linear dimensionality reduction technique adapted to classification problem. Indeed, if all remaining parameters are kept the same, a pre-classifier built on features given by PCA returns, on average, two to ten times less candidates than one constructed using features derived by ICA-fx [Maret *et al.*, 2006c]. A possible reasoning for this is the following. A good projection should separate, as much as possible, the clusters of feature vectors representing the duplicates of each original in the database. With ICA-fx this separation works well for the originals used for training since the algorithm maximises the separability of the corresponding classes. However, no guarantee is provided for other original images. On the other

hand, PCA reduces the dimensionality of the feature space by finding the directions along which the scatter of the cloud of points is maximised. These directions are therefore not linked to a particular classification problem, thus leading to a representation of the data that works well for the pre-classification task.

6.4.2 R-Tree indexing

The chosen indexing structure is based on R-trees [Guttman, 1984], which are dynamic structures used to efficiently index high-dimensional spaces. An R-tree is a height-balanced tree with index records in its leaf nodes, containing pointers to data objects. Originally, R-trees were created to index spatial objects using their bounding boxes. Therefore, the R-tree structure is constructed so as to efficiently answer the point-based query “Return all records whose bounding boxes include the search point \mathbf{p} ,” and the box-based query “Return all records whose bounding boxes intersect the search box \mathbf{b} .”

The choice of creating a pre-classifier around an indexing structure working on bounding boxes is motivated as follows. Firstly, bounding boxes can be efficiently determined. Secondly, a bounding box is very flexible since each of its side can be independently adjusted. Finally, the indexing algorithms given thereafter can be readily adapted to more complex indexing structures such as M-Trees [Ciaccia *et al.*, 1997].

The two next sections introduce two distinct indexation schemes. More precisely, the first scheme uses a single bounding box per original image, we call it coarse indexation, while the second scheme builds on the first and uses multiple bounding boxes per original, we call it fine indexation. The coarse and fine indexation schemes are used exactly the same way in algorithm 2 but differ on how an original is indexed or, in other words, the training procedure is different.

Coarse indexation scheme

Since the features extracted from images exhibit a certain degree of robustness against image manipulations, the features of a duplicate are localised around those of the corresponding original image. Therefore, an R-tree, optimised for duplicate detection, can be constructed by associating a bounding box, encompassing all duplicate examples, with each original image known to the system. In fact, since we are dealing with a d -dimensional space, the bounding boxes are d -dimensional orthotopes, or generalised rectangular parallelepiped. The choice of these bounding boxes is critical for the performance of the R-tree. Indeed, if the bounding boxes are too large, many of them overlap. This results in a large number of elements in \mathcal{C} . On the other hand, if the bounding boxes are too small, a duplicate can fall outside the bounding box corresponding to its original, which is thus not included in \mathcal{C} .

In order to construct the bounding box associated with an original image, we generate duplicate examples by making use of a set of image manipulations. More precisely, the bounding box is defined by two vectors \mathbf{c}_- and \mathbf{c}_+ , which control its extent in each dimension

$$\mathbf{c}_-(\alpha) = \min_i \mathbf{f}_i(\alpha), \quad (6.1)$$

$$\mathbf{c}_+(\alpha) = \max_i \mathbf{f}_i(\alpha), \quad (6.2)$$

Algorithm 3 Coarse indexation

Require: the original image \mathbf{I} , its identifier ID , the parameter $\delta \in [-1, +1]$, and the duplicate examples $\{\mathbf{D}\}_{i=1}^D$

Ensure: the R-tree contains the duplicate region estimation for the original

```

1: procedure COARSE_INDEXATION( $\mathbf{I}$ ,  $ID$ ,  $\delta$ ,  $\{\mathbf{D}\}_{i=1}^D$ )
2:   for  $i = 1$  to  $D$  do
3:      $\mathbf{f}_i = \text{FEATURE\_EXTRACION}(\mathbf{D}_i)$ 
4:      $\mathbf{c}_- = [\min_{i=1, \dots, D} \mathbf{f}_i(\alpha)]_{\alpha=1}^d$ 
5:      $\mathbf{c}_+ = [\max_{i=1, \dots, D} \mathbf{f}_i(\alpha)]_{\alpha=1}^d$ 
6:      $\mathbf{s} = \mathbf{c}_+ - \mathbf{c}_-$  ▷ compute the side lengths  $s(\alpha)$  of the bounding box
7:     INSERT( $R_{tree}$ ,  $\mathbf{c}_\pm \pm \delta \cdot \mathbf{s}/2$ ,  $ID$ ) ▷ algorithm INSERT in [Guttman, 1984]

```

where the $\mathbf{f}_i(\alpha)$ correspond to the α -th feature of the i -th duplicate example, and the $\mathbf{c}(\alpha)$ denotes the α -th element of the vectors \mathbf{c} . The examples used to compute the bounding boxes are detailed in section 6.3. Additionally, the size of the indexed box can be tuned by adding $\delta \cdot \mathbf{s}/2$ to \mathbf{c}_+ and subtracting the same amount to \mathbf{c}_- . Now, the value of δ controls the tightness of the indexed box around the duplicate examples. For instance, if δ is larger than zero, the indexed box is larger than the bounding box. Conversely, if δ is smaller than zero, the indexed box is smaller than the bounding box. The corresponding indexation procedure is given in algorithm 3.

The feature vector of a duplicate obtained by a manipulation less severe than those used to build the R-tree is expected to be contained in the bounding box corresponding to its original. Conversely, the feature vector of a duplicate generated by a more severe manipulation usually falls outside the corresponding bounding box. Nonetheless, it can still be retrieved by making use of a box-based query by using a value of delta larger than zero in algorithm 2. However, this implies a larger set of candidates.

Fine indexation

The coarse indexation scheme given previously works well for light image manipulations but fails for more severe transformations [Maret *et al.*, 2006c]. Indeed, when the modifications undergone by the image are important, the resulting feature vector will lie far from that resulting from the corresponding original image. Consequently, a pre-classifier, using the coarse indexation scheme, returns most of the original images when such difficult duplicates have to be detected. This, of course, defeats the purpose of using a pre-classifier. For this reason, we now detail a more sophisticated method to index duplicates. The basic idea behind the proposed algorithm is to index a box for each training example. Each box partially estimates the subspace spanned by the duplicates while its entire estimation is given by their union.

The size of a box is chosen such that a fixed number of the considered training example nearest neighbours are covered by it. The idea behind using the nearest neighbours is twofold. On the one hand, it creates an estimated duplicates' subspace composed of as few connected components as possible. Indeed, each box is connected to, at least, as many other boxes as the number of used nearest neighbours. On the other hand, since the number of duplicates used for training is limited, it is necessary to ensure that novel duplicates falls within one of the boxes with high probability. If the sampling of the duplicate examples generated by a single transformation is dense enough, it is likely

Algorithm 4 Fine indexation

Require: the original image \mathbf{I} , its identifier ID , the parameters $\delta \in [-1, +1]$ and k , and the duplicate examples $\{\mathbf{D}\}_{i=1}^D$

Ensure: the R-tree contains the duplicate region estimation for the original

- 1: **procedure** FINE_INDEXATION(\mathbf{I} , ID , δ , k , $\{\mathbf{D}\}_{i=1}^D$)
- 2: **for** $i = 1$ to D **do**
- 3: $\mathbf{f}_i = \text{FEATURE_EXTRACTION}(\mathbf{D}_i)$
- 4: **for** $i = 1$ to D **do** \triangleright add a box per duplicate example
- 5: $\sigma = \text{ORDER_BY_CONTENT}(\{\mathbf{f}_j\}_{j=1}^D, \mathbf{f}_i)$
- 6: $\mathbf{c}_- = [\min_{j=1, \dots, k} \mathbf{f}_{\sigma(j)}(\alpha)]_{\alpha=1}^d$
- 7: $\mathbf{c}_+ = [\max_{j=1, \dots, k} \mathbf{f}_{\sigma(j)}(\alpha)]_{\alpha=1}^d$
- 8: $\mathbf{s} = \mathbf{c}_+ - \mathbf{c}_-$ \triangleright compute the side lengths $\mathbf{s}(\alpha)$ of the bounding box
- 9: $\text{INSERT}(R_{tree}, \mathbf{c}_{\pm} \pm \delta \cdot \mathbf{s}/2, ID)$ \triangleright algorithm INSERT in [Guttman, 1984]
- 10: **procedure** ORDER_BY_CONTENT($\{\mathbf{c}_i\}_{i=1}^N, \mathbf{c}$)
- 11: **for** $i = 1$ to N **do**
- 12: $v_i = (\mathbf{c}_i - \mathbf{f})^T \cdot (\mathbf{c}_i - \mathbf{f})$
- 13: $\sigma = \text{SORT}(\{v_i\}_{i=1}^N)$ $\triangleright \sigma$ is a permutation of $1, \dots, N$ s.t. $v_{\sigma(i)} \geq v_{\sigma(i-1)}$
- 14: **return** σ

that a novel duplicate created by the same transformation falls in-between two of the generated duplicate examples. Thus, the boxes generated around these two duplicate examples are likely to include the novel duplicate, assuming that they are part of each other nearest neighbours. Clearly, the duplicate manifold estimated by the union of these boxes is likely to encompass many of the potential duplicates. Conversely, it is also important that unrelated images do not fall within the estimated manifold. This implies that the content, or higher-dimensional volume, of the partition has to be somehow minimised. For this reason, the nearest neighbours are determined by making use of the content of the box delimited by each pair of examples rather than by the conventional Euclidian metric. This measure ensures that the determined boxes are those with the minimal contents, for the given algorithm and used parameters. By extension, the estimated manifold is also the one with the minimal content, again for the given algorithm and used parameters.

The above observations lead us to devise the indexation algorithm presented in the following. More precisely, algorithm 4 describes the constructions of the subspace spanned by the duplicates for a given original image. Synthetic duplicates are first generated, and features are extracted from them and from the original. To achieve the subspace estimation, a box is created around each duplicate example. First, the nearest neighbours of the duplicate example are determined, using as measure the content delimited by each pair of examples. Then, the extremal coordinates of the k nearest neighbours are used to determine the box corners; the tuning parameter δ permits to increase the box size. Finally, the box is indexed using the INSERT procedure from [Guttman, 1984]; the used key contains the original identifier ID .

In the following, we use $k = d$ in order to decrease the probability that single training examples define more than two boundaries. Indeed, a d dimensional box is defined by its $2 \cdot d$ boundaries. For example, if $k = d - 1$ there is at least two training examples that define three boundaries, or one training example that defines four boundaries. The parameter σ , controlling the bounding boxes sizes, can be seen as a regularisation parameter, see section 2.3.1 for more information.

Indeed, a large value of σ corresponds to large bounding boxes and, consequently, to a relatively coarser estimation of the duplicate partition. On the other hand, a small value of σ corresponds to small bounding boxes and, thus, to a finer estimation of the duplicate partition. This also signifies that the smaller the value of δ , the higher the risk of overtraining. The value of δ is hence quite critical for obtaining good performance. Consequently, the value of σ is chosen through a cross-validation procedure similar to that used in section 5.3.3. In other words, σ is chosen as the one that maximises the F-score.

6.5 Results for the pre-classifier

In this section, we present experimental results in order to evaluate the performance of the proposed pre-classifier. The first experiment, presented in section 6.5.1, compares the proposed pre-classifier with a system based on a standard L_1 metric. The second experiment, described in section 6.5.2, explores the scalability of the proposed pre-classifier.

6.5.1 Baseline

In this first experiment, we compare the performance of the proposed pre-classification algorithm with that of a simpler method — based on the standard L_1 metric. More precisely, the L_1 metric is used to select the most likely originals given a test image. To achieve this, the feature vectors are first projected on a lower-dimensional space, as presented in section 6.4.1. In other words, the exact same features than for the proposed algorithm are used. The distances, based on the L_1 metric, are then computed between the projected feature vector of the test image and those corresponding to the original images. More specifically, the distance between two vectors \mathbf{x} and \mathbf{y} is given by $\sum_{\alpha} |\mathbf{x}_{\alpha} - \mathbf{y}_{\alpha}|$. Finally, the k nearest neighbour algorithm is used to select the k most likely original images. For instance, if k is set to one, the potential original is the one with the smallest L_1 distance to the features representing the test image.

Baseline — experimental setup

We now compare the proposed pre-classifier to a simpler one based on the L_1 metric, as presented previously. For this purpose, two-hundred original images are indexed using algorithm 4 as described in section 6.4. Test images, corresponding to duplicate and unrelated images, are then fed to the pre-classifier, given in algorithm 2, which is parameterised with different δ , or sizes of the search box. Each box size corresponds to a given miss rate, the fraction of test duplicates for which the set of candidates \mathcal{C} does not contain the corresponding original, and to a given hit rate, the average size of \mathcal{C} . L_1 -based pre-classifier go through the same procedure, using the same original and test images, except that the size of \mathcal{C} is always equal to the number of nearest neighbour k , we use $k = 1, 2, \dots, 5$. The best possible performances are obtained, on duplicate test images, for a miss rate equals of zero and for a hit rate of one and, on unrelated test images, for a hit rate of zero.

The performance is evaluated on two different image collections, MM270k and CGFA, which are described in more detail in section 4.3. The first collection contains 18 785 photographs while

the second collections contains photographs of 9000 artworks. Then, two benchmarks, extensively described in section 4.3, are used to test each collection. They contain the same unrelated images but differ in the duplicates' generation. The first benchmark, Qamra, contains transformations mainly based on colour modifications. On the other hand, the second test set, StirMark, contains transformation mainly based on geometric modifications.

Baseline — MM270k image collection

Figure 6.1 shows the performances obtained by the L_1 -based pre-classifier compared to those achieved by the proposed pre-classifier on the MM270k collection and the two benchmarks. The left hand side column shows the hit rate versus the miss rate for duplicate test images while the right hand side column depicts the same but for unrelated test images. Note that the figure on the unrelated images does not show the L_1 results. For the L_1 -based pre-classifier, the curves obtained for the unrelated and duplicate test images are exactly the same. Also, recall that for the L_1 -based pre-classifier, the k is nothing else than the average size of the set of candidates.

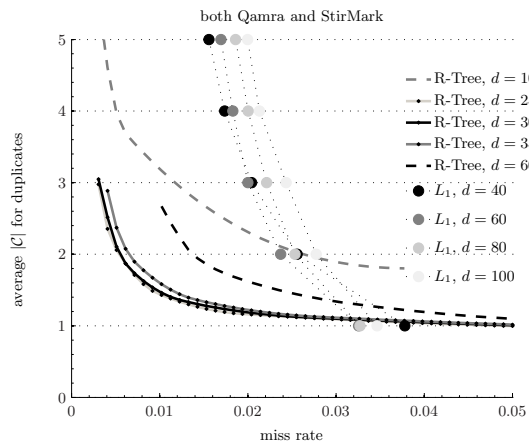
For the Qamra benchmark, figure 6.1c indicates that the L_1 pre-classifier slightly outperforms the proposed pre-classifier for $k = 1$ but is, in turn, slightly outmatched by the proposed pre-classifier for $k \geq 2$. For example, the proposed pre-classifier returns an average of 1.2 candidates for a miss rate of 0.01 while the L_1 pre-classifier needs only one candidate to achieve the same miss rate. This indicates that for light transformations, as present in the Qamra benchmark, a standard indexing scheme based on the L_1 metric is sufficient to perform well.

For the StirMark benchmark, figure 6.1e indicates that the L_1 pre-classifier performs similarly to the proposed pre-classifier for $k = 1$ but is greatly outperformed by the proposed pre-classifier for $k \geq 2$. For example, the proposed pre-classifier returns an average of 1.6 candidates for a miss rate of 0.01 while the L_1 pre-classifier would need more than five candidates to achieve the same miss rate. This indicates that for severe transformations, as present in the StirMark benchmark, a standard indexing scheme based on the L_1 metric is clearly not adapted.

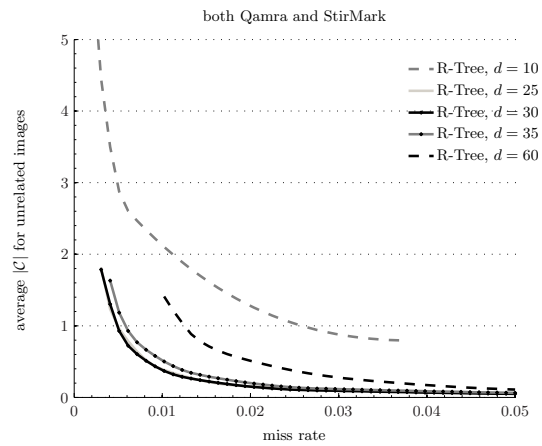
Another interesting difference between the proposed pre-classifier and the L_1 pre-classifier relates to the number of dimension d necessary to achieve the best results. While the L_1 pre-classifier necessitates $d = 60$ to achieves them, the proposed pre-classifier needs only $d = 30$.

We, now, analyse in more details the results obtained by the proposed pre-classifier. It can be observed that the hit rate increases sharply when the miss rate decreases below 0.004 for the Qamra benchmark and below 0.008 for StirMark. This behaviour can be attributed to a few images for which a limited number of modifications results in feature vectors very different than those of the corresponding original images. More precisely, these transformations are colourising of grey-level images for the Qamra benchmark and rotation and down-scaling for the StirMark benchmark.

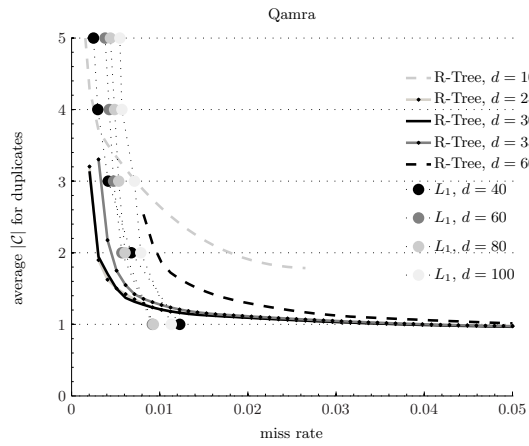
The right hand side column shows the hit rate for unrelated test images. It can be seen that the shapes of the curves are very similar to those obtained for test duplicates. The most notable difference is that the curves on the right hand side are vertically shifted down by about one with respect to those on the left hand side. This is easily explained since the set of candidates for a duplicate test image contains with high probability the corresponding original image, this is not the case for an unrelated test image. Moreover, assume that, for unrelated test images, a working



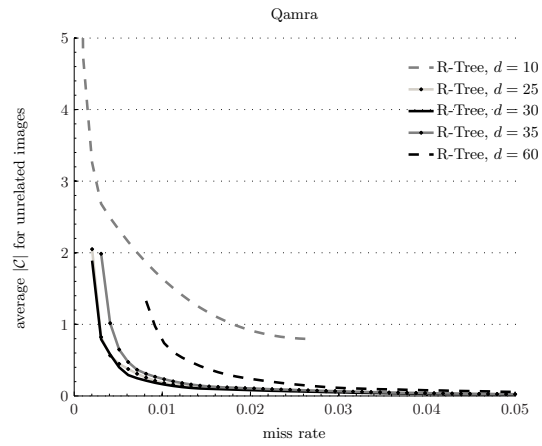
(a) duplicate test images — both Qamra and StirMark benchmarks



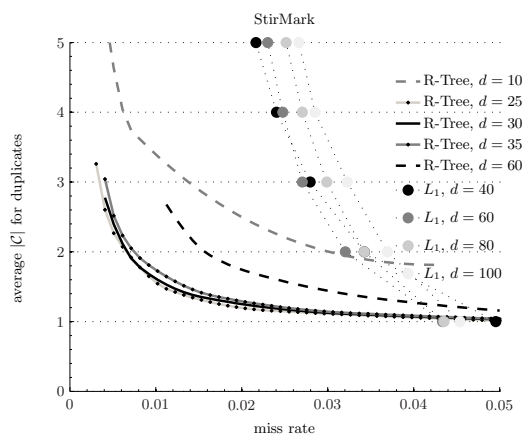
(b) unrelated test images — both Qamra and StirMark benchmarks



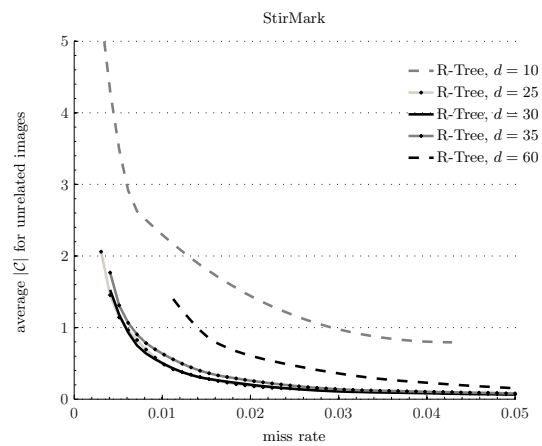
(c) duplicate test images — Qamra benchmark



(d) unrelated test images — Qamra benchmark



(e) duplicate test images — StirMark benchmark



(f) unrelated test images — StirMark benchmark

Figure 6.1: *MM270k* collection — *pre-classifier baseline*. This figure shows the performances obtained by the L_1 -based pre-classifier compared to those achieved by the proposed pre-classifier.

point on the curve is given by a miss rate of m and a hit rate of h . Now, it is possible to estimate the corresponding working point for duplicate test images: the miss rate remains the same while the hit rate is given by $1 - m + h$. Indeed, since the miss rate is m , it means that the correct original is present with a probability of $1 - m$. Additionally, if the original of the test images were not present in the index, the hit rate would behave as for an unrelated test image. This latter fact accounts for the additional h .

Finally, notice the importance of the dimension d for the performance of the proposed pre-classifier. Indeed, for $d = 10$ the pre-classifier is performing quite badly although still better than the L_1 -based one for low miss rates. This counter-performance occurs because the information given by features containing only ten values is too poor to obtain a good estimation of the subspace spanned by the duplicates. On the other hand, the proposed pre-classifier also under-performs for $d = 60$, which is more surprising at first but yet quite comprehensible. Indeed, the number of training examples is clearly insufficient to permit a fine approximation of the subspace for $d = 60$. More precisely, recall that we use $k = d$ for the number of nearest neighbour in algorithm 4. This value, while previously justified, might not be an optimal choice because it tends to create larger bounding boxes as d increases. Consequently, larger bounding boxes results in a coarser approximation of the duplicates manifold. Future research are thus necessary to discover an optimal value for k . Another possible explanation is related to the curse of dimensionality [Donoho, 1998], which implies that a classifier tends to be overtrained as the number of dimensions grows.

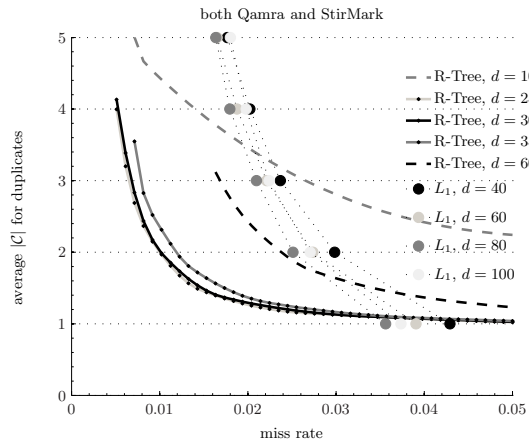
Baseline — CGFA image collection

Figure 6.2 shows the performances obtained by the L_1 -based pre-classifier compared to those achieved by the proposed pre-classifier on the CGFA collection and for the two benchmarks. The left hand side column shows the hit rate versus the miss rate for duplicate test images while the right hand side column depicts the same but for unrelated test images. These results are quite similar to those obtained previously for the MM270k collection.

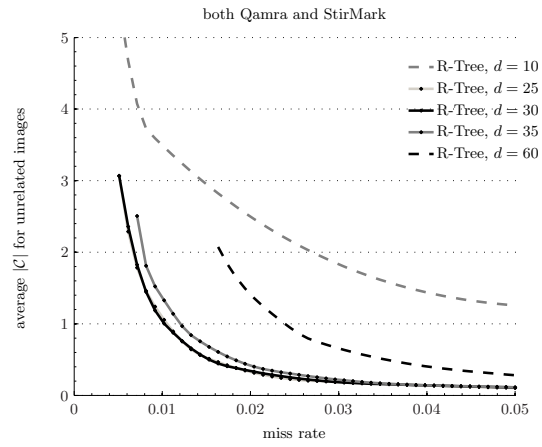
For the Qamra benchmark, figure 6.2c indicates that the L_1 pre-classifier slightly outperforms the proposed pre-classifier for $k = 1$ but is, in turn, slightly outmatched by the proposed pre-classifier for $k \geq 2$. For example, the proposed pre-classifier returns an average of 1.3 candidates for a miss rate of 0.01 while the L_1 pre-classifier needs only one candidate to achieve the same miss rate. As for the MM270k collection, this result indicates that for light transformations, as present in the Qamra benchmark, a standard indexing scheme based on the L_1 metric is sufficient to perform well.

For the StirMark benchmark, figure 6.2e indicates that the L_1 pre-classifier performs similarly to the proposed pre-classifier for $k = 1$ but is greatly outperformed by the proposed pre-classifier for $k \geq 2$. For example, the proposed pre-classifier returns an average of 2.2 candidates for a miss rate of 0.01 while the L_1 pre-classifier would need more than five candidates to achieve the same miss rate. As for the MM270k collection, this indicates that for difficult transformations, as present in the StirMark benchmark, a standard indexing scheme based on the L_1 metric is clearly not adapted.

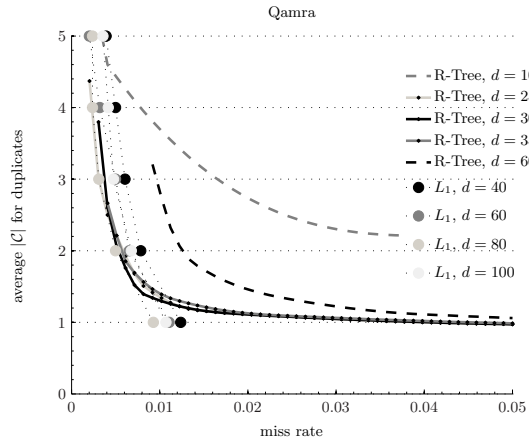
Another interesting difference between the proposed pre-classifier and the L_1 pre-classifier



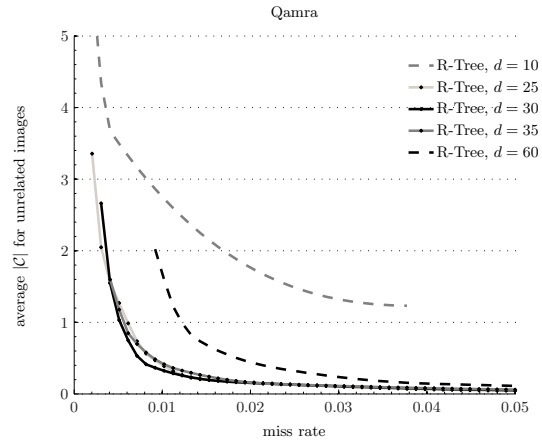
(a) duplicate test images — both Qamra and StirMark benchmarks



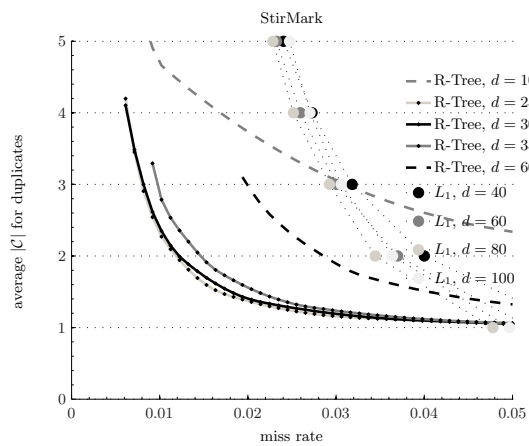
(b) unrelated test images — both Qamra and StirMark benchmarks



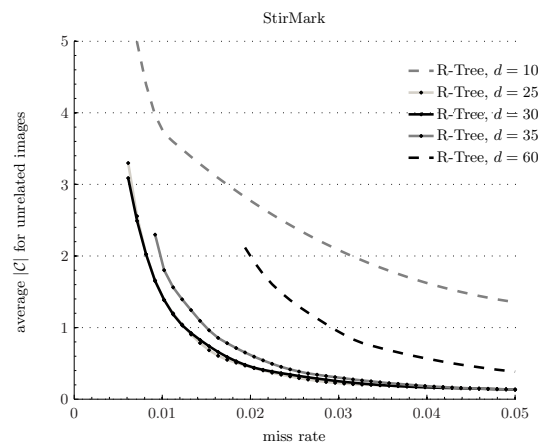
(c) duplicate test images — Qamra benchmark



(d) unrelated test images — Qamra benchmark



(e) duplicate test images — StirMark benchmark



(f) unrelated test images — StirMark benchmark

Figure 6.2: *CGFA* collection — *pre-classifier* baseline. This figure shows the performances obtained by the L_1 -based pre-classifier compared to those achieved by the proposed pre-classifier.

relates to the number of dimension d necessary to achieve the best results. While the L_1 pre-classifier needs $d = 80$ to achieves its best performances, the proposed pre-classifier needs only $d = 30$ to reach them. Additionally, notice that for the L_1 pre-classifier the number of dimensions d necessary to achieve the best performance depends on the image collection. Indeed, d is equal to 60 for the MM270k collection while it is equal to 80 for the CGFA collection. On the other hand, for the proposed pre-classifier, it remains equal to $d = 30$ for both collections.

Now, notice that the performances obtained on the CGFA collection are slightly below those obtained on the MM270k. This is as expected since, as already remarked in section 5.4.1, the CGFA collection is more difficult than the MM270k collection. Indeed, it contains very similar images: only photographs of paintings. In the previous chapter, it was also noticed that the MM270k collection contains near-duplicates images of original images. These near-duplicates influenced the results, and made the duplicate detectors perform better on the CGFA collection than on the MM270k collection. However, this effect is not noticeable here. Indeed, these near-duplicates concern only a few originals and, in the case of the pre-classifier, their influence is reduced because the size of the set of candidates \mathcal{C} is an average on all the test images whereas, in chapter 5, the average was taken only on all original images.

Baseline — conclusion

The crux of the baseline experiment is that for light transformations, as present in the Qamra benchmark, a standard indexing scheme based on the L_1 metric is sufficient. On the other hand, for more difficult transformations, such as these present in the StirMark benchmark, a standard indexing scheme based on the L_1 metric is clearly not adapted. However, the proposed pre-classifier works very well in both cases and greatly outperforms the L_1 pre-classifier for the difficult transformations. Additionally, it requires less information than a L_1 pre-classifier to do so.

6.5.2 Scalability

We now turn our attention to the scalability of the proposed pre-classifier. By scalability, we mean the behaviour of the size of the set of candidates as the number of original images known to the system grown. Ideally, a pre-classifier keeps the number of candidates constant as the number of original images known to the system grows. For example, this is the case with the L_1 -based k nearest neighbour method discussed in section 6.5.1. Unfortunately, it needs quite a large value k to achieve low miss rates for difficult transformations. More realistically, a good pre-classifier keeps the number of candidates to a fraction of the total number of images known to the system.

In the following, we first give an overview of the experimental setup used to study the scalability of the proposed pre-classifier. Then, the results are analysed for the image collections MM270k and CGFA. Finally, conclusions on the pre-classifier scalability are drawn.

Scalability — experimental setup

To test the scalability of the system, we use the same image collections and also the same benchmarks as in section 6.5.1. Likewise, N original images are indexed using algorithm 4 with

$d = 30$ as described in section 6.4. In this experiment, the number of original images N is first set to 25, then to 50, 100 and 200. For the case $N = 200$, the two hundred images are the same than those used in section 6.5.1. On the other hand, the original images for the other cases ($N < 200$) are randomly chosen among these two-hundred images. In each case, the miss rate and the hit rate are then measured for different search box size, see section 6.5.1 for more information. In order to obtain smooth curves, the experiments are run five times in the cases where $N < 200$. For each run different original images are selected; and the results are finally averaged.

Scalability — MM270k and CGFA

Figure 6.3 shows the scalability results for the image collection MM270k. The first row (figure 6.3a and figure 6.3b) gives the hit rate, average size of the candidate set \mathcal{C} , in function of the miss rate while the second row (figure 6.3c and figure 6.3d) reports the same information but normalised with respect to the number of original images known to the system. Similarly, the first column (figure 6.3a and figure 6.3c) gives the hit rate found for duplicate test images while the second column (figure 6.3b and figure 6.3d) shows the hit rate obtained for unrelated test images. The first column additionally depicts the same information obtained using the L_1 -based k nearest neighbour (KNN) pre-classifier.

As expected, the L_1 -based KNN pre-classifier is affected only slightly by the number of original images. On the other hand, the average number of candidates returned by the proposed pre-classifier grows proportionally with the number of original images known to the system. Figure 6.3c shows that the differences between the proposed and the L_1 -based KNN pre-classifiers decrease as the number of original images increases. Further experimentations are necessary in order to determine until which point the proposed pre-classifier is better, or if the performances of the L_1 -based KNN pre-classifier remain unaffected as the number of original images grows.

One really interesting point lies in the second rows of figure 6.3. Recall that figure 6.3c and figure 6.3d show a hit rate normalised with respect to the number of original images known to the system. For duplicate images, figure 6.3c shows that the normalised hit rate diminishes as the number of original images grows. This is expected since, except for low miss rates, the hit rate for duplicate test images is around one.

More interestingly, figure 6.3d shows that, for unrelated test images, the average fraction of original images contained in the set of candidates is virtually independent from the number of original images known to the system. This result is significant for two reasons. Firstly, it means that the pre-classifier indeed works as expected because it selects a fixed fraction of the original images as candidates and discard the rest. Secondly, it gives also an idea on the behaviour of the normalised hit rate for duplicate test images as the number of original images grows large. More precisely, let a working point on the curve duplicate test images be estimated as follows: the miss rate is denoted by m and the estimated hit rate is given by $1 - m + h$ where m is the miss rate and h is the corresponding hit rate obtained for unrelated test images, for more details refer back to section 6.5.1. Then, the normalised hit rate is given by $(1 - m + h)/N$ where N is the number of original images, which is equals to $(1 - m)/N + h/N$. Note that the first term, $(1 - m)/N$, is at most equals to one and quickly vanishes as N grows. On the other hand, the second term, h/N , is

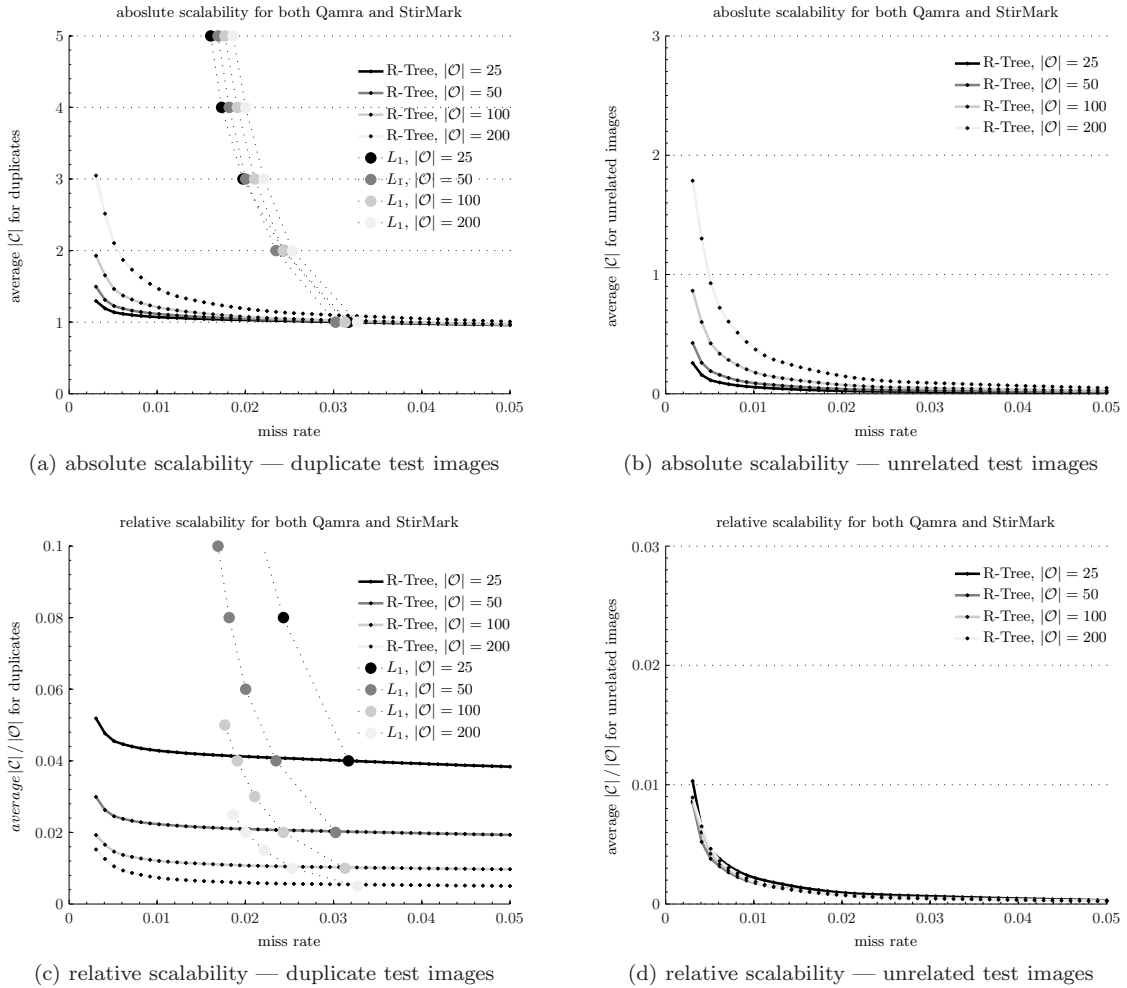


Figure 6.3: *MM270k* collection — *pre-classifier scalability*. This figure shows the scalability obtained by the L_1 -based pre-classifier compared to that achieved by the proposed pre-classifier. Both Qamra and StirMark benchmarks are used.

nearly constant, as observed in figure 6.3d. This means that h/N dominates the hit rate for large values of N . Consequently, the normalised hit rate on duplicate test images tends toward that obtained on unrelated test images as N grows. This implies that the fraction of original images returned by the pre-classifier becomes also constant for duplicate test images as the number of original images known to the system grows.

We now give an example of the efficiency of the proposed pre-classifier. In this paragraph, we consider a working point corresponding to an average miss rate of 0.005. At this working point, the pre-classifier returns $0.005 \cdot N$ potential candidates on average, where N is the total number of original images. In other words, 99.5 percent of the original images are discarded while the correct original, if the test image is a duplicate, is kept in 99.5 percent of the cases.

Finally, quite similar results are obtained for the CGFA collection, as shown in figure 6.4. The same analysis than that made for the MM270k collection applies. As for many other experiments,

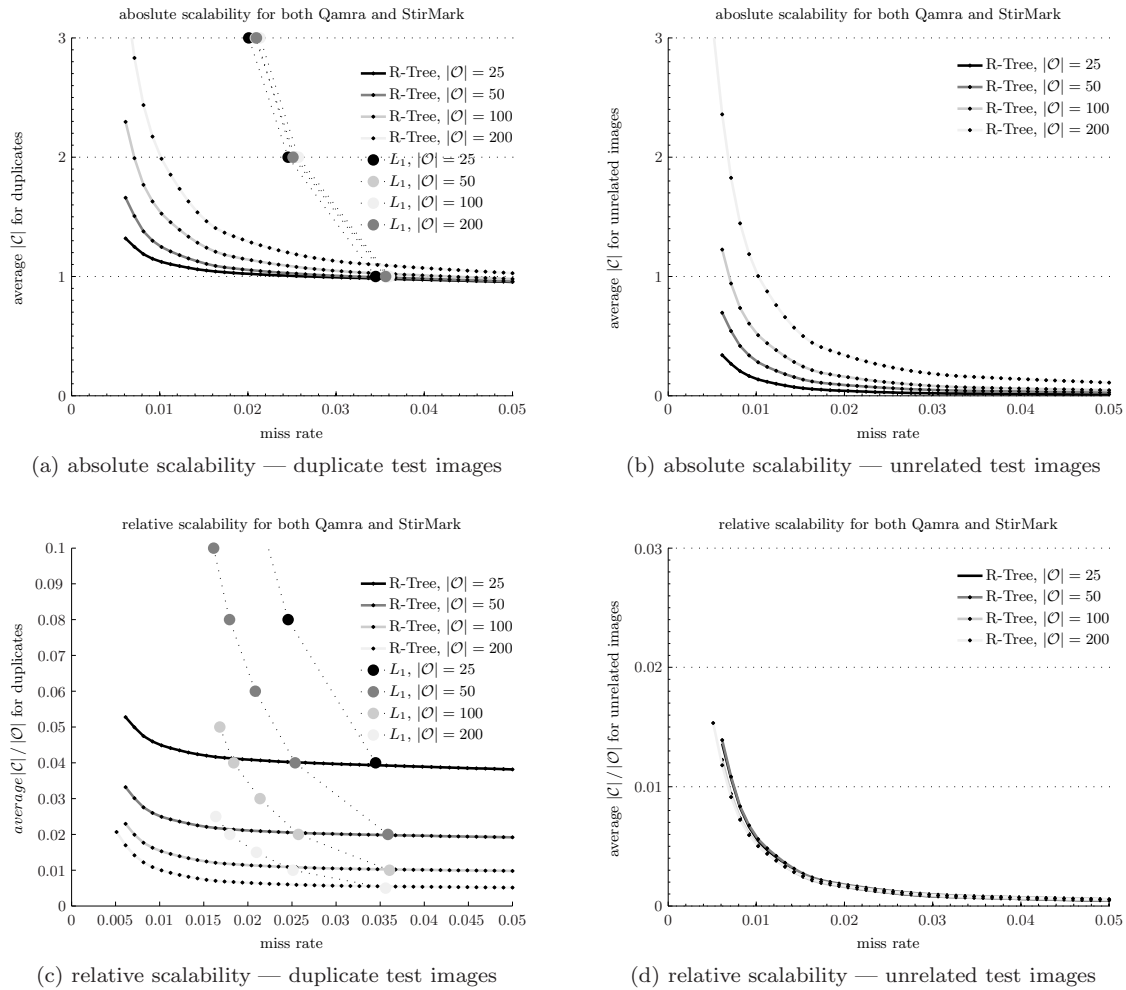


Figure 6.4: *CGFA* collection — *pre-classifier scalability*. This figure shows the scalability obtained by the L_1 -based pre-classifier compared to that achieved by the proposed pre-classifier. Both Qamra and StirMark benchmarks are used.

it can be seen that the performances attained on the CGFA collection are lower than those obtained on the MM270k.

Scalability — conclusion

The proposed pre-classifier scales well as the number of original images increases. Indeed, it returns, in average, a fixed fraction of the total number of original images irrespective of the number of original images known to the system. For example, for an average miss rate of 0.005, the pre-classifier returns $0.005 \cdot N$ potential candidates on average where N is the total number of original images. In other words, 99.5 percent of the original images are discarded.

6.6 Results for the system

In this section, we present experimental results in order to evaluate the proposed duplicate detector. The first experiment, presented in section 6.6.1, presents the performance obtained by the complete multiple original images duplicate detector. The second experiment, accounted for in section 6.6.2, present the storage space and the computational resource required by the proposed duplicate detection system. The final experiments, described in section 6.6.3, analyses the proposed system’s performance with respect to two other state of the art methods.

6.6.1 Performance

In this experiment we explore the performance of the proposed multiple original images duplicate detection system.

To test the performance of the system, we use the same image collections and also the same benchmarks as in section 6.5.1. Additionally, the system whose performances are assessed is the one presented in algorithm 1. The pre-classifier is trained according to algorithm 4 using $d = 30$ dimensions, see section 6.4 for more information. Finally, the binary classifiers are constructed as described in chapter 5 using $\alpha = 10^5$ for the F-score, and the used training examples are given in table 5.3. The number of original images is set to $N = 200$.

The metric used to evaluate the system’s performances is given in section 4.3. The performances are measured in terms of tradeoff between false positives error rate and false negatives error rate. More precisely, a false positive is a true unrelated image detected as a duplicate of one of the original images. Conversely, a false negative is true duplicate image detected as an unrelated image. Recall that in the case of the multiple original images duplicate detection system, a working point where the false positive and false negative error rates equal, say, 0.0001 and 0.01 respectively, signifies the following: given a randomly chosen original, the system detects a fraction of one out of ten-thousand unrelated test images as duplicates of this original while one out of one-hundred duplicate test images of this original are not detected as such.

There are two ways of estimating the false positives error rate. More precisely, the false positives error rate can be first estimated by taking into account that the duplicates of one original images are unrelated to any of the other original images. In this first way of estimating the false positives error rate, a duplicate test image detected as a duplicate of the wrong original is considered to be a false positive. This is quite correct but it gives rise to a skew in the estimation of the false positives rate. For instance, forty test duplicates are generated per each original for the Qamra benchmark. In this case, there are an additional $7960 = 199 \times 40$ test images that are considered unrelated to each original. Now, in the CGFA collection there is about the same number of real unrelated test images, actually 8800. Since it is quite likely that the system correctly classify the duplicate test images, this means that about half of the images used to estimate the false positives generates very few false positives. Consequently, the false positives error rate is, in the author opinion, underestimated by a factor up to two. Due to its larger size, this effect is less important for the MM270k collection but it is nevertheless present. The second way of estimating the false positive error rate is simply not to take into account the duplicate test images in its estimation.

In the following, both estimations are shown because some published works use the first method [Qamra *et al.*, 2005] and others use the second method [Ke *et al.*, 2004]. Note that this problem is not present in chapter 5 since the binary classifiers are tested independently. We denote the first false positives error rate estimator E_{fp}^1 and second one E_{fp}^2

Now, the system can be tested under different constraints. For instance, the size δ of the search box used in the pre-classifier can be changed as shown in algorithm 2. Additionally, it is also possible to vary the threshold u used to decide whether to trust or not the label returned by algorithm 1. To synthesis, as much as possible, the different possible parameterisation of the system we use three different values of δ , namely 0.025, 0.05 and 0.1, and, for each of them, we estimate the tradeoff between the false negatives and the false positives error rates by varying the threshold u between zero and one. Each values of δ corresponds to an average miss rate achieved by the pre-classifier as well to an average hit rate, size of the set of candidates. The miss rate relates to the the performance of the system while the hit rate is linked to the computational efficiency of the system.

Performance — MM270k and CGFA

Figure 6.3 shows the performances obtained on the image collection MM270k. More precisely, figure 6.5a depicts the performances achieved for both the Qamra and StirMark benchmarks together while figure 6.5c and figure 6.5d picture the performance obtained for each benchmark separately. Additionally, figure 6.5b shows the average size of the set of candidates for different values of δ .

The false negatives versus false positives error rates curves can be split into two parts. In the first part, right hand side of the curves, the system’s performance is limited by the pre-classifier miss rate. Indeed, the system’s false negatives error rate cannot go below the miss rate imposed by the pre-classifier and the curves flatten out. Consequently, the minimal achievable false negatives rate is linked to the δ size of the search box used in algorithm 2. In the second part, left hand side of the curves, the system performance is limited by the binary classifiers. Clearly, the influence of the pre-classifier diminishes as the false positives rates diminishes and the curves obtained for the different values of δ tend to the same asymptote.

Additionally, it can be seen that the false negatives error rate increases dramatically once the false positives error rate goes below 10×10^{-5} . This behaviour has two explanations. The first reason relates to the presence of near-duplicate images in the MM270k collection, refer to section 5.4.3 for more details. As already explained in chapter 5, some unrelated test images are always detected by the system as duplicates because they are photographs taken at the same place than some of the original images but at slightly different time. The second reason concerns the limitation on the discriminatory power of the features; the used features do not permit to differentiate between some visually similar yet unrelated images. As already hinted in chapter 5, this is the key downside of any content-based duplicate detection system.

Now, we examine the effect of using one or the other method to estimate the false positives error rate. Basically, both methods result in virtually the same curves but shifted on the horizontal axis. Actually for the Qamra benchmark, a false positives error rate estimated using E_{fp}^1 is, for the same

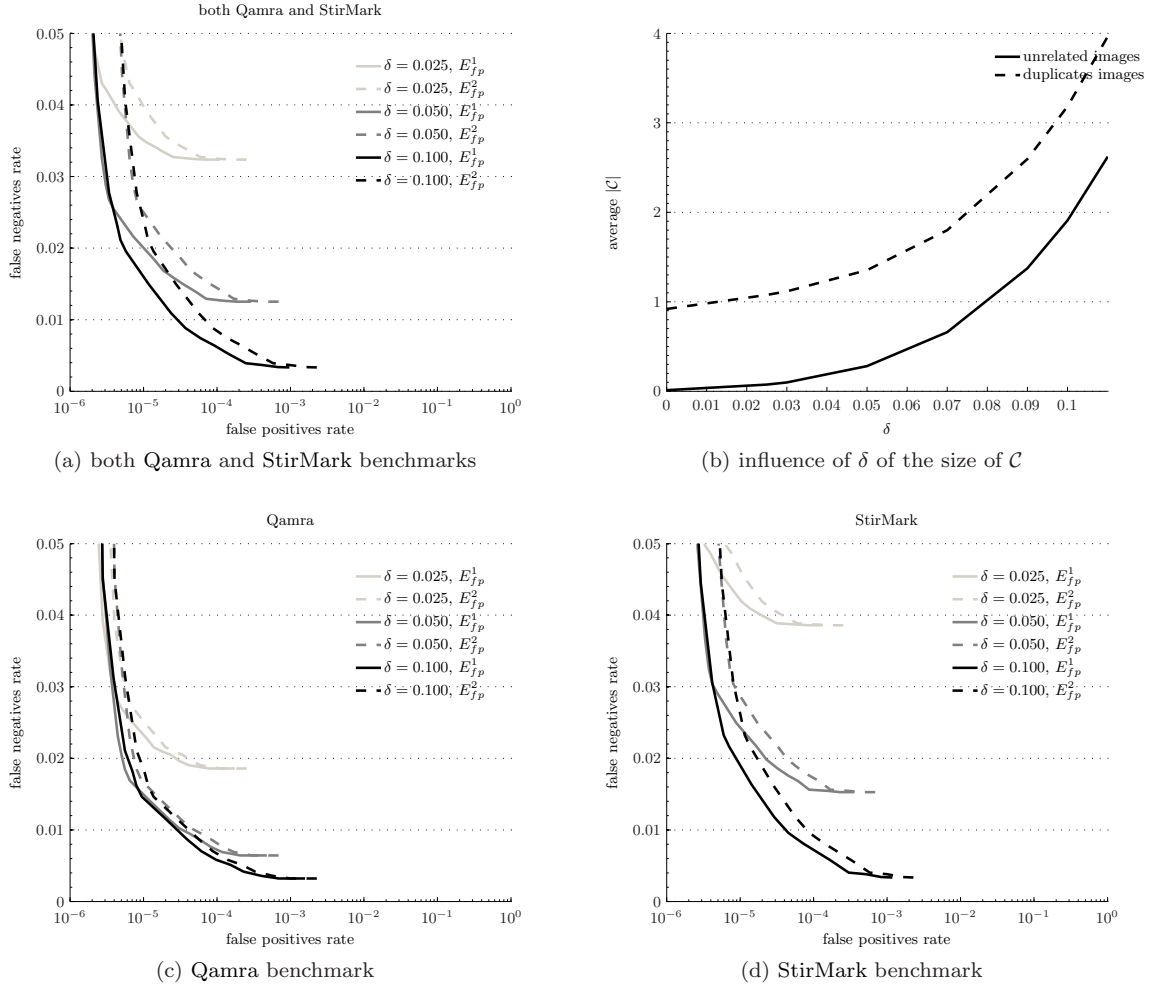


Figure 6.5: *MM270k* collection — system performance. This figure shows the performances achieved by the proposed multiple original images duplicate detection system.

false negatives rate, smaller by a factor roughly equals to 1.41 than that estimated using E_{fp}^2 . For this method, 7960 duplicate test images are used, in addition of the 18 585 unrelated test images, to estimate the false positives. We believe, see the remarks given previously, that a false positives rate estimated by this method can be underestimated by a factor up to $(18\,585 + 7960)/18\,585 = 1.43$. This estimated value is slightly larger than the one observed in reality but not by much.

Finally, the results obtained for the CGFA collection are quite similar to those obtained for the MM270k collection. There are two main differences. The first difference is the absence of a sharp increase of the false negatives rate below false positives rates of 10^{-4} . This is due to the absence of near-duplicates in the CGFA collection. The second difference concerns the ratio between false positives error rates estimated by one or the other method. For the CGFA collection, it is equal to values ranging from 1.6 to 1.8. As already said previously, a false positives rate estimated by taking into account duplicates can be underestimated by a factor up to $(8800 + 7960)/8800 = 1.9$.

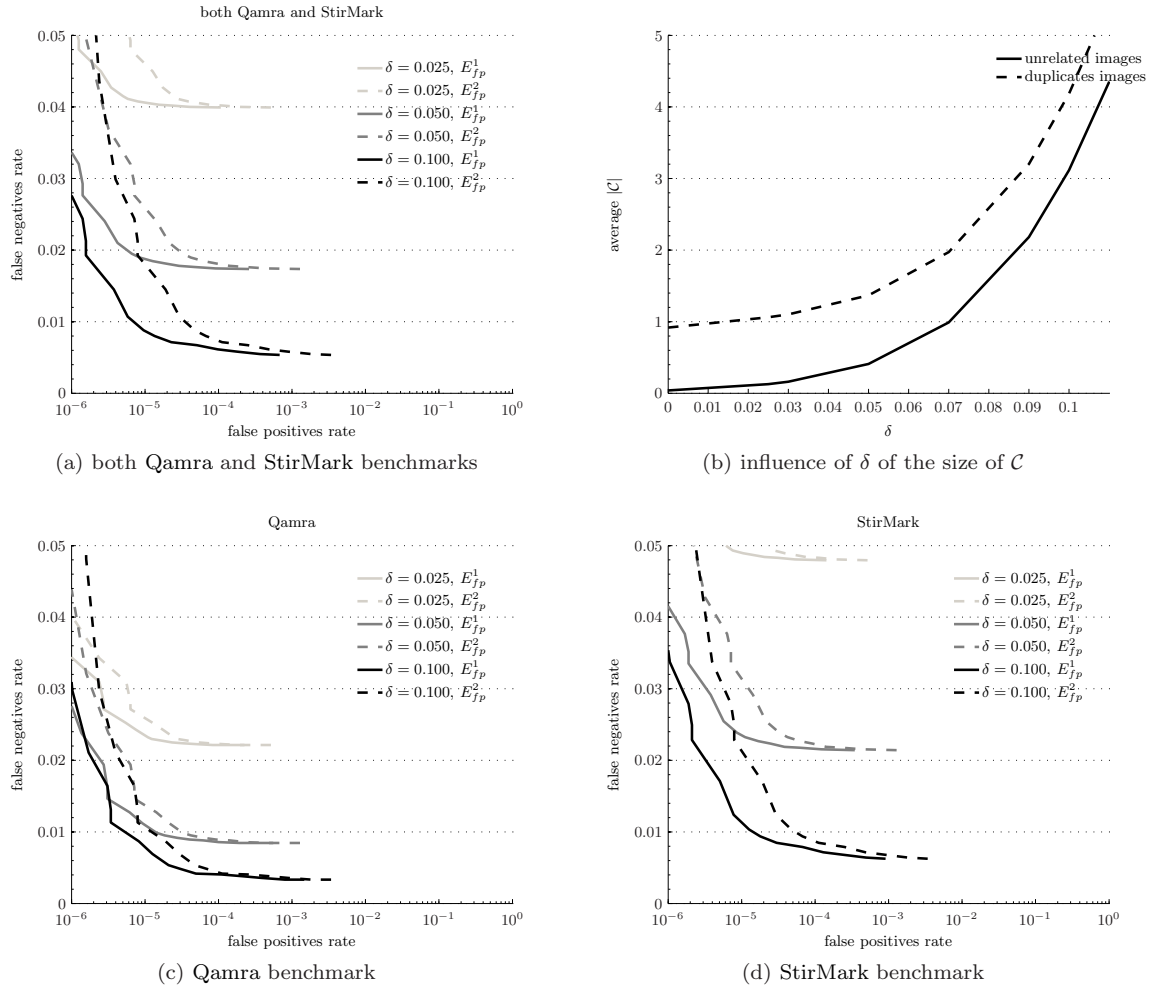


Figure 6.6: *CGFA collection* — *system performance*. This figure shows the performances achieved by the proposed multiple original images duplicate detection system.

Performance — conclusion

The proposed duplicate detection system performs quite well. For instance, on the MM270k collection and for a randomly selected original image it can, on average, detect 99 percent of its duplicate (generated by the Qamra benchmark) while assigning a fraction of only about 3×10^{-5} of the unrelated test image to that original. The performance are even better on the CGFA collection where detecting 99 percent of the duplicate test images corresponds a false positives error rate of only about 1.5×10^{-5} . Additionally, it is remarked that there different ways of estimating the false positives rate can result in quite different estimations. Finally, it is noticed that the minimal false negatives rate achievable by the system is linked to the δ size of the search box (see algorithm 2). On the other hand, for low false positives rate, the system's performance is limited by the binary classifiers.

Table 6.1: *Storage requirements estimation and average running time for testing.*

name	size, B	original	name	time, s
pre-classifier, projection	$162 \cdot 30 \cdot 2 = 9720$	independent	preprocessing	0.1
pre-classifier, indexation	$200 \cdot 30 \cdot 2 = 12\,000$		feature extraction	0.5*
PCA projection matrix	$162 \cdot 162 \cdot 2 = 52\,488$		pre-classifier, projection	45×10^{-6}
normalisation constants	$2 \cdot 162 \cdot 2 = 648$		pre-classifier, search	1.1×10^{-3}
SVC, support vectors \mathbf{x}_i	$162 \cdot 130 \cdot 2 = 21\,060$	dependent	PCA projection	10×10^{-6}
SVC, $y_i \alpha_i$	$162 \cdot 2 = 324$		normalisation	60×10^{-6}
			decision function	50×10^{-6}

(a) *Storage requirements estimation.* Real number are coded on 16 bits (two bytes).

(b) *Average running time for testing.* The experiments were carried out on a PC with a 2.8GHz processor and 2Go of memory.

6.6.2 Requirements on storage and computational effort

The proposed duplicate detection method requirements are now analysed in terms of storage space and computational effort. For this purpose, we build on the analysis already reported in chapter 5.

A number of parameters are needed to compare a test image to a given original. Namely, they are the PCA projection matrix for the pre-classifier, the R-Tree indexing for the pre-classifier, the PCA projection matrix for the binary detectors, the normalisation constants for the binary detectors, and the support vectors of the decision functions of the binary detectors. The PCA projection matrices used for the pre-classifier and that used for the binary detectors are independent of the original images. The remaining parameters depend on the original images and are, in the following, referred to as the description of the original image. The storage requirements are detailed in table 6.2a. On average, about 33kB are needed to store the description of each original. In other words, one megabyte can held, on average, up to thirty originals. This is a negligible amount of memory for today's computers.

Another important aspect is that of the computational complexity of the method. The proposed method requires training for each original image. Training is computationally complex and it can, indeed, take up to twelve minutes to train a detector on a PC with a 2.8 GHz processor and 2 Go of memory. Feature extraction from the synthetic duplicate examples, cross-validation to find good parameters of the SVC and cross-validation to find good parameters of the pre-classifier are the most complex parts of the training, and together take up to ninety percent of the running time. Since training can be done off-line, its computational complexity is less critical than that of testing.

The computational complexity of testing is estimated in table 6.2b. Note that except for the SVC and the R-Tree parts, the method is implemented in Matlab without any optimisation. This incurs longer running time. For instance, the feature extraction could be reduced to, at least, 0.1seconds [Qamra *et al.*, 2005]. In the discussion that follows, we assume an optimised feature extraction step. The preprocessing, feature extraction and the pre-classifier steps are independent of the original image, and take about 0.22seconds. On the other hand, the remaining steps depend on the original image, they take about 0.1×10^{-3} seconds per binary detector. However, not all originals have to be tested since the pre-classifier discard most of them. The exact number of original images that are discarded depends on the search box size used in algorithm 2. In the following, we use a search box size such that, on average, 99 percent of the original are discarded.

This corresponds to a miss rate inferior to 0.005 for both Qamra and StirMark benchmarks and on both MM270k and CGFA collections.

Let us consider the following scenario. A company is checking images circulating on the Internet to see whether they contain duplicates of original images for which it holds copyright. In this scenario, the company has to test an image with, on average, one percent of the detectors. When the number of owned original images is less than 22 000, most of the testing time is spent on preprocessing, extracting features from the test images, and in the pre-classifier. In that case, up to four test images can be processed per second and per computer. For a larger number of original images, most of the testing time is spent on the original image dependent steps. The number of test images that can be processed per second decreases linearly as the number of original images grows.

6.6.3 Comparison with existing duplicate detection methods

We now compare the performance of the proposed method with that of existing duplicate detection systems. The same existing works are used as in chapter 5, namely key points (KPs) [Ke *et al.*, 2004] and perceptual distance function (DPF) [Qamra *et al.*, 2005].

Comparison — results and analyse

Figure 6.7 compares the performance of the proposed duplicate detection system with state of the arts techniques reported in [Ke *et al.*, 2004; Qamra *et al.*, 2005]. The black line corresponds to the DET curve obtained with our system. The light grey line represents the performance of a duplicate detection method based on DPF [Qamra *et al.*, 2005]. The cross indicates the performance of a duplicate detection system based on KPs [Ke *et al.*, 2004].

It can be seen that the proposed method achieves quite good performance. For instance, on the CGFA collection, an average FN error rate of 5×10^{-3} corresponds to a fixed false positive error rate of 5×10^{-5} . On the other hand, on the MM270k collection, an average FN error rate of 2×10^{-4} corresponds to a fixed false positive error rate of 2×10^{-3} . This is not as good than on the CGFA collection because the MM270k collection contains near-duplicate as explained in section 5.4.3.

Now, comparing the performance of the DPF method with that of the proposed system two things can be observed. First, the DPF method achieves no FN error for false positives error rates above 1×10^{-3} . However, once below that point the performance degrades extremely rapidly. Second, while DPF performs somewhat better than the proposed system for false positive error rates above 1×10^{-3} , it is clearly outclassed below that threshold.

On the other hand, the proposed method is outperformed by KPs. Indeed on the CGFA collection, KPs achieves a FN error rate of 1.5×10^{-3} for no false positive error. On the other hand, the proposed method never reaches, for the same test set, no false positive error. However, the performance gap is quite low for the MM270k collection but these results are less significant since they are extrapolated for the KPs method, and possibly its performances are inflated. The explanation for the better results of the KPs are the same than given in chapter 5.

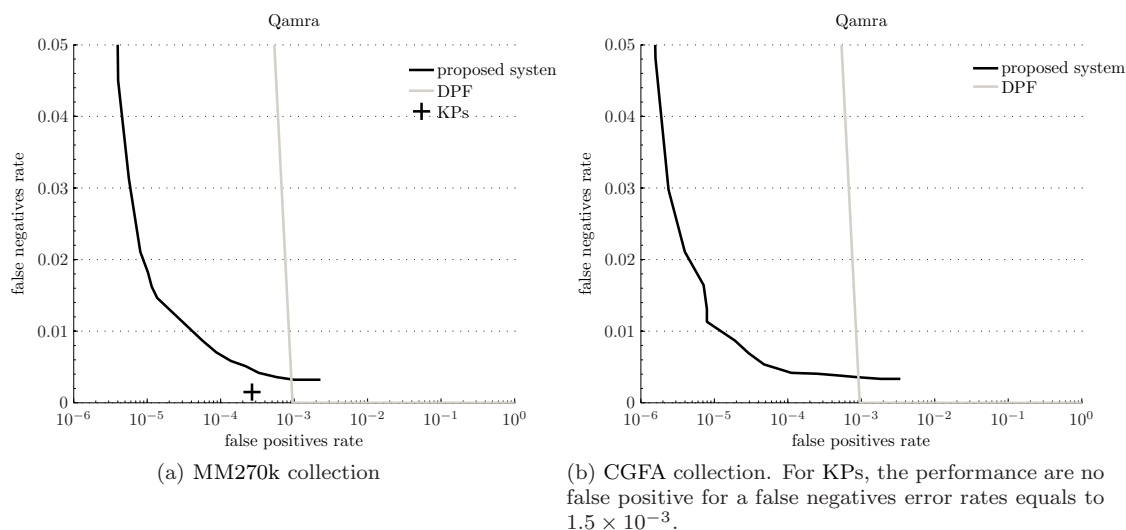


Figure 6.7: *Comparison with state of the art methods.* The proposed system is compared with two state of the art system, namely KPs [Ke *et al.*, 2004] and DPF [Qamra *et al.*, 2005]. The test are carried out on two different image collections, MM270k and CGFA, and the Qamra benchmark is used to generate the test duplicates.

6.7 Exploratory and future works

In this section, we present two directions of research concerning the pre-classifier. The first direction of research, given in section 6.7.1, concerns an indexing scheme that works well on high-dimensional spaces. The second avenue of research, reported in section 6.7.2, relates to efficiently describe image regions rather than the whole image.

6.7.1 Random projection

In section 6.5, it is noticed that the pre-classifier performs better when a low number of dimensions, namely thirty, is used to index the duplicate than when more dimensions, namely sixty, are used. This is a good example of the effect the curse of dimensionality [Donoho, 1998]. On the other hand, the L_1 -based pre-classifier behaves similarly but the optimal number of dimensions is higher in this case, namely eighty. Consequently, the difference of fifty in the optimal number of dimensions shows that the added features, in the L_1 case, still carry discriminative information. The question is then how to efficiently use this additional information?

Indexing of high dimensional space has been extensively studied, and works abound on how to avoid or lessen the curse of dimensionality. Nonetheless, no method exists that entirely solves it. A popular solution that works quite well is called locally sensitive hash [Gionis *et al.*, 1999; Indyk and Motwani, 1998]. In short, this approach consists in randomly projecting the features into a smaller space by randomly selecting subset of the entire features' set. A single feature vector is thus represented by many random projections. Finally, each projection can be indexed within a small dimensionality space and hence avoid the curse of dimensionality. Later, the result of a

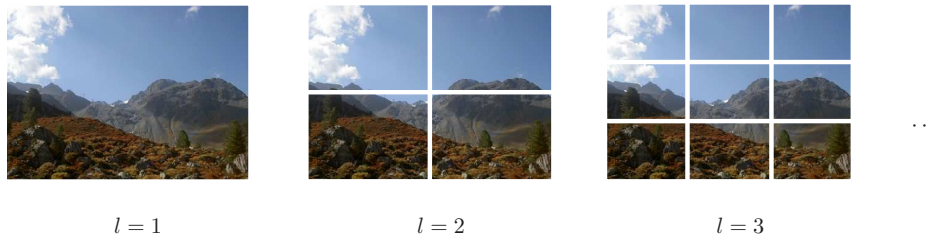


Figure 6.8: *Image Patches*. This figure represents the image patches at different granularity levels.

query to the database consists in the records whose several random projections match those of the query.

This approach could be applied to the pre-classifier presented in this chapter since it is based on indexing. However, adaptation of the algorithm and experimentations are necessary so as to determine the impact of random projection on the performance of the pre-classifier.

6.7.2 Hierarchical duplicate detection

We propose to analyse images at different granularity levels l . At each granularity level, the image is subdivided into patches of the same size. For instance, at the coarsest granularity level there is one patch of the size of the image, at the next level there are 4 patches, then 9 patches and so on. Figure 6.8 shows these patches for the three first granularity levels. Each patch is then described by the features detailed in section 4.4.2.

We next explain the potential behind the different granularity levels that are used for the image description. Clearly, an image is composed of different regions, each having different characteristics, as visible in figure 6.8. Global features, features describing the image as a whole, give an averaged version of the characteristics of every regions and perform well for duplicate detection [Maret *et al.*, 2006a; Qamra *et al.*, 2005]. It is however possible for unrelated images to have very similar global features, in which case they will be considered to be duplicates of each other. The use of an image descriptions with granularity levels permits to lessen the number of such clashes. While unrelated images might have similar global features, it is less likely for the majority of their patches to have similar features. This is the main idea underlying the hierarchical approach proposed in the following. The subspace spanned by the duplicates is constructed for each granularity level. Then, the potential originals of a test image are determined using only the global features granularity level. Every patches of the test image are subsequently tested on the finer granularity partition of each original found earlier, and the corresponding original is kept only if the number of matching patches is sufficient. Finally, the operation can be repeated for the remaining originals until reaching the finest granularity level, we experimented with up to $L = 3$. The above observations lead us to devise the hierarchical pre-classifier presented thereafter.

Hierarchical pre-classifier

The proposed pre-classifier's extension works in a hierarchical way. It starts at level $l = 1$ and continues at finer granularity levels, possibly up to $l = L$. The initial set of candidates is determined

Algorithm 5 Finds the potential originals of a test image

Require: originals to be indexed in the R-tree with algorithm 6

```

1: procedure HIERARCHICAL_PRE_CLASSIFIER( $\mathbf{I}$ ,  $\delta$ ,  $\mathbf{m}$ )
2:   for  $l = 1$  to  $L$  do
3:      $\mathcal{R} = \emptyset$ 
4:     for  $b = 1$  to  $(l + 1)^2$  do ▷ treat each patch separately
5:        $\tilde{\mathbf{I}} = \text{GET\_PATCHES}(\{\mathbf{D}\}_{i=1}^D, l, b)$ 
6:        $\{(ID_i, \tilde{l}_i, \tilde{b}_i)\}_i = \text{PRE\_CLASSIFY}(\tilde{\mathbf{I}}, \delta_i)$  ▷ see algorithm 2
7:        $\mathcal{R} = \mathcal{R} \cup \{(ID_i, \tilde{l}_i, \tilde{b}_i)\}_i$ 
8:     if  $l = 1$  then
9:        $\mathcal{C} = \{ID \text{ such that } (ID, l, 1) \in \mathcal{R}\}$  ▷ determine the initial set potential originals
10:    else
11:      for  $ID \in \mathcal{C}$  do
12:        if  $|\{b \text{ such that } (ID, l, b) \in \mathcal{R}\}| < \mathbf{m}(l)$  then
13:           $\mathcal{C} = \mathcal{C} \setminus ID$  ▷ not enough patches match
14:  return  $\mathcal{C}$ 

```

Algorithm 6 Estimates a set of duplicate manifolds of an original and indexes it

Require: the original image \mathbf{I} , its identifier ID , the parameters $\delta \in [-1, +1]$, k and L , and the duplicate examples $\{\mathbf{D}\}_{i=1}^D$

Ensure: the R-tree contains the duplicate region estimation for the original

```

1: procedure HIERARCHICAL_INDEXATION( $\mathbf{I}$ ,  $ID$ ,  $\delta$ ,  $k$ ,  $L$ ,  $\{\mathbf{D}\}_{i=1}^D$ )
2:   for  $l = 1$  to  $L$  do ▷ treat each level separately
3:     for  $b = 1$  to  $(l + 1)^2$  do ▷ treat each patch separately
4:        $\{\tilde{\mathbf{D}}\}_{i=1}^D = \text{GET\_PATCHES}(\{\mathbf{D}\}_{i=1}^D, l, b)$ 
5:       FINE_INDEXATION( $\mathbf{I}$ ,  $(ID, l, b)$ ,  $\delta$ ,  $k$ ,  $\{\tilde{\mathbf{D}}\}_{i=1}^D$ ) ▷ see algorithm 4

```

at level $l = 1$. At finer granularity levels, this set is pruned by removing the originals with not enough matching patches. The operation is repeated until the finest granularity level is reached. Algorithm 5 gives the pseudo-code of the hierarchical classifier. It makes use of the pre-classifier given by algorithm 2 in section 6.4.

Algorithm 6 describes the constructions of the set of duplicate manifolds for a given original image. At each granularity level l , the training examples are subdivided into $(l + 1)^2$ patches. Each patch is then described by a feature vector. A duplicate manifold is then estimated for each patch at a given granularity level. To achieve this, the algorithm 4 developed in section 6.4 is used.

Results and remarks

The proposed hierarchical algorithm is implemented and was presented with more details in [Maret *et al.*, 2006b]. The preliminary results are encouraging since the average size of the set of candidates, for unrelated test images, can be reduced by a factor 2.6 for two levels of granularity and by a factor 3.4 for three levels of granularity. Moreover, it should be noted that this hierarchical approach could also be applied to the binary detectors. Of course, the applicability of such a hierarchical system necessitates further research.

6.8 Chapter summary

In this chapter we presented a multiple original images duplicate detection system based on the binary detector previously presented in chapter 5. To create an efficient system, a test image is checked only on the binary detectors corresponding to the most likely original images. Their selection, using a pre-classifier, is the main contribution of this chapter. The performance of the pre-classifier is then analysed. Subsequently, the entire system is analysed and compared with state of the art methods. Finally, a possible improvement on the system is proposed.

The proposed system is composed of the five steps outlined thereafter. In the first step, global statistics are used to describe the image. In the second step, the number of features is reduced. In the third step, the most likely originals are selected by means of an R-Tree. They form the set of candidates. In the fourth step, the binary detectors developed in the chapter 5 are applied to each element of the set of candidates. Finally, the element with the highest probability is selected and the test image is estimated, by the system, to be a duplicate of the corresponding original. The system also provides a probability estimate of the correctness of this choice.

The performance of the proposed system is assessed, using standard benchmarks, and the result is analysed. It is found out that the proposed multiple original images duplicate detector greatly outperforms detectors using the same features but based on the L_1 metric. The system is additionally compared to state of the art duplicate detection techniques. It outperforms the DPF method, which uses more feature to describe the image. While the proposed method is slightly outperformed by the KPs method, it is five to ten times computationally less complex.

Finally, the performance of the proposed system can be greatly improved by using a hierarchical system that subdivides the images into a pyramid of patches and create detectors tuned to each patch. A second direction of research relates to a pre-classifier that performs better on high-dimensional spaces. However, these avenues of research necessitate further works.

7

General Conclusions

7.1 Summary of the achievements

In chapter 3, we saw that the problem of duplicate image detection originates from different fields, namely watermarking and content-based retrieval. The pros and cons of each approach were then reviewed. Basically, content-based duplicate detection is more flexible but not yet as mature as watermarking in terms of precision and recall rates. Additionally, an inherent weakness of any content-based technique is the impossibility to distinguish between photographs of the same scene taken from slightly different angles or at different time. On the other hand, watermarking is less flexible than content-based method because it requires modifying the image prior to its dissemination. This requirement is the cause of the two major drawbacks of watermarking as described thereafter. Firstly, watermarking is adapted only if one has total control over the original artwork and is ready to modify it. Secondly, a watermarked image is detectable as long as a mean to efficiently remove the watermark is not discovered. Once the watermark has been removed from an image, it is definitely impossible to detect copies of that unmarked image by using watermarking techniques. All in all, watermarking and content-based approaches are quite complementary.

In chapter 4, we developed a framework for content-based duplicate detection systems. The duplicate detection framework first consists in a model of the subspace spanned by the duplicates of an original image. This model permits to explore some characteristics of the duplicates of an image; for example it is found that, under certain assumptions, the duplicates form a manifold embedded within the image space. The second element of the framework is a generic duplicate detection system. Through this generic system, we develop our view of duplicate detection, namely the classification of a test image into one of $K + 1$ classes. K classes correspond to the K original images known to the system, or in other words “the test image is a duplicate of one of the known originals,” while the remaining class stands for “the test image is unrelated to any of the known

original images.” Finally, the last element of the framework concerns the evaluation methodology of a duplicate detection system based on the presented classification approach.

Still in chapter 4, we gave an overview of the actual duplicate detection system developed in this thesis. The system is composed of four steps, namely feature extraction, pre-classifier, binary duplicate detectors, and final decision. Feature extraction consists in describing images by means of relevant visual statistics. The pre-classifier aims at selecting a limited number of originals among the K original images; an original is selected if the test image is potentially one of its duplicates. The binary duplicate detectors consist in binary classifiers used to determine the probabilities that the test image is a duplicate of each selected original image. In the last step, the decision simply consists in selecting the most probable original.

In chapter 5, we presented our binary duplicate detector. The main idea behind the proposed detector is to adapt duplicate detection to a specific original image. The system is then able to classify test images as duplicates of the original image or as unrelated images. The binary detector uses the image description given in chapter 4, and is composed of the three steps outlined thereafter. In the first step, the features are linearly projected so as to obtain a better separation between duplicates of the original image and unrelated images. In the second step, the elements of the projected feature are normalised according to the statistical distribution of the duplicates. In the last step, a non-linear decision function, based on a support vector classifier, is used to determine the probability that the test image is a duplicate of the original image. The performance of the proposed system is assessed and the results are analysed. It is found out that the proposed SVC-based duplicate detector greatly outperforms detectors using the same features but based on the L_1 metric. The proposed binary detector is then compared to state of the art system. It outperforms the perceptual distance function (DPF) method, which uses more feature to describe the image. While the proposed method is slightly outperformed by the key points (KPs) method, it is five to ten times less computationally complex.

In chapter 6, we gave an account of the entire duplicate system. Contrary to chapter 5, the system knows a set of original images and not only a single original. The proposed system uses the image description given in chapter 4, and is additionally composed of the three steps outlined hereafter. In the first step, the number of features is reduced. In the second step, the originals that are most likely to be duplicates of the test image are selected by means of an R-Tree. They form the set of candidates. In the third step, the binary detectors developed in the chapter 5 are applied to each element of the set of candidates. Finally, the element with the highest probability is selected. The system estimates that the test image is a duplicate of this original if the corresponding probability is higher than a certain threshold. The performance of the proposed system is assessed and the results are analysed. It is found out that the proposed multiple original images duplicate detector greatly outperforms detectors using the same features but based on the L_1 metric. Additionally, it also outperforms the DPF method, which uses more features to describe the image. While the proposed method is slightly outperformed by the KPs method, it is five to ten times computationally less complex.

To conclude our summary, we would like to point out that this thesis’s nature is mainly exploratory. Indeed, to the best of the author knowledge, it is one of the first attempts to apply machine learning techniques to the problem of content-based duplicate detection.

7.2 Perspectives

The work proposed in this dissertation can be improved and extended in several ways. Some directions for further works are proposed below.

- The model given in chapter 4 can be extended by incorporating more knowledge on the nature of the image transformations used to create the duplicates. A possible starting point is the research's results reported in [Simard *et al.*, 1998].
- The benchmark procedure presented in chapter 4 can be standardised and offered to the community in a manner similar to what has been done for watermarking. However, while watermarking benchmarks only need to include transformations, content-based benchmarks should also standardise sets of original images as well as sets of unrelated images. Indeed, an important aspect of content-based duplicate detection methods is that of image description, which clearly depends on the used images.
- The binary duplicate detectors presented in chapter 5 can be improved in several ways:
 - The combination of several simpler classifiers per original can greatly improve the detection performances. A possible starting point is the seminal paper [Breiman, 1996].
 - The optimal choice of the training examples remains still an open issue. More precisely, the duplicate examples used to train the classifier are manually chosen and might not be optimal. It would be interesting to devise an automatic algorithm to determine a set of good, possibly optimal, training examples given a set of transformations to be detected.
 - The projection step is, as implemented now, independent of the original image. This step could be made original-dependant so as to find a representation of the images' feature that separates well the duplicates of a particular original image from the unrelated images. Such a method might be based on existing dimensionality reduction techniques.
 - In this thesis, we used a support vector classifier to decide whether a test image is a duplicate or unrelated to an original image. Many other types of classifiers exist, and it could be enriching to try different approaches.
 - Under some assumptions (the transformations are smooth), the subspace spanned by the duplicated is a smooth manifold embedded within the image space. For this reason, it would be interesting to use non-linear principal component analysis to represent the features describing an original and its duplicates because this technique is able to project manifolds on simpler objects [Karhunen and Joutsensalo, 1994].
- The pre-classifier presented in chapter 6 can be improved in several ways:
 - A second direction of research relates to a pre-classifier that performs better on high-dimensional spaces. Indeed, the proposed pre-classifier performs best when the number of features is around thirty but it was shown, in chapter 6, that additional features contain information that helps to better pre-classify duplicates. A possible way of

improving the pre-classifier behaviour in higher dimensional spaces is that of random projections [Indyk and Motwani, 1998].

- The pre-classifier is based on the R-Tree indexing scheme that uses high dimensional rectangles. There exist many other spatial access methods that can be used. For example, some indexing schemes are based on high dimensional rectangles, hyperspheres, a mix of both, or generic metrics [Ciaccia *et al.*, 1997]. It would prove certainly enriching to adapt the proposed pre-classifier algorithm to these different spatial access methods.
- The system can be extended in several ways that either bring improved performance or new functionalities:
 - An extension of the proposed method consists in using, for example, the more complex key points method developed by Ke *et al.* as a refinement step on those test images estimated to be duplicates by the proposed system.
 - The features used to describe the images are of a global nature. Ke *et al.* showed that the use of local features can greatly improve the performance of duplicate detection. Possible approaches to incorporate local information to the proposed framework is, one, to subdivide the image in rectangular tiles and to independently describe each tile and, two, the work of Lowe. Additionally, it would permit to adapt the method to the detection of duplicates of an image subpart or of an object within test images.
 - Another possible extension, related to the previous proposition, is to adapt duplicate detection to video. In this case the goal is to detect a particular object within a video scene. For example, the police would like to detect a particular car on videos taken on a closed-circuit televisions used for surveillance. In this case, the considered transformations are of a particular nature, namely viewpoint changes and occlusions. This means that the three-dimensional nature of the object has to be taken into account.
 - Finally, an important avenue of research is that of adapting the system to the genre of the test and original images. Indeed, images are varied and can represent outdoor scenes, city pictures, paintings, cartoons, people, and more. Each visual genre can be better described using different features, for example a city picture is better described if the features contain descriptors about straight lines while this kind of descriptors might not be very useful for a natural scenery. A possible approach is to use a multitude of features that cover most visual genres. However, this approach is not realistic because, one, each added feature results in higher computational complexity, two, more training examples are needed to avoid overtraining (curse of dimensionality). For this reason, we believe that a content-based duplicate detector should adapt itself to the genre of each image. This kind of adaptation first presupposes the existence of a visual genre classifier, or at least the existence of a method that selects, based on the visual genre of the image, the features that best describe an image. Secondly, this also means that all subsequent methods have to cope with different image representations. We believe this approach to be the way to go for content-based duplicate detection.

Bibliography

- E. L. Allwein, R. E. Schapire, Y. Singer (2000). Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers. *Journal of Machine Learning Research* **1**:113–141.
- M. Barnett (1999). Digital watermarking: applications, techniques and challenges. *IEEE Electronics & Communication Engineering Journal* **11**(4):173–183.
- M. Barni (2003a). What is the future for watermarking? (part I). *Signal Processing Magazine, IEEE* **20**(5):55–60.
- M. Barni (2003b). What is the future for watermarking? (Part II). *Signal Processing Magazine, IEEE* **20**(6):53–59.
- R. Bayer, E. M. McCreight (1972). Organization and maintenance of large ordered indexes. *Acta Informatica* **V1**(3):173–189.
- N. Beckmann, H.-P. Kriegel, R. Schneider, B. Seeger (1990). The R*-tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD International Conference on Management of Data*, pp. 322–331.
- J. L. Bentley (1975). Multidimensional Binary Search Trees Used for Associative Searching. *Commun. ACM* **18**(9):509–517.
- C. Boehm, S. Berchtold, D. Keim (2001). Searching in high-dimensional spaces: Index structures for improving the performance of multimedia databases. *ACM Comput. Surv.* **33**(3):322–373.
- R. Boynton (1989). Eleven colors that are almost never confused. In *Proceedings of the SPIE Symposium Human Vision, Visual Processing, and Digital Display*, vol. Volume 1077, pp. 322–332, Bellingham.
- L. Breiman (1996). Bagging Predictors. *Machine Learning* **24**(2):123–140.
- C. J. C. Burges (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery* **2**(2):121–167.
- C. Chang, C. Lin (2007). LIBSVM: a Library for Support Vector Machines.

- N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* **16**:321–357.
- P.-H. Chen, C.-J. Lin, B. Schölkopf (2005). A tutorial on nu-support vector machines. *Applied Stochastic Models in Business and Industry* **21**:111–136.
- P. Ciaccia, M. Patella, P. Zezula (1997). M-tree: An Efficient Access Method for Similarity Search in Metric Spaces. In *VLDB '97: Proceedings of the 23rd International Conference on Very Large Data Bases*, pp. 426–435, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- I. Cox, M. Miller, J. Bloom (2001). *Digital Watermarking: Principles & Practice*. Morgan Kaufmann.
- I. J. Cox, M. L. Miller (2002). The First 50 Years of Electronic Watermarking. *EURASIP Journal on Applied Signal Processing* **2**:126–132.
- F. C. Crow (1984). Summed-Area Table for Texture Mapping. *Computer Graphics* **18**(3):207–212.
- S. R. Deans (1983). *The Radon Transform and Some of Its Applications*. Krieger Publishing Company.
- Y. Deng *et al.* (2001). An efficient color representation for image retrieval. *Image Processing, IEEE Transactions on* **10**(1):140–147.
- D. L. Donoho (1998). High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality.
- K. Duan, S. S. Keerthi, A. N. Poo (2003). Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing* **51**:41–59.
- R. Duda, P. Hart, D. Stork (2001). *Pattern Classification*. Wiley-Interscience.
- D. J. Elzinga, D. W. Hearn (1972). The minimum covering sphere problem. *Management Science* **19**:96–104.
- R. Fagin, J. Nievergelt, N. Pippenger, H. R. Strong (1979). Extendible hashing — a fast access method for dynamic files. *ACM Trans. Database Syst.* **4**(3):315–344.
- C. Faloutsos *et al.* (1994). Efficient and Effective Querying by Image Content. *Journal of Intelligent Information Systems* **3**(3/4):231–262.
- T. Fawcett (2003). ROC Graphs: Notes and Practical Considerations for Data Mining Researchers. In *Technical Report HPL-2003-4*, HP Labs.
- G. Finlayson, G. Schaefer (2001). Hue that is Invariant to Brightness and Gamma. In *Proc. British Machine Vision Conf.*, pp. 303–312, Manchester, England.
- M. M. Fleck, D. A. Forsyth, C. Bregler (1996). Finding Naked People. In *ECCV(2)*, pp. 593–602.

- J. Fridrich (1999). Robust Bit Extraction from Images. In *IEEE International Conference On Multimedia Computing And Systems*, IEEE, Florence.
- J. Fridrich (2000). Visual Hash for Oblivious Watermarking. In *SPIE Photonic West Electronic Imaging, Security and Watermarking of Multimedia Contents Conference*, San-Jose.
- V. Gaede, O. Guenther (1998). Multidimensional access methods. *ACM Comput. Surv.* **30**(2):170–231.
- A. Gionis, P. Indyk, R. Motwani (1999). Similarity Search in High Dimensions via Hashing. In *25th International Conference on Very Large Data Bases*, pp. 518–529, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- R. Gonzalez, R. Woods (2002). *Digital Image Processing, 2/E*. Prentice-Hall.
- C. Gotlieb, H. Kreszig (1994). Texture descriptors based on co-occurrence matrices. *Computer vision, graphics and image processing* **51**:70–86.
- M. Grabner, H. Grabner, H. Bischof (2006). Fast approximated SIFT. In *Lecture Notes in Computer Science*, vol. 3851, pp. 918–927, Springer.
- A. Gross, L. Latecki (1995). Digitizations Preserving Topological and Differential Geometric Properties. *Computer Vision and Image Understanding: CVIU* **62**(3):370–381.
- A. Guttman (1984). R-trees: a dynamic index structure for spatial searching. In *ACM SIGMOD international conference on Management of data*, pp. 47–57, ACM Press, New York, NY, USA.
- R. Haralick, K. Shanmugam, I. Dinstein (1973). Texture features for image classification. *IEEE Transaction on System, Man, and Cybernetic* **3**(6):610–621.
- C. Harris, M. Stephens (1988). A Combined Corner and Edge Detector. In *Proceedings of The Fourth Alvey Vision Conference*, pp. 147–151, Manchester.
- F. Hartung, M. Kutter (1999). Multimedia watermarking techniques. *Proceedings of the IEEE* **87**(7):1079 – 1107.
- C. Herley (2002). Why watermarking is nonsense. *Signal Processing Magazine, IEEE* **19**(5):10–11.
- C. W. Hsu, C. C. Chang, C. J. Lin (2003). A Practical Guide to Support Vector Classification. <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.
- C.-Y. Hsu, C.-S. Lu (2004). Geometric distortion-resilient image hashing system and its application scalability. In *Proceedings of the 2004 workshop on Multimedia and security*, pp. 81–92, New York.
- M.-K. Hu (1962). Visual Pattern Recognition by Moment Invariants. *IEEE Transaction on Information Theory* **8**:179–187.
- A. Hyvarinen, E. Oja (2000). Independent Component Analysis: Algorithms and Applications. *Neural Networks* **13**(4-5):411–430.

- P. Indyk, R. Motwani (1998). Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the ACM symposium on Theory of computing*, pp. 604–613, ACM Press, New York, NY, USA.
- J. Jackson (1991). *A User's Guide to Principal Components*.
- M. Johnson, K. Ramchandran (2003). Dither-based secure image hashing using distributed coding. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, pp. II-751–4 vol.3.
- T. Kalker, J. Haitisma, J. Oostveen (2001). Issues with digital watermarking and perceptual hashing. In *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 4518, pp. 189–197, Philips Research Eindhoven, Prof. Holstlaan 4, 5656 AA, Eindhoven, Netherlands.
- D. Kapur, Y. Lakshman, T. Saxena (1995). Computing invariants using elimination methods. In *Computer Vision, 1995. Proceedings., International Symposium on*, pp. 97–102.
- J. Karhunen, J. Joutsensalo (1994). Representation and separation of signals using nonlinear PCA type learning. *Neural Netw.* **7**(1):113–127.
- Y. Ke, R. Sukthankar (2004). PCA-SIFT: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2, pp. II-506–II-513 Vol.2.
- Y. Ke, R. Sukthankar, L. Huston (2004). Efficient Nearduplicate Detection and Subimage Retrieval. In *ACM International Conference on Multimedia*, pp. 869–876.
- S. S. Keerthi, C.-J. Lin (2003). Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Computation* **15**(7):1667–1689.
- C. Kim (2003). Content-based image copy detection. *Signal Processing: Image Communication* **18**(3):169–184.
- H.-P. Kriegel (1984). Performance comparison of index structures for multi-key retrieval. In *SIGMOD '84: Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pp. 186–196, ACM Press, New York, NY, USA.
- N. Kwak, c.-H. Choi (2003). Feature extraction based on ICA for binary classification problems. *IEEE Transactions on Knowledge and Data Engineering* **15**(6):1374–1388.
- D. T. Lee, B. J. Schachter (1980). Two algorithms for constructing a Delaunay triangulation. *International Journal of Parallel Programming* **V9**(3):219–242.
- F. Lefèbvre, B. Macq, J.-D. Legat (2002). RASH: RAdon Soft Hash algorithm. In *EURASIP European Signal Processing Conference*, France.
- F. Lefebvre, J. Czyz, B. Macq (2003). A robust soft hash algorithm for digital image signature. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 2, pp. II-495–8 vol.3.

- H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, L. Amsaleg (2006a). Scalability of local image descriptors: a comparative study. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pp. 589–598, ACM Press, New York, NY, USA.
- H. Lejsek, F. H. Ásmundsson, B. Thór-Jónsson, L. Amsaleg (2006b). Blazingly Fast Image Copyright Enforcement. In *ACM International Conference on Multimedia, demonstration*, pp. 489–490, Santa Barbara, CA, USA.
- J.-L. Leu (1991). Computing a Shape's Moments from its Boundary. *Pattern Recognition* **24**(10):949–957.
- B. Li, E. Chang, C.-T. Wu (2002). DPF - a perceptual distance function for image retrieval. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, vol. 2, pp. II-597–II-600 vol.2.
- T. Lindeberg, J.-O. Eklundh (1992). Scale-space primal sketch: construction and experiments. *Image Vision Comput.* **10**(1):3–18.
- D. Lowe (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision* **60**(2):91–110.
- C.-S. Lu, C.-Y. Hsu (2005). Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. *Multimedia Systems* **V11**(2):159–173.
- C.-S. Lu, C.-Y. Hsu, S.-W. Sun, P.-C. Chang (2004). Robust mesh-based hashing for copy detection and tracing of images. In *Multimedia and Expo 2004*, vol. 1, pp. 731–734 Vol.1.
- B. Manjunath, W. Ma (1996). Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(8):837–842.
- B. Manjunath, J.-R. Ohm, V. Vasudevan, A. Yamada (2001). Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on* **11**(6):703–715.
- Y. Maret, F. Dufaux, T. Ebrahimi (2005a). Image Replica Detection based on Support Vector Classifier. In *Proc. SPIE Applications of Digital Image Processing XXVIII*, Santa Barbara, USA.
- Y. Maret, F. Dufaux, T. Ebrahimi (2006a). Adaptive Image Replica Detection based on Support Vector Classifiers. *Signal Processing : Image Communication* **21**(8):688–703.
- Y. Maret, G. Molina, T. Ebrahimi (2005b). Identification of Image Variations based on Equivalence Classes. In *Proc. SPIE Visual Communications and Image Processing*, SPIE, Beijing, China.
- Y. Maret, G. G. Molina, T. Ebrahimi (2005c). Images Identification Based on Equivalence Classes. In *Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS 2005)*.
- Y. Maret, D. M. Sanjuan, F. Dufaux, T. Ebrahimi (2006b). Hierarchical Indexing using R-trees for Replica Detection. In *SPIE, SPIE, SPIE*.

- Y. Maret *et al.* (2006c). A Novel Replica Detection System using Binary Classifiers, R-trees, and PCA. In *International Conference on Image Processing*, IEEE.
- A. Martin *et al.* (1997). The DET Curve in Assessment of Detection Task Performance. In *Proc. Eurospeech '97*, pp. 1895–1898, Rhodes, Greece.
- D. Medin, R. Goldstone, D. Gentner (1993). Respects for Similarity. *Psychological Review* **100**(2):254–278.
- M. Mihçak, R. Venkatesan (2001). New Iterative Geometric Methods for Robust Perceptual Image Hashing. In *ACM Workshop on Security and Privacy in Digital Rights Management*, Philadelphia.
- V. Monga, A. Banerjee, B. Evans (2004). Clustering Algorithm for Perceptual Image Hashing. In *IEEE International Conference on Image Processing*, Singapore.
- V. Monga, A. Banerjee, B. Evans (2006). A clustering based approach to perceptual image hashing. *Information Forensics and Security, IEEE Transactions on* **1**(1):68–79.
- V. Monga, B. L. Evans (2004). Robust Perceptual Image Hashing Using Feature Points. In *IEEE International Conference on Image Processing*, Singapore.
- P. Moulin (2003). Comments on “Why watermarking is nonsense”. *Signal Processing Magazine, IEEE* **20**(6):57–59.
- MPEG12816 (2006). Local region descriptor robust invariant geometrical transformation.
- MPEG12841 (2006). Proposed test conditions for MPEG-7 Visual Core Experiments 6.
- MPEG13152 (2006). Visual identifier which is robust and invariant to image modification.
- MPEG13579 (2006). The non-geometric model of local region descriptor as the visual identifier.
- MPEG13861 (2006). A Test Image Management System for MPEG-7 core experiments.
- K. Muller *et al.* (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks* **12**(2):181–201.
- W. Niblack *et al.* (1993). QBIC project: querying images by content, using color, texture, and shape. In W. Niblack (ed.), *Proc. SPIE Vol. 1908, p. 173-187, Storage and Retrieval for Image and Video Databases, Wayne Niblack; Ed.*, pp. 173–187.
- J. Nievergelt, H. Hinterberger, K. C. Sevcik (1984). The Grid File: An Adaptable, Symmetric Multikey File Structure. *ACM Trans. Database Syst.* **9**(1):38–71.
- L. Penna, A. Clark, G. Mohay (2005). Challenges of automating the detection of paedophile activity on the Internet. In *Systematic Approaches to Digital Forensic Engineering, 2005. First International Workshop on*, pp. 206–220.

- F. A. P. Petitcolas, M. Kutter (2001). Fair Evaluation Methods for Image Watermarking Systems. *Journal of Electronic Imaging* **9**(4):445–455.
- J. Platt (2000). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In B. S. D. S. A.J. Smola, P. Bartlett (ed.), *Advances in Large Margin Classifiers*, pp. 61–74.
- A. Qamra, Y. Meng, E. Chang (2005). Enhanced Perceptual Distance Functions and Indexing for Image Replica Recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **27**(3):379–391.
- Y. Rui, T. Huang, S. Chang (1999). Image retrieval: current techniques, promising directions and open issues. *Journal of Visual Communication and Image Representation* **10**(4):39–62.
- Y. Rui, A. She, T. Huang (1996). Modified fourier descriptors for shape representation – a practical approach. In *Proceeding of First International Workshop on Image Databases and Multi Media Search*.
- Russell, Sinha (2002). A perceptual comparison of image similarity metrics. *Journal of Vision* **2**(7):679–679.
- B. Schoelkopf, A. Smola, R. Williamson, P. Bartlett (2000). New support vector algorithms. *Neural Networks* **22**:1083–1121.
- B. Seeger, H.-P. Kriegel (1990). The buddy tree: an efficient and robust access method for spatial data base. In *Proceedings of the sixteenth international conference on Very large databases*, pp. 590–601, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- T. Sellis, N. Roussopoulos, C. Faloutsos (1987). The R⁺-Tree: A Dynamic Index for Multi-Dimensional Objects. In *The VLDB Journal*, pp. 507–518.
- J. Seo, J. Haitsma, T. Kalker, C. Yoo (2003). Affine Transform Resilient Image Fingerprinting. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Hong Kong.
- J. S. Seo, J. Haitsma, T. Kalker, C. D. Yoo (2004). A robust image fingerprinting system using the Radon transform. *Signal Processing: Image Communication* **19**(4):325–339.
- P. Simard, Y. LeCun, J. Denker, B. Victorri (1998). Transformation Invariance in Pattern Recognition-Tangent Distance and Tangent Propagation. In *Neural Networks: Tricks of the Trade, this book is an outgrowth of a 1996 NIPS workshop*, pp. 239–27, Springer-Verlag, London, UK.
- A. Smeulders *et al.* (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(12):1349–1380.
- J. Smith, S.-F. Chang (1994). Transform features for texture classification and discrimination in large image databases. In *IEEE International Conference on Image Processing*, vol. 3, pp. 407–411 vol.3.

- J. Smith, A. Natsev (2002). Spatial and Feature Normalization for Content Based Retrieval. In *IEEE International Conference Multimedia and Expositions*, vol. 1, pp. 193–196.
- J. R. Smith, S.-F. Chang (1995). Single color extraction and image query. In *ICIP '95: Proceedings of the 1995 International Conference on Image Processing (Vol. 3)-Volume 3*, p. 3528, IEEE Computer Society, Washington, DC, USA.
- I. Steinwart (2003). On the Optimal Parameter Choice for ν -Support Vector Machines. *IEEE Transaction on Pattern Analysis and Machine Intelligence* **25**(10):1274–1284.
- M. Still (2005). *The Definitive Guide to ImageMagick (Definitive Guide)*. Apress, Berkely, CA, USA.
- D. Stinson (2002). *Cryptography: Theory and Practice*. Chapman and Hall/CRC, 2nd edn.
- M. A. Stricker, M. Orengo (1995). Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pp. 381–392.
- M. Swain, D. Ballard (1990). Indexing via color histograms. In *Computer Vision*, pp. 390–393.
- H. Tamura, N. Yokoya, T. Yamawaki (1978). Texture features corresponding to visual perception. *IEEE transaction on System, Man and Cybernetic* **8**(6):460–473.
- D. Tax, R. Duin (2001). Uniform object generation for optimizing one-class classifiers. *Journal of Machine Learning Research* **2**:155–173.
- D. M. Tax, R. P. Duin (2004). Support Vector Data Description. *Machine Learning* **55**:45–66.
- A. Tversky (1977). Features of Similarity. *Psychological Review* **84**(4):327–352.
- V. N. Vapnik (2000). *The Nature of Statistical Learning Theory*. Springer.
- R. Venkatesan, M.-H. Jakubowski (2000). Image Hashing. In *DIMACS Workshop on Protection of Intellectual Property*, New Brunswick.
- R. Venkatesan, S.-M. Koon, M.-H. Jakubowski, P. Moulin (2000). Robust Image Hashing. In *IEEE International Conference on Image Processing*, Vancouver.
- P. Viola, M. Jones (2001). Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I-511 – I-518.
- B. Wang, Z. Li, W.-Y. Li, Mingjing Ma (2006). Large-Scale Duplicate Detection For Web Image Search. In *IEEE International Conference on Multimedia and Expo*, pp. 353–356.
- J. Z. Wang, G. Wiederhold, O. Firschein (1997). System for Screening Objectionable Images Using Daubechies' Wavelets and Color Histograms. In *IDMS '97: Proceedings of the 4th International Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, pp. 20–30, Springer-Verlag, London, UK.

-
- T.-F. Wu, C.-J. Lin, R. C. Weng (2004). Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research* **5**:975–1005.
- B. Zadrozny, C. Elkan (2002). Transforming classifier scores into accurate multiclass probability estimates. In *ACM international conference on Knowledge discovery and data mining*, pp. 694–699, ACM Press, New York, NY, USA.
- C. Zahn, R. Roskies (1972). Fourier Descriptors for Plane Closed Curves. *IEEE Transactions on Computers* **21**(3):269–281.
- P. Zwicke, Z. Kiss (1983). A new implementation of the Mellin transform and its application to radar classification. *IEEE Transaction on Pattern Analysis Machine Intelligence* **5**(2):191–199.

Curriculum Vitæ

Name: Yannick MARET
Citizenship: Swiss
Birthdate: December 31st, 1975
Birthplace: Sion, Switzerland
Marital status: Married to Maria Del Carmen



Contact information

Address: Rue centrale 24
CH-1994 Aproz
Phone: +41.79.389.99.53
Email: yannick.maret@a3.epfl.ch

Research and development interests

My current research concerns the *analysis of images* in order to detect modified duplicates of valuable images. To achieve this, I use a wide range of techniques such as *image processing*, *machine learning*, or *object indexing*. Furthermore, I am interested in many other aspects of computer sciences, such as 3D processing or hardware implementations (analogue or digital) of complex algorithms.

Work experience

- **May 2007 – present:** Scientist at ABB Switzerland Ltd, Corporate Research, Daettwil, Switzerland
 - Technology survey in analogue electronics
 - Research and development in analogue electronics for sensors
 - Apprentices and students supervision
 - Liaison with universities in the domain of analogue electronics

- **May 2003 – April 2007:** Research assistant at the Signal Processing Institute, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland
 - Research and development on image duplicates detection
 - Author or co-author of eleven peer-reviewed publications
 - Responsible for EPFL contributions to EC-funded projects: *VISNET 1/2* and *K-Space*
 - Co-organisation of international conferences and meetings
 - Advisor of four M.S. students
 - Teaching assistant, lectures on *Image and Video Processing* and *Multimedia Security*
- **1999 — 2001:** Engineer, Haute Ecole Valaisanne Sion, Switzerland
 - Nine months fulltime between 1999 and 2001
 - Design, implementation, and documentation of a PCI core in VHDL
- **1999 — 2001:** Engineer, Institut de Recherche en Ophtalmologie Sion, Switzerland
 - Six months fulltime between 1999 and 2001
 - Analysis of fundus signals by means of wavelet
 - Development of the mechanical and optical parts of an infrared camera

Education

- **October 2003 — present:** *Ph. D. student* in image processing. Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
 - Specialisation in image analysis and machine learning
 - Expected graduation date: May 2007
 - Research Topic: *Efficient Image Duplicates Detection Based on Image Analysis*
- **1999 — 2003:** *M.S., Electrical Engineering* at Swiss Federal Institute of Technology Lausanne, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland.
 - Specialisation in signal processing, and process control
 - Cumulative grad-point average: 5.42 / 6
 - Master thesis done at Universidad Politécnica de Madrid
 - Thesis title: *Implementation of MPEG-4's Subdivision Surfaces Tools*
- **1995 — 1999:** *B.E., Electrical Engineering* at Haute Ecole Valaisanne, Sion, Switzerland.
 - Specialisation in embedded system, computer system, electronic
 - Thesis title: *Analysis of Speech Signals by Means of Wavelets*

Awards

- Recipient of CASC award for an exceptional bachelor thesis
- Invited lecturer at the Conference on Cryptology and Digital Content Security, May 2007

Skills

Languages

- French: mother tongue
- English: fluent, Cambridge Certificate of Proficiency in English (grade B) in 2006
- Spanish: good knowledge, spoken and written
- German: basic knowledge, spoken and written

Computer literacy

- Programming: C/C++, Java, Matlab, HTML, PHP, mySQL, Assembly, VHDL
- Applications: MS Visual Studio, Eclipse, Matlab, Mentor Graphic VHDL, MS Office
- Operating Systems: MS Windows, Linux/Unix, Mac OS X
- Version control: CVS (client) and SVN (client and server)

Extra-curricular activities

My preferred sports are rock climbing, and chess (I captain an Internet-based team). Moreover, I enjoy reading books or watching films based on historical facts. Additionally, I have a passion for science and technology, especially astronomy and computer hardware. Finally, I like being with friends or family and simply having a good time.

Personal Publications

Journal paper

- Y. Maret, F. Dufaux and T. Ebrahimi, *Adaptive Image Replica Detection based on Support Vector Classifiers*, Signal Processing : Image Communication, Vol. 21, No 8, pp. 688-703, September 2006

Chapter in books

- T. Ebrahimi and Y. Maret, *Stéganographie*, Enjeux de la sécurité multimédia, Hermes — Collection Informatique, pp. 173-186, 2006

Conference papers

- D. Marimon Sanjuan, Y. Maret, Y. Abdeljaoued and T. Ebrahimi, *Particle filter-based camera tracker fusing marker- and feature point-based cues*, Proceedings of the SPIE Conference on Visual Communications and Image Processing, January 2007
- Y. Maret, S. Nikolopoulos, F. Dufaux, T. Ebrahimi and N. Nikolaidis, *A Novel Replica Detection System using Binary Classifiers, R-trees, and PCA*, Proceedings of the IEEE International Conference on Image Processing, October 2006
- Y. Maret, D. Marimon Sanjuan, F. Dufaux and T. Ebrahimi, *Hierarchical Indexing using R-trees for Replica Detection*, Proceedings of the SPIE Applications of Digital Image Processing Conference, August 2006
- Y. Maret, S. Nikolopoulos, F. Dufaux, C. Costace, T. Ebrahimi and N. Nikolaidis, *Reduced Complexity Replica Detection Systems using Binary Classifiers and R-trees*, Proceedings of International Workshop on Immersive Communication and Broadcast Systems, October 2005
- Y. Maret, F. Dufaux and T. Ebrahimi, *Image Replica Detection based on Support Vector Classifier*, Proceedings of the SPIE Applications of Digital Image Processing Conference, August 2005

- Y. Maret, G. Garcia Molina and T. Ebrahimi, *Identification of Image Variations based on Equivalence Classes*, Proceedings of the SPIE Conference on Visual Communications and Image Processing, July 2005
- Y. Maret, G. Garcia Molina and T. Ebrahimi, *Images Identification Based on Equivalence Classes*, Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services, April 2005
- Y. Maret and T. Ebrahimi, *Data Hiding on 3D Polygonal Meshes*, Proceedings of the ACM Multimedia and Security Workshop, pp. 68-74, September 2004
- N. Aspert, E. Drelic Gelasca, Y. Maret and T. Ebrahimi, *Steganography for 3D Polygonal Meshes*, Proceedings of the SPIE Applications of Digital Image Processing Conference, July 2002