EXTENDED BAG-OF-WORDS FORMALISM FOR IMAGE CLASSIFICATION

Sandra Avila

▶ **To cite this version:**

Sandra Avila. EXTENDED BAG-OF-WORDS FORMALISM FOR IMAGE CLASSIFICATION. Computer Vision and Pattern Recognition [cs.CV]. Université Pierre et Marie Curie - Paris VI, 2013. English. NNT : . tel-00958547

HAL Id: tel-00958547
https://theses.hal.science/tel-00958547

Submitted on 12 Mar 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT DE
L'UNIVERSITÉ PIERRE ET MARIE CURIE
ET L'UNIVERSITÉ FÉDÉRALE DE MINAS GERAIS
(Spécialité: Informatique)

L'École Doctorale Informatique, Télécommunications et Électronique

Présentée par
SANDRA ELIZA FONTES DE AVILA

Pour obtenir le grade de
DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE
et de l'UNIVERSITÉ FÉDÉRALE DE MINAS GERAIS

EXTENDED BAG-OF-WORDS FORMALISM FOR IMAGE CLASSIFICATION

(EXTENSION DU MODÈLE PAR SAC DE MOTS VISUELS POUR LA CLASSIFICATION D'IMAGES)

Soutenue le 14 juin 2013 devant le jury composé de:

Florent PERRONNIN	Xerox Research Centre Europe	*Rapporteur*
Mario CAMPOS	Université Fédérale de Minas Gerais	*Rapporteur*
Cordelia SCHMID	INRIA Grenoble	*Examinateur*
Patrick PÉREZ	Technicolor Research Innovation	*Examinateur*
Patrick GALLINARI	Université Pierre et Marie Curie	*Examinateur*
Nicolas THOME	Université Pierre et Marie Curie	*Encadrant*
Matthieu CORD	Université Pierre et Marie Curie	*Directeur de Thèse*
Arnaldo ARAÚJO	Université Fédérale de Minas Gerais	*Directeur de Thèse*

Resumo

Informação visual, na forma de imagens e vídeos digitais, tornou-se tão onipresente em repositórios de dados, que não pode mais ser considerada uma "cidadã de segunda classe", eclipsada por informações textuais. Neste cenário, a *classificação de imagens* tornou-se uma tarefa crítica. Em particular, a busca pela identificação automática de conceitos semânticos complexos, representados em imagens, tais como cenas ou objetos, tem motivado pesquisadores em diversas áreas como, por exemplo, Recuperação de Informação, Visão Computacional, Processamento de Imagem e Inteligência Artificial. No entanto, em contraste com os documentos de texto, cujas palavras apresentam conteúdo semântico, imagens consistem de pixels que não têm nenhuma informação semântica por si só, tornando a tarefa muito difícil.

O problema abordado nesta tese refere-se à representação de imagens com base no seu conteúdo visual. Objetiva-se a detecção de conceitos em imagens e vídeos, por meio de uma nova representação que enriquece o modelo saco de palavras visuais. Baseando-se na quantização de descritores locais discriminantes por um dicionário, e na agregação desses descritores quantizados em um vetor único, o modelo saco de palavras surgiu como uma das abordagens mais promissora para a classificação de imagens. Nesta tese, é proposto BossaNova, uma nova representação de imagens que preserva informações importantes sobre a distribuição dos descritores locais em torno de cada palavra visual.

Os resultados experimentais em diversas bases de classificação de images, tais como ImageCLEF Photo Annotation, MIRFLICKR, PASCAL VOC e 15-Scenes, mostraram a vantagem da abordagem BossaNova quando comparada às técnicas tradicionais, mesmo sem fazer uso de combinações complexas de diferentes descritores locais.

Uma extensão da representação BossaNova também foi estudada nesta tese. Trata-se da combinação da abordagem BossaNova com uma outra representação muito competitiva baseada nos vetores de Fisher. Os resultados consistemente alcançam outras representações no estado-da-arte em diversas bases de dados, demonstrando a complementaridade das duas abordagens. Este estudo resultou no segundo lugar, na competição ImageCLEF 2012 Flickr Photo Annotation Task, dentre as 28 submissões,

na categoria de informação visual.

Ademais, a representação BossaNova também foi avaliada na aplicação real de detecção de pornografia. Os resultados validaram, mais uma vez, a relevância da abordagem BossaNova em relação às técnicas tradicionais em uma aplicação real.

Abstract

Visual information, in the form of digital images and videos, has become so omnipresent in computer databases and repositories, that it can no longer be considered a "second class citizen", eclipsed by textual information. In that scenario, *image classification* has become a critical task. In particular, the pursuit of automatic identification of complex semantical concepts represented in images, such as scenes or objects, has motivated researchers in areas as diverse as Information Retrieval, Computer Vision, Image Processing and Artificial Intelligence. Nevertheless, in contrast to text documents, whose words carry semantic, images consist of pixels that have no semantic information by themselves, making the task very challenging.

In this dissertation, we have addressed the problem of representing images based on their visual information. Our aim is content-based concept detection in images and videos, with a novel representation that enriches the Bag-of-Words model. Relying on the quantization of highly discriminant local descriptors by a codebook, and the aggregation of those quantized descriptors into a single pooled feature vector, the Bag-of-Words model has emerged as the most promising approach for image classification. We propose BossaNova, a novel image representation which offers a more information-preserving pooling operation based on a distance-to-codeword distribution.

The experimental evaluations on many challenging image classification benchmarks, such as ImageCLEF Photo Annotation, MIRFLICKR, PASCAL VOC and 15-Scenes, have shown the advantage of BossaNova when compared to traditional techniques, even without using complex combinations of different local descriptors.

An extension of our approach has also been studied. It concerns the combination of BossaNova representation with another representation very competitive based on Fisher Vectors. The results consistently reaches other state-of-the-art representations in many datasets. It also experimentally demonstrate the complementarity of the two approaches. This study allowed us to achieve, in the competition ImageCLEF 2012 Flickr Photo Annotation Task, the 2nd among the 28 visual submissions.

Finally, we have explored our BossaNova representation in the challenging real-

world application of pornography detection. Once again, the results validated the relevance of our approach compared to standard techniques on a real application.

Résumé

L'information visuelle, représentée sous la forme d'images ou de vidéos numériques, est devenue si omniprésente dans le monde numérique d'aujourd'hui, qu'elle ne peut plus être considérée comme un "citoyen de seconde zone", par rapport à l'information textuelle. Néanmoins, contrairement aux documents textuels, les images sont constituées de pixels ne portant pas d'information sémantique directement accessible, ajoutant ainsi une difficulté à la tâche d'interprétation. Dans ce contexte, *la classification d'images* est devenue une tâche critique. En particulier, l'identification automatique d'objets complexes et de concepts sémantiques dans les images, a suscité de nombreux travaux récents, aussi bien en Recherche d'Information, Vision par Ordinateur, Traitement d'Image qu'en Intelligence Artificielle.

Dans cette thèse, nous traitons le problème de la représentation des images. Notre objectif est la détection de concepts à partir d'une analyse du contenu visuel des images et des vidéos. Pour cela, nous introduisons une nouvelle représentation qui enrichit le modèle classique par sacs de mots visuels. S'appuyant sur la quantification de descripteurs locaux, et l'agrégation de ces descripteurs quantifiés en un vecteur de caractéristique unique, le modèle par sacs de mots visuels a émergé comme l'approche la plus efficace pour la classification d'images. Nous proposons BossaNova, une nouvelle représentation d'images permettant de conserver plus d'information lors de l'opération d'agrégation (pooling) en exploitant la distribution des distances entre les descripteurs locaux et les mots visuels.

L'évaluation expérimentale sur plusieurs bases de données de classification d'images, telles que ImageCLEF Photo Annotation, MIRFLICKR, PASCAL VOC et 15-Scenes, a montré l'intérêt de Bossanova vis-à-vis des techniques traditionnelles, même sans utiliser de combinaisons complexes de multiples descripteurs locaux.

Une extension de notre approche a également été étudiée. Elle concerne la combinaison de BossaNova avec une autre représentation basée sur des vecteurs de Fisher très coupétitive. Les résultats obtenus sont systématiquement meilleurs atteignant l'état de l'art sur de nombreuses bases. Ils permettent ainsi de démontrer expérimentallement

la complémentarité des deux approches. Cette étude nous a permis d'obtenir la seconde place lors de notre participation à la compétition ImageCLEF 2012 Flickr Photo Annotation Task parmi les 28 soumissions sur la partie visuelle.

Enfin, nous avons appliqué notre stratégie de représentation BossaNova dans un contexte vidéo, en vue de faire de la détection de séquences à caractère pornographique. Les résultats ont permis de valider une nouvelle fois l'intérêt de notre approche par rapport à des détecteurs standards du marché sur une application réelle.

List of Figures

List of Tables

List of Acronyms

AMT	Amazon Mechanical Turk
ANOVA	Analysis of Variance
AP	Average Precision
BN	BossaNova
BoF	Bag-of-Features
BOSSA	Bag Of Statistical Sampling Analysis
BoW	Bag-of-Words
CI	Confidence Interval
CLEF	Cross Language Evaluation Forum
CM	Constellation Model
CNN	Convolutional Neural Network
DBN	Deep Belief Network
DoG	Difference-of-Gaussian
FV	Fisher Vector
GLP	Geometric ℓ_p-norm Pooling
GmAP	Geometric Mean Average Precision
GMM	Gaussian Mixture Model
GLOH	Gradient Location and Orientation Histogram
H-BoW	Hierarchical-BoW
HIT	Human Intelligence Task
HoG	Histogram of Gradient
HOG	Histograms of Oriented Gradients

ISM	Implicit Shape Model
k-NN	k Nearest Neighbors
KPCA	Kernel Principal Component Analysis
mAP	Mean Average Precision
LBG	Linde Buzo Gray algorithm
LCC	Local Coordinate Coding
LLC	Locality-constrained Linear Coding
LoG	Laplacian-of-Gaussian
MKL	Multiple Kernel Learning
NBNN	Naive Bayes Nearest Neighbor
PCA	Principal Component Analysis
QP	Quadratic Program
RBM	Restricted Boltzmann Machines
RETIN	Retrieval and Interactive Tracking of Images
RIFT	Rotation Invariant Feature Transform
SC	Sparse Coding
SFV	Spatial Fisher Vector
SGD	Stochastic Gradient Descent
SGDQN	Stochastic Gradient Descent Quasi-Newton
SIFT	Scale Invariant Feature Transformation
SMO	Sequential Minimal Optimization
SPM	Spatial Pyramid Matching
SSC	Sparse Spatial Coding
SSIM	Self-Similarity
SURF	Speeded Up Robust Features
SVC	Super-Vector Coding
SVM	Support Vector Machine
VLAD	Vector of Locally Aggregated Descriptors
VLAT	Vector of Locally Aggregated Tensors
VOC	Visual Object Classes

Contents

Chapter 1

Introduction

The growth of the Internet, the availability of cheap digital cameras, and the ubiquity of cell-phone cameras have tremendously increased the amount of accessible visual information, especially images and videos. The example given by Flickr, a photo-sharing application, is illustrative, with more than 8 billion photos hosted as of December 2012. The accelerated expansion of social networks has increased the amount of images available online even further (Table 1.1).

Table 1.1: Estimation of the numbers of photos available online from social networks and photo sharing applications[a].

220 billion	Estimated number of photos on Facebook (October 2012)[b]
300 million	Average number of photos uploaded to Facebook per day
8 billion	Photos hosted on Flickr (December 2012)
4.5 million	Average number of photos uploaded to Flickr per day
5 billion	Estimated number of photos on Instagram (September 2012)
5 million	Average number of photos uploaded to Instagram per day

[a] http://royal.pingdom.com/2013/01/16/internet-2012-in-numbers/

[b] http://gigaom.com/2012/10/17/facebook-has-220-billion-of-your-photos-to-put-on-ice/

In order to enjoy that immense and increasing collection of images, people need tools to retrieve them using semantically rich terms. One solution to that problem is manual annotation. But annotating images is a tedious task, and although there is a number of ways to provide it (*e.g.*, HTML language, EXIF meta-data, user-provided tag), most users are unwilling to perform that task in with meaningful, useful terms.

Furthermore, while we could conceive the manual semantic indexing of a *personal* image collection, hand-processing large-scale/web-scale image collections is clearly unfeasible: there is simply not enough human labor to cope with the exponential growth of those data. Crowdsourcing, either by harnessing interactions in games and social networks, either by micropayment systems like Amazon's Mechanical Turk, presents an opportunity to address part of the problem, and relatively large collections can be annotated by those means. Still, those collections represent a tiny fraction of all publicly available images on the web.

It is clear the we need some way to automatically annotate the images, or at least to propagate labels from those small annotated collections to arbitrarily large unlabeled ones. The challenge, however, is that the low-level image representation (*i.e.*, the pixels) provide no clue about its semantic concepts. Smeulders et al. [2000] call the absence of this relationship "*semantic gap*".

In order to bypass this problem, we can employ Machine Learning to create statistical models that relate image content to the semantic annotations. In that way, we can employ the small annotated collections as *training sets* for models meant to propagate the labels for an arbitrary number of images. Therefore, although manual indexing of images cannot provide a direct solution to the problem, it is still one of the crucial steps for that solution.

Another critical step in the solution is the extraction of adequate features from the images, which are used to characterize the visual content. That "relevant" information depends on the task. For example, features based on color would be able to differentiate between some concepts, such as *night* scenes and *sunset* scenes (see Figure 1.1), but they would not be able to distinguish an *adult person* from an *old person*. Therefore, a feature vector such as a color histogram, can be adequate for some specific tasks without being able to solve the general problem.

In fact, bridging the semantic gap is probably too difficult for the usual, simple image descriptors, such as color histograms, texture descriptors and simple shape descriptors. Recently, more elaborated image representations, known as *mid-level* representations (*i.e.*, richer representations of intermediate complexity), have been proposed to deal with the complexity of the task, by aggregating hundreds, and even thousands of low-level local descriptions about the image into a single feature vector. Exploring those mid-level representations is the subject of this dissertation.

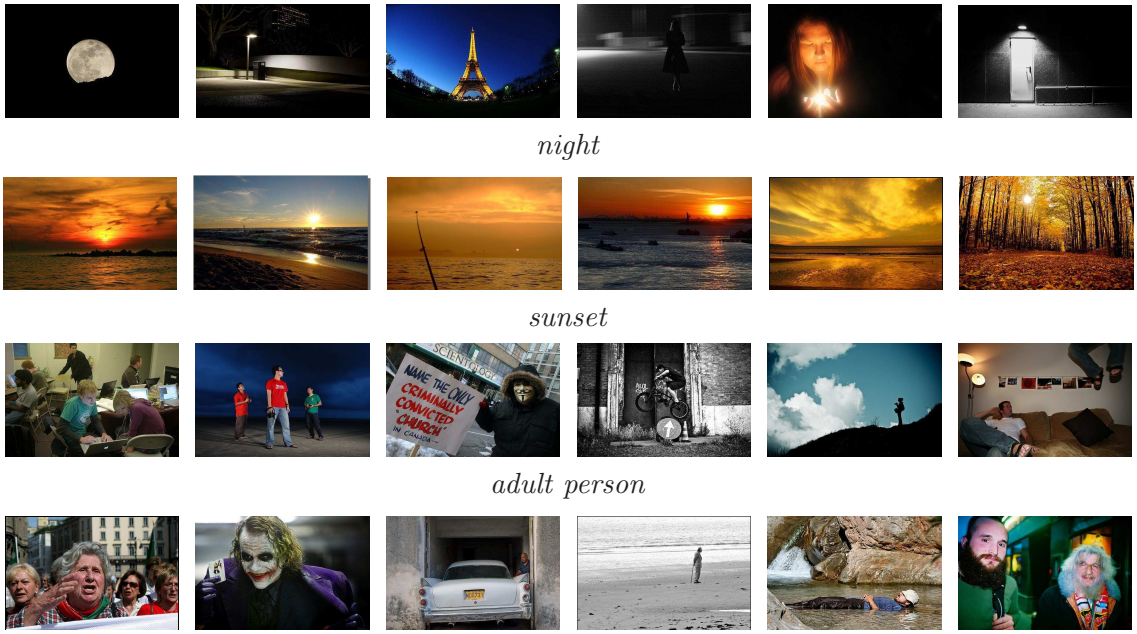


Figure 1.1: Example images from ImageCLEF 2011 Photo Annotation dataset [Nowak et al., 2011]. Simple features, such as color histograms, are able to differentiate between some concepts, such as *night scenes* and *sunset scenes*, but they cannot solve the general problem of distinguish an *adult person* from an *old person*.

1.1 Motivation

Recognizing categories of objects and scenes is a fundamental human ability and an important, yet elusive, goal for computer vision research. One of the challenges lies in the large scale of the semantic space. In particular, humans can recognize visually more than 10,000 of object classes and scenes [Biederman, 1995]. For humans, visual recognition is fast, effortless and robust with respect to viewpoint, lighting conditions, occlusion and clutter. Also, learning new categories requires minimal supervision and a handful of examples [Biederman, 1987; Thorpe et al., 1996]. Achieving this level of performance in a machine would enable a great number of useful applications, such as:

Web Search: Internet image search engines are currently the only way to find images on the Internet. Their erratic performance is due to their excessive reliance on textual metadata associated to the image, or contextual text close to the images in the web pages, rather than the actual image content. As image recognition becomes more precise, search by visual content will become a more reliable alternative. As it is difficult to specify a visual query directly, the user might select a few examples, similar in nature to the desired image. Robust and efficient

classification methods can greatly help such application.

Personal Photo Search: The task of organizing and managing a personal photographic collection becomes more difficult as the collection increases in size. Many people have thousands of photos on their computers which are only loosely organized. Searching a photo in a collection requires much effort and is a time-consuming activity.

Surveillance: Video-surveillance has become a key aspect of public safety: most large cities have thousands of close-circuit TV cameras. Currently, the visual flow from those cameras must currently be scrutinized by human operators. Automated surveillance systems could provide an interesting aid or alternative to those operators, because they do not "get tired" or "distracted". If they are to be useful, those systems must detect and track objects/people, classify those objects and identify suspicious activities, and perform well in crowded environments, *e.g.*, stadiums and airports.

Biometrics: Biometric systems are, essentially, pattern recognition systems used to detect and recognize a person for security purposes, using their physiological and/or behavioral characteristics. Examples of physiological characteristics are images of the face, fingerprint, iris, and palm; examples of behavioral characteristics are dynamics of signature, voice, gait, and key strokes.

Robot Vision: The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks, such as object recognition, navigation, visual servoing for object tracking and manipulation. Real-time processing of visual content would enable robots to quickly infer the world around and make them useful for a variety of situations. In that way, a completely autonomous robot specialized to recognize certain objects would be able to substitute humans in hostile environments, *e.g.*, underwater exploration.

While there are many applications for recognition, no practical solution exist. That is due to the challenges inherent to the problem, which will be introduced next.

1.2 Challenges

Successful approaches to image classification must address a variety of issues: viewpoint and illumination changes, partial occlusion, background clutter, large intra-class visual

diversity, and visual similarity between different classes. We discuss those challenges in the following:

Viewpoint: Objects pose can suffer many transformations (*e.g.*, translation, rotation, scaling) that significantly change their appearance in the images. Even rigid objects, like airplanes (Figure 1.2(a)), appear considerably differently according to viewpoint.

Illumination: Changes of illumination causes large variations in pixel intensity values. The change can be a shift or scaling of the pixel values or, if the light source changes position, a non-linear transformation, complicated by objects' proper and cast shadows (Figure 1.2(b)).

Occlusion: Some parts of the target object may be hidden by other objects present in the scene. Additionally, self-occlusion will almost always occur, since most objects are opaque, and not all parts of the object will be visible at once. Figure 1.2(c)) illustrates that challenge.

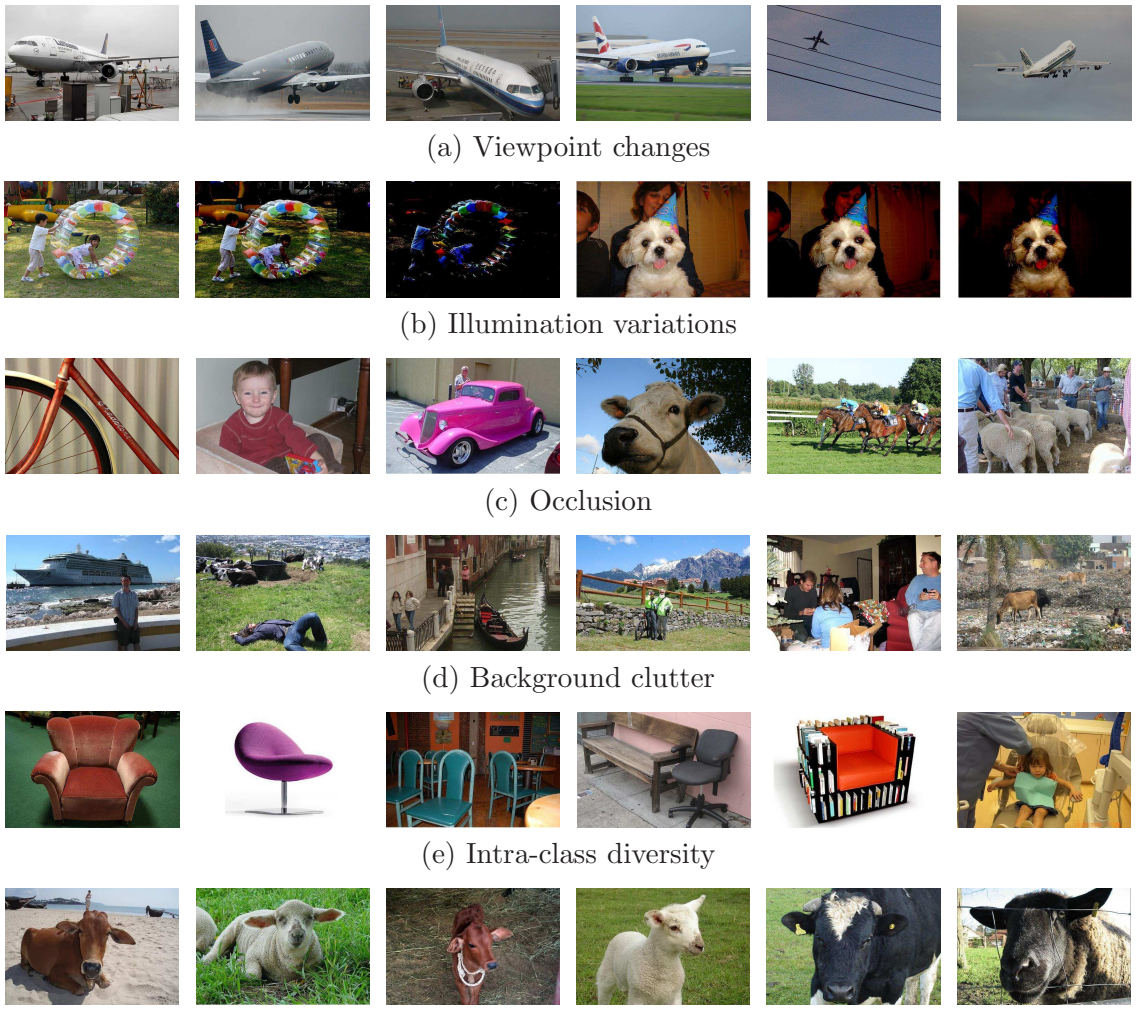
Background Clutter: Typically, images contain many more objects in addition to the one of interest. Those background objects may confound the detection, especially because we cannot assume that the target object is clearly separated from the background (Figure 1.2(d)).

Intra-class Diversity: The category of interest might have a large degree of visual variability, in the geometry, appearance, texture and so on. Even a seemingly simple concept, like "chair" (Figure 1.2(e)) may manifest huge visual diversity.

Inter-class Similarity: Conversely, a category might have similar appearance/structure to other categories, at least for some viewpoints. *E.g.*, for some viewpoints cows and sheep (Figure 1.2(f)) may be very similar.

1.3 Hypotheses

The canonical mid-level model is the Bag-of-Words (BoW) model. BoW for images was inspired from the homonymous model for text retrieval [Baeza-Yates and Ribeiro-Neto, 1999], where a document is represented by a set of words, disregarding structural aspects. In the case of images [Sivic and Zisserman, 2003] the "visual words" come from a "dictionary" induced by quantizing the feature space of a low-level local descriptor



(a) Viewpoint changes

(b) Illumination variations

(c) Occlusion

(d) Background clutter

(e) Intra-class diversity

cow *sheep* *cow* *sheep* *cow* *sheep*

(f) Inter-class similarity

Figure 1.2: Illustration of several challenges which makes the image classification problem much harder, namely (a) viewpoint changes, (b) illumination variations, (c) occlusion, (d) background clutter, (e) intra-class variation and (f) inter-class similarity. Images from PASCAL VOC 2007 dataset.

(*e.g.*, SIFT [Lowe, 2004], SURF [Bay et al., 2008]). The classical image BoW consists of a histogram of the occurrences of those visual words in each image.

BoW models (the classical one described above, and the many extensions that followed) can be understood as the application of two critical steps [Boureau et al., 2010a]: *coding* and *pooling*. The coding step quantizes the image local features according to a codebook or dictionary[1]. The pooling step summarizes the codes obtained into a single feature vector. In the classical BoW, the coding step simply associates

[1]The codebook (or visual dictionary) is usually built by clustering a set of local descriptors. It can be defined by the set of codewords (or visual words) corresponding to the centroids of clusters.

the image local descriptors to the closest element in the codebook (this is called *hard-assignment coding*), and the pooling takes the average of those codes over the entire image (this is called *average-pooling*).

A hypothesis often found in the literature, and central to this dissertation, is that *the choice of coding and pooling functions has a huge impact on the performance of BoW representations.*

Concerning that hypothesis, we focus more specifically on the pooling step. In general, the objective of pooling is to summarize a set of encoded features into a more usable representation based on a single feature vector, which should preserve important information while discarding irrelevant detail [Boureau et al., 2010b]. The crux of the matter is to balance the invariance obtained and the ambiguity introduced by the pooling function. Invariance to different backgrounds or object positioning is obtained because the codewords will be activated despite the precise positioning of the descriptors. However, since all activations are combined, ambiguities can arise if different concepts represented in the image (*e.g.*, a person and a car) end up activating sets of codewords that overlap too much. If that confusion happens, the following step of classification will have difficulty separating the concepts.

One way to mitigate that problem is to preserve more information about the encoded descriptors during the pooling step. Instead of compacting all information pertaining to a codeword into a single scalar, as performed by classical BoW representations, more detailed information can be kept.

Our main hypothesis is that *a density function-based pooling strategy allows us to better represent the links between visual codewords and low-level local descriptors in the image signature.*

Also, secondary hypotheses are considered and validated in this dissertation:

- *Soft-assignment coding, with the density function-based pooling strategy, is relevant to obtain effective mid-level image representations.* Instead of using hard-assignment coding, some weight may be given to related codewords. The soft coding enjoys computational efficiency and conceptual simplicity [Liu et al., 2011a].

- *The normalization of mid-level image representation (obtained from the density function-based pooling strategy) has important impact on classification tasks.*

Formally, the dissertation problem statement can be formulated as follows.

Given an image, how to represent its visual content information for a classification task?

1.4 Contributions

The main contribution of this dissertation is the development of a mid-level image representation for classification tasks. By analyzing the BoW model, we pointed out the weaknesses of the standard pooling operation. Thus, we propose the BossaNova representation [Avila et al., 2011, 2013], which offers a more information-preserving pooling operation based on a distance-to-codeword distribution.

In order to accomplish that goal, BossaNova departs from the fully parametric models commonly found in the literature (*e.g.*, [Perronnin et al., 2010c; Zhou et al., 2010; Krapac et al., 2011]), by employing histograms. That density-based approach allows us to conciliate the need to preserve low-level descriptor information and keeping the mid-level feature vector at a reasonable size (see Chapter 4).

Another contribution is the empirical comparison of our approach against state-of-the-art representations based on the BoW model for classification tasks. The experimental evaluations on many challenging image classification benchmarks, such as ImageCLEF Photo Annotation, MIRFLICKR, PASCAL VOC and 15-Scenes, have shown the advantage of BossaNova when compared to traditional techniques. Moreover, our participation at the competition ImageCLEF 2012 has achieved the 2nd rank among the 28 visual submissions and 13 teams [Avila et al., 2012] (see Chapter 5).

Finally, there is the empirical evaluation of our BossaNova representation in the specialized task of pornography detection, and the development of our own Pornography dataset to support this task [Avila et al., 2011, 2013] (see Chapter 6).

This dissertation has led to one refereed journal, four refereed international conference papers and three refereed Brazilian conference papers (see Chapter 7).

1.5 Outline

The remainder of the text is organized as follows.

Chapter 2 – Literature Review We establish the foundations of the work. We review the typical image classification pipeline: (i) low-level visual feature extraction, (ii) mid-level feature extraction (in particular, the BoW representation and extensions), and (iii) supervised classification. We also discuss several approaches for image classification.

Chapter 3 – Challenges and Benchmarks Addressed We introduce the variety of benchmark image datasets used in the dissertation. We detail each dataset and discuss how they differ from one another.

Chapter 4 – BossaNova Representation We give a detailed description of our BossaNova representation, which is based on a new pooling strategy. We investigate the complementarity of BossaNova and Fisher Vector representations. We also present a generative formulation of our BossaNova strategy.

Chapter 5 – Experimental Results We analyze our empirical results, comparing BossaNova performance with state-of-the-art methods in several datasets, validating its enhancements over the previously proposed BOSSA representation, and studying its behavior as its key parameters change.

Chapter 6 – Application: Pornography Detection We explore BossaNova in the real-world application of pornography detection, which because of its high-level conceptual nature, involves large intra-class variability.

Chapter 7 – Conclusions and Perspectives We present our concluding remarks and discuss future work directions.

Chapter 2

Literature Review

Visual information, in the form of digital images and videos, has become so omnipresent in computer databases and repositories, that it can no longer be considered a "second class citizen", eclipsed by textual information. In that scenario, *image classification* has become a critical task. In particular, the pursuit of automatic identification of complex semantical concepts represented in images, such as scenes or objects, has motivated researchers in areas as diverse as Information Retrieval, Computer Vision, Image Processing and Artificial Intelligence [Smeulders et al., 2000; Lew et al., 2006; Datta et al., 2008; Gosselin et al., 2008; Benois-Pineau et al., 2012]. Nevertheless, in contrast to text documents, whose words carry semantic, images consist of pixels that have no semantic information in themselves, making the task very challenging.

The typical image classification pipeline is composed of the following three layers: (i) *low-level* visual feature extraction, (ii) *mid-level* feature extraction, and (iii) supervised *classification*. The low-level features, extracted from the image pixels, are still purely perceptual, but aim at being invariant to viewpoint and illumination changes, partial occlusion, and affine geometrical transformations. Mid-level features aim at combining the set of local features into a global image representation of intermediate complexity. The mid-level features may be purely perceptual or they may incorporate semantic information from the classes, the former case being much more usual in the literature. Finally, the goal of supervised classification is to learn a function which assigns (discrete) labels to arbitrary images. That layer is intrinsically semantic, since the class labels must be known during the training phase. Recently, several authors have focused on improving the second layer (*i.e.*, mid-level feature extraction), which is the core subject of this dissertation.

This chapter reviews all three layers, and also briefly discusses alternative models for image classification, like models inspired by biology, or generative part-based

models. Our aim is to establish the foundations of the current dissertation, and to put it within that context. We start by presenting an overview of low-level visual feature extraction (Section 2.1). Next, we survey the literature on mid-level feature extraction, in particular we approach the Bag-of-Words representation (Section 2.2). We have dedicated an entire section to the issue of feature normalization (Section 2.3), since this is emerging in the literature as an important step for the good performance of representations. Then, we provide a brief description of machine learning algorithms for image classification (Section 2.4). Finally, we discuss alternative approaches to image classification (Section 2.5), based on biology, deep connectionist learning, and generative part-based models.

2.1 Low-level Visual Feature Extraction

Low-level visual feature extraction is the first crucial step of all image analysis procedures, aiming at extracting visual properties from certain regions of the image via pixel-level operations. According to the relative area of those regions, the extracted features are commonly referred to as global or local. Intuitively, a *global feature* is computed over the entire image, reflecting global characteristics of the image; by contrast, a *local feature* is computed over relatively small regions of the image, encoding the detailed traits within those specific areas. This section provides an overview of the low-level visual feature extraction. In particular, we focus on local features/descriptors, which are one of the main actors of the astonishing advances of visual recognition systems in the past 10 years.

Local feature extraction usually includes two distinct steps: feature detection and feature description. The former aims at finding a set of interest points, or salient regions in the image that are invariant to a range of image transformations. The latter step aims at obtaining robust local descriptors from the detected features. We start by describing some feature spaces (*i.e.*, visual properties) used by local (and global) features (Section 2.1.1). Next, we compare local with global features (Section 2.1.2). Then, we introduce some local feature detection operators (Section 2.1.3) and reference local descriptors[1] (Section 2.1.4).

[1]The terms "features" and "descriptors" are applied more or less interchangeably by different authors. Sometimes "feature" denotes an invariant/covariant element in the image (*e.g.*, an interest point), and "descriptor" denotes a numerical representation extracted from an image patch inside or around that feature. The terminology is somewhat confusing because the latter concept is often also called a "feature vector".

2.1.1 Feature Spaces

Local and global approaches are often based on universal visual properties such as color, texture, shape and edge. Many algorithms are available to extract descriptors/feature vectors based on those properties. The choice of such algorithms organizes the images in a geometry induced by the vector space of the descriptor, and the distance function chosen to compare those descriptors (*e.g.*, the Euclidean distance). Those geometries are called *feature spaces*.

We briefly review below some choices of feature spaces, according to the visual properties in which they are based:

Color is perhaps the most expressive of all visual properties [Trémeau et al., 2008]. In order to create a feature space, a color space must be chosen which might have an impact on its performance: indeed, one of the main aspects of color feature extraction is the choice of a suitable color space for a specific task. Many of the color features in the literature are based on color spaces other than standard RGB, such as YUV, HSV, XYZ, L*u*v. Examples of color-based feature spaces are *Color histograms* [Swain and Ballard, 1991], *color average descriptor* [Faloutsos et al., 1994], *color moments* [Stricker and Orengo, 1995], *color coherence vector* [Pass and Zabih, 1996], *color correlogram* [Huang et al., 1999], *Border/Interior pixel classification* [Stehling et al., 2002].

Texture is an intuitive concept, since it is easily perceived by humans, that defies the formulation of a precise definition [Tuceryan and Jain, 2000]. Texture is related to the spatial organization (or lack of it thereof) of colors and intensities, *e.g.*, the spots of a leopard, or the blades of grass in a lawn, or the grains of sand in beach. Literature demonstrates that the "definition" of texture is formulated by different people depending upon the particular application, without a consistently agreed-upon definition. Del Bimbo [1999] classified texture feature extractors into three different approaches: (i) space-based models (*e.g.*, *co-occurrence matrix* [Haralick et al., 1973], one the most traditional techniques for encoding texture information), (ii) frequency-based models (*e.g.*, *Gabor wavelet coefficients* [Manjunath and Ma, 1996]), and (iii) texture signatures (*e.g.*, [Tamura et al., 1978]).

Shape is perhaps the most "high-level" of the visual properties, being thus an important characteristic to identify and distinguish objects [Costa and Cesar Jr., 2000; Zhang and Lu, 2004]. Shape descriptors are classified into (i) boundary-based and (ii) region-based methods. This classification takes into account whether

shape features are extracted from the contour only or from the whole shape region. Many shape descriptors have been proposed [Yang et al., 2008], but often they assume that the image has been previously segmented with confidence into objects, and segmentation has proved itself a very elusive goal. Shape-based descriptors include *moments invariants* [Hu, 1962], *curvature scale space* [Mokhtarian, 1995], *signature histogram* [Ankerst et al., 1999], *shape context* [Belongie et al., 2002; Frome et al., 2004].

Edge points can be thought of as pixel locations of abrupt gray-level changes. Edges characterize object boundaries and are therefore useful for recognition of objects [Ahmad and Choi, 1999]. Thus, edge detection plays an important role in the description of images. Hence, many methods have been proposed for detecting edges in images [Ziou and Tabbone, 1998; Nadernejad et al., 2008]. Some of the earlier methods, such as the *Sobel* [Sobel, 1970] and *Prewitt* detectors [Prewitt, 1970], used local gradient operators which only detect edges having certain orientations. Since then, more sophisticated methods have been developed [Basu, 2002; Gonzalez and Woods, 2006]. For instance, the *Histogram of Oriented Gradient* (HOG) descriptor [Dalal and Triggs, 2005] counts occurrences of quantized gradient orientations in localized portions of an image.

2.1.2 Local versus Global Features

As we have mentioned before, the essential difference between local and global features refers to the relative region they describe, the former aiming at relatively small portions of the image, and the latter aiming at the entire image. Historically, though, global features have appeared first (*e.g.*, the color histogram [Swain and Ballard, 1991]) and have aimed at general-purposed image classification and retrieval, while local features appeared almost ten years latter, and have initially aimed at Computer Vision applications such as aerial view reconstruction, 3D reconstruction, and panoramic image stitching, before they were "discovered" by the image retrieval community. The success of the local descriptor approach is explained due to the fact that classical global features have difficulty in distinguishing foreground from background objects, and thus are not very effective in recognition tasks for cluttered images [Tuytelaars and Mikolajczyk, 2008].

In addition to the classical color histogram [Swain and Ballard, 1991], many other examples can be found. The GIST descriptor [Oliva and Torralba, 2001] is worth mentioning because even recently it has received attention in the context of scene recognition [Torralba et al., 2003, 2008]. It is based on the idea of developing a low

dimensional representation of the scene, without need of segmentation. The image is divided into a 4×4 grid, for which orientation histograms are extracted. This arbitrary segmentation, however, does not allow the recognition of images that have suffered strong cropping, or images of objects from different viewpoints [Douze et al., 2009].

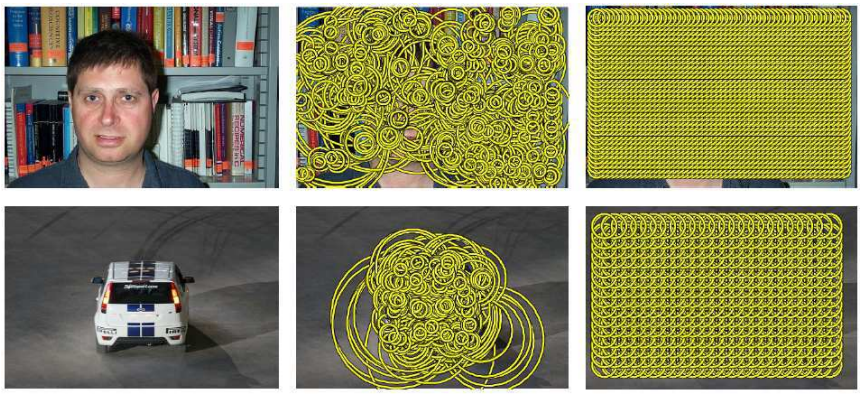
An approach to overcome the limitations of global descriptors is to segment the image into a limited number of regions or segments, with each such region corresponding to a single object or part thereof. The best known example of this approach is proposed by Carson et al. [2002], who segment the image based on color and texture. However, image segmentation is a very challenging task in itself, usually requiring a high-level understanding of the image content. For the general case, color and texture cues are insufficient to obtain meaningful segmentation.

Local features overcome those issues, allowing to find correspondences in spite of large changes in illumination conditions, viewpoint, occlusion, and background clutter. They also yield interesting descriptions of the visual image content for object or scene classification tasks (both for specific objects as well as for categories), without needing segmentation.

A *local feature* is an image pattern which differs from its immediate neighborhood [Tuytelaars and Mikolajczyk, 2008]. It is usually associated with a change of an image property or several properties simultaneously. Local features can be points, edgels or small image patches. Typically, two types of patch-based approaches can be distinguished [Tuytelaars, 2010]: (i) *interest points*, such as corners and blobs, whose position, scale and shape are computed by a feature detector algorithm (see Section 2.1.3) or (ii) *dense sampling*, where patches of fixed size are placed on a regular grid (possibly repeated over multiple scales). See Figure 2.1 for illustration.

Interest points focus on 'interesting' locations in the image and include various degrees of viewpoint and illumination invariance, resulting in better repeatability scores. However, when the contrast in an image is low, no interest point is detected, making the image representation useless. Dense sampling, on the other hand, gives a better coverage of the entire object or scene and a constant amount of features per image area. Regions with less contrast contribute equally to the overall image representation.

These two patch-based approaches are compared by Jurie and Triggs [2005] on object recognition and image categorization tasks. They concluded that dense representations outperform equivalent interest points based ones on those tasks. Dense sampling is also used in [Bosch et al., 2007; Vedaldi et al., 2009; Chatfield et al., 2011], boosting image classification and object detection results. However, due to computational constraints, a combination of interest points and dense sampling can be useful [Leibe and Schiele, 2003; Tuytelaars, 2010; Kim and Grauman, 2011].



original image interest points dense sampling

Figure 2.1: Illustration of interest points and dense sampling. Interest points focus on interesting locations in the image, while dense sampling gives a better coverage of the image, a constant amount of features per image area, and simple spatial relations between features. Figure from [Tuytelaars, 2010].

It should be mentioned that the current Bag-of-Words model has blurred somewhat the distinction between local and global descriptors, because they propose a single (global) feature vector based on several (local) features. The distinction can be particularly questioned when dense sampling is employed instead of the Computer Vision techniques of interest points or salient regions.

2.1.3 Local Feature Detection Operators

Feature detection is the identification of particular local features in the image (*e.g.*, blobs, corners, edges, interest points). The main property of local feature detection algorithms is the repeatability, *i.e.*, given two images of the same object or scene, taken under different viewing conditions, a high percentage of the features detected on the scene part visible in both images should be found in both images [Tuytelaars and Mikolajczyk, 2008]. Besides the repeatability property, good features detectors should have distinctiveness, locality, quantity, accuracy and efficiency. The importance of those different properties depends on the actual application and settings. In the following, we approach some local feature detection approaches. We especially concentrate on interest-point based detectors.

The *Hessian* [Beaudet, 1978] and the *Harris* detectors [Harris and Stephens, 1988] focus on a particular subset of points, namely those exhibiting signal changes in two directions. The former searches for image locations that exhibit strong derivatives in two orthogonal directions. The latter defines the interest points to be corner-like

structures. It was explicitly designed for geometric stability. In general, it can be stated that Harris locations are more specific to corners, while the Hessian detector also returns many responses on regions with strong texture variation. Also, Harris points are typically more precisely located as a result of using first derivatives rather than second derivatives and of taking into account a larger image neighborhood. Thus, Harris points are preferable when looking for exact corners or when precise localization is required, whereas Hessian points can provide additional locations of interest that result in a denser cover of the object.

As pointed out by Schmid et al. [2000], Harris and Hessian detectors are robust to image plane rotations, illumination changes, and noise. Nevertheless, the locations returned by both detectors are only repeatable up to relatively small scale changes. Hence, in order to be invariant to scale changes, the *Harris-Laplace* detector [Mikolajczyk and Schmid, 2002] proposed combining the Harris operator's specificity for corner-like structures with the Laplacian-based scale selection [Lindeberg, 1998]. As a drawback, however, the original Harris-Laplace detector typically returns a small number of points. An updated version of the Harris-Laplace detector has proposed based on a less strict criterion [Mikolajczy and Schmid, 2004], which yields more interest points at a slightly lower precision. As in the case of the Harris-Laplace, the same idea was applied to the Hessian detector, leading to the *Hessian-Laplace* detector [Mikolajczy and Schmid, 2004].

Additionally, both Harris-Laplace and Hessian-Laplace detectors were extended to yield affine-invariant region, resulting in the *Harris-Affine* and *Hessian-Affine* detectors [Mikolajczy and Schmid, 2004]. Detailed experimental comparisons can be found in [Mikolajczyk et al., 2005; Tuytelaars and Mikolajczyk, 2008; Li and Allinson, 2008].

2.1.4 Feature Description

Once a set of local features have been detected from an image, some measurements are taken from a region centered on local features and converted into *local descriptors*. Researchers have been developed a variety of local descriptors for describing the image content, such as: *SIFT* [Lowe, 2004], *SURF* [Bay et al., 2008], *HOG* [Dalal and Triggs, 2005], *GLOH* [Mikolajczyk and Schmid, 2005], *DAISY* [Tola et al., 2010]. In the following, we summarize the SIFT and SURF descriptors, the most commonly used local descriptors for visual recognition tasks.

2.1.4.1 Scale Invariant Feature Transformation (SIFT)

SIFT [Lowe, 1999, 2004] is the most widely used local approach for recognition tasks. It was originally proposed as combination of a difference-of-Gaussian (DoG) interest region detector and a histogram of gradient (HoG) locations and orientations feature descriptor. However, both components have also been used in isolation. In particular, a series of studies has confirmed that the SIFT descriptor is suitable for combination with all of the above-mentioned detectors and that it usually achieves good performance [Mikolajczyk and Schmid, 2005]. The SIFT descriptor has 128-dimensional feature vectors. It is invariant to scale, rotation, affine transformations, and partially invariant to illumination changes.

Initially, the SIFT descriptor was proposed to enable efficient point-to-point matching in object recognition tasks [Lowe, 1999]. In more recent works, this technique have been explored in the Bag-of-Words representation [Sivic and Zisserman, 2003], formally introduced in Section 2.2.

Several extensions of the original SIFT have been proposed in the literature. For example, *PCA-SIFT* [Ke and Sukthankar, 2004] applies PCA on normalized gradient patches to reduce the size of the original SIFT descriptor. *RIFT* [Lazebnik et al., 2005] divides each image patch into concentric rings of equal width to overcome the problem of dominant gradient orientation estimation required by SIFT. *GLOH* [Mikolajczyk and Schmid, 2005] extends SIFT by changing the location grid to a log-polar one and using PCA to reduce the size. *Rank-SIFT* [Li et al., 2011] sets each histogram bin to its rank in a sorted array of bins. Also, different ways of extending the SIFT descriptor from grey-level to color images have been proposed by different authors [Bosch et al., 2006; van de Weijer and Schmid, 2006; Burghouts and Geusebroek, 2009; van de Sande et al., 2010].

2.1.4.2 Speeded Up Robust Features (SURF)

SURF [Bay et al., 2006, 2008] is a scale and rotation-invariant interest point detector and descriptor. The detector is based on the Hessian matrix, but rather than using a different measure for selecting the location and the scale (as was done in the Hessian-Laplace detector), Bay et al. apply the determinant of the Hessian for both. The descriptor, on the other hand, describes a distribution of Haar-wavelet responses within the interest point neighborhood. The SURF descriptor is based on similar properties of localized information and gradient distribution as SIFT, with a complexity stripped down even further.

The main interest of the SURF descriptor lies in its fast computation of approximate differential operators in the scale-space, based on integral images and box-type convolution filters. Moreover, only 64 dimensions are used, reducing the time for feature computation and matching, and increasing simultaneously the robustness.

2.2 Mid-level Image Representations: BoW Models and Extensions

Mid-level feature extraction aims at transforming low-level descriptors into a global and richer image representation of intermediate complexity [Boureau et al., 2010a]. That image representation is commonly referred to as *mid-level* representation, since global features built upon low-level ones typically remain close to image-level information without attempts at high-level semantic analysis.

In order to get the mid-level representation, the standard processing pipeline follows three steps [Boureau et al., 2010a]: (i) low-level local feature extraction (previously addressed in the Section 2.1), (ii) *coding*, which performs a pointwise transformation of the descriptors into a representation better adapted to the task and (iii) *pooling*, which summarizes the coded features over larger neighborhoods. Classification algorithms are then trained on the mid-level vectors obtained (see Section 2.4).

In this section, we approach the family of mid-level representations which is most central to this dissertation. In particular, we focus on the Bag-of-Visual-Words representation (BoW) [Sivic and Zisserman, 2003], the most popular mid-level image representation. Here, instead of an exhaustive survey, we opt for a more formal development: our target is to lay out the mathematical cornerstones common to all BoW representations, exploring how those cornerstones have been established in early works, and how they are evolving in very recent works.

2.2.1 Early Techniques

Inspired by the Bag-of-Words Model from textual Information Retrieval [Baeza-Yates and Ribeiro-Neto, 1999], where a document is represented by a set of words, the Bag-of-Visual-Words Model (BoW) describes an image as a histogram of the occurrence rate of "visual words" in a "visual vocabulary" induced by quantizing the space of a local descriptor (*e.g.*, SIFT [Lowe, 2004]). The visual vocabulary of k visual words, also known as visual codebook or visual dictionary, is usually obtained by unsupervised learning over a sample of local descriptors from the training data.

As far as we know, the NeTra toolbox [Ma and Manjunath, 1999] was the first work to introduce that scheme, proposing dense grids of color points and unsupervised learning to build the codebook, using the LBG algorithm [Linde et al., 1980]. The RETIN system [Fournier et al., 2001] is based on a similar scheme, using local Gabor feature vectors and learning the codebook with Kohonen self-organized maps [Kohonen, 1988]. The technique was definitively popularized with the intuitive "Video Google" formalism [Sivic and Zisserman, 2003], which employs SIFT local descriptors and builds the codebook with k-means clustering algorithm [Duda et al., 2001]. Sivic and Zisserman applied the BoW scheme for object and scene retrieval, while Csurka et al. [2004] first exploited for the purpose of object categorization.

2.2.2 Current Formalism

Let us denote the "Bag-of-Features" (BoF), *i.e.*, the unordered set of local descriptors extracted from an image, by $\mathcal{X} = \{\mathbf{x}_j\}$, $j \in \{1, \ldots, N\}$, where $\mathbf{x}_j \in \mathbb{R}^D$ is a local descriptor vector and N is the number of local descriptors (either fixed grid points, either detected points of interest) in the image.

Let us suppose we have obtained (*e.g.*, by an unsupervised learning algorithm) a codebook, or visual dictionary $\mathcal{C} = \{\mathbf{c}_m\}$, $\mathbf{c}_m \in \mathbb{R}^D$, $m \in \{1, \ldots, M\}$, where M is the number of codewords, or visual words. Obtaining the codebook is essential for the BoW model, since the representation will be based on the codewords. Currently, the vast majority of methods obtains the codebook using unsupervised learning over a sample of local descriptors from the training images, usually employing k-means clustering algorithm [Duda et al., 2001]. In Figure 2.2, we illustrate the procedure to form the codebook. Other codebook learning algorithms are explored in Section 2.4.4.

The construction of the BoW representation can be decomposed into the sequential steps of coding and pooling [Boureau et al., 2010a]. The coding step encodes the local descriptors as a function of the codebook elements; while the pooling step aggregates the codes obtained into a single vector. The global aim is gaining invariance to nuisance factors (positioning of the objects, changes in the background, small changes in appearance, etc.), while preserving the discriminating power of the local descriptors.

The coding step can be modeled by a function f:

$$\begin{aligned} f : \mathbb{R}^D &\longrightarrow \mathbb{R}^M, \\ \mathbf{x}_j &\longrightarrow f(\mathbf{x}_j) = \alpha_{\mathbf{j}} = \{\alpha_{m,j}\}, \quad m \in \{1, \ldots, M\}. \end{aligned} \tag{2.1}$$

It can be understood as an activation function for the codebook, activating the

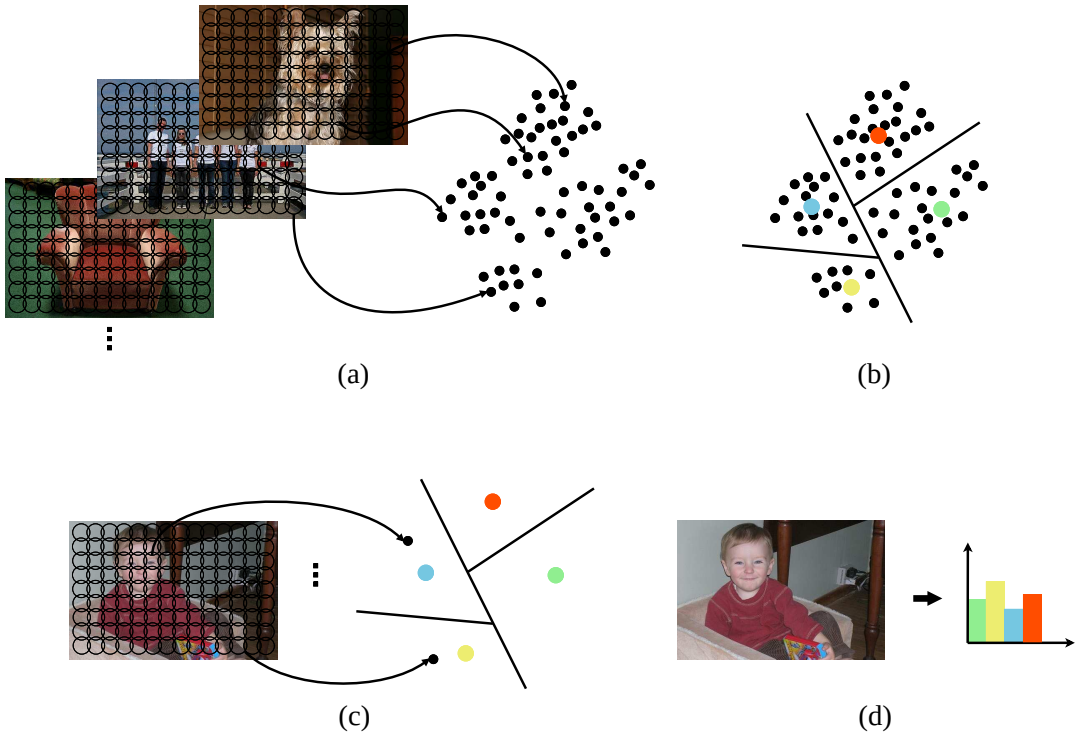


Figure 2.2: Illustration of visual codebook construction and codeword assignment. (a) A large sample of local features are extracted from a representative corpus of images. The black circles denote local feature regions in the images, and the small black circles denote points in some feature space, *e.g.*, SIFT descriptors. (b) Next, the sampled features are clustered in order to quantize the space into a discrete number of codewords. Those codewords are denoted with the large colored circles. (c) Now, given a new image, the nearest visual codeword is identified for each of its features. This maps the image from a set of high-dimensional descriptors to a list of codeword numbers. (d) A bag-of-words histogram can be used to summarize the entire image. It counts how many times each of the visual codewords occurs in the image.

codewords according to the local descriptor. In the classical BoW representation, the coding function activates only the codeword closest to the descriptor, assigning zero weight to all others:

$$\alpha_{m,j} = 1 \text{ iff } m = \underset{k \in \{1,\ldots,M\}}{\arg\min} \|\mathbf{x}_j - \mathbf{c}_k\|_2^2, \tag{2.2}$$

where $\alpha_{m,j}$ is the m^{th} component of the encoded vector α_j. That scheme corresponds to a *hard coding* or hard quantization over the codebook. The resulting binary code is very sparse, but suffers from instabilities when the descriptor being coded is on the boundary of proximity of several codewords [van Gemert et al., 2010].

The pooling step takes place after the coding step, and can be represented by a function g:

$$\begin{aligned} g : \mathbb{R}^N &\longrightarrow \mathbb{R}, \\ \alpha_{\mathbf{j}} = \{\alpha_{m,j}\}, j \in \{1, \ldots, N\} &\longrightarrow g(\{\alpha_j\}) = \mathbf{z}. \end{aligned} \tag{2.3}$$

Traditional BoW considers the *sum-pooling* operator:

$$g(\{\alpha_j\}) = \mathbf{z} : \ \forall m, \ z_m = \sum_{j=1}^{N} \alpha_{m,j}. \tag{2.4}$$

The vector $\mathbf{z}$, the final image representation, is given by sequentially coding, pooling and concatenating: $\mathbf{z} = [z_1, z_2, \cdots, z_M]^{\mathrm{T}}$. Regarding image classification, the goal is to find out which operators f and g provide the best classification performance using $\mathbf{z}$ as input. In Figure 2.3, we illustrate the BoW image classification pipeline showing coding and pooling steps.

2.2.3 BoW-based Approaches

The classical BoW representation has important limitations, and many alternatives to that standard scheme have been recently developed. Both steps of coding and pooling have been subject to important improvements, aiming at preserving more information while keeping the robustness to geometrical transformations that is inherent to BoW. A recent comparison of coding and pooling strategies is presented in [Koniusz et al., 2013]. Here, we start with the enhancements that concerns the coding step.

The simplest coding in the literature assigns a local descriptor to the closest visual codeword, giving one (and only one) nonzero coefficient. Quantization effects of that *hard coding* are found to be a source of ambiguity [Philbin et al., 2008]. In order to attenuate the effect of coding errors induced by the descriptor space quantization, one can rely on *soft coding* [van Gemert et al., 2008, 2010]. It is based on a soft-assignment to each codeword, weighted by distances/similarities between descriptors and codewords. The soft-assignment $\alpha_{m,j}$ to the codeword $\mathbf{c}_m$ is computed as follows:

$$\alpha_{m,j} = \frac{exp(-\beta \|\mathbf{x}_j - \mathbf{c}_m\|_2^2)}{\sum_{m'=1}^{K} exp(-\beta \|\mathbf{x}_j - \mathbf{c}_{m'}\|_2^2)}, \tag{2.5}$$

where β is a parameter that controls the softness of the soft-assignment (hard-assignment is the limit when $\beta \to \infty$).

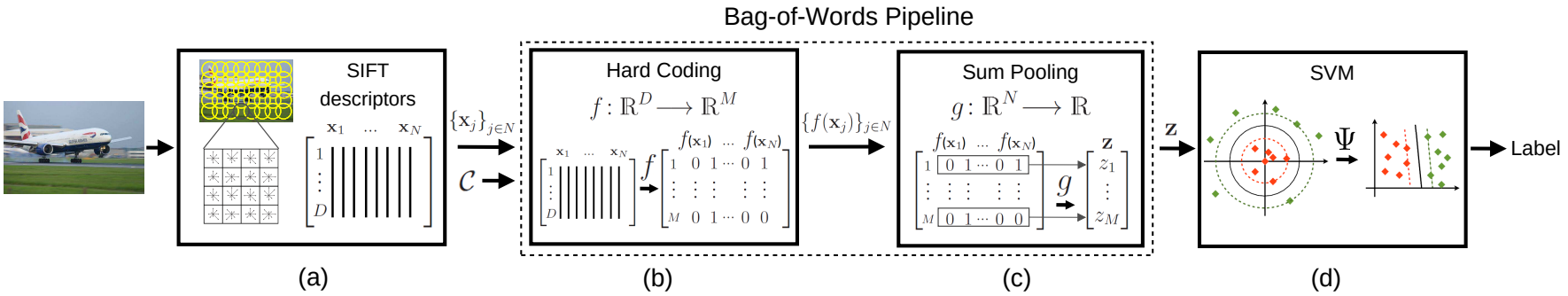


Figure 2.3: Overview of Bag-of-Words image classification pipeline showing coding and pooling steps. The construction of the BoW mid-level representation is highlighted by the dashed box. (a) $\{\mathbf{x}_j\}_{j \in N}$, where $\mathbf{x}_j \in \mathbb{R}^D$, SIFT local descriptors are extracted from an image. (b) At the coding step, the f coding function activates only the codeword closest to the descriptor, assigning zero weight to all others, which corresponds to a hard coding over the $\mathcal{C}$ visual codebook. M is the number of codewords. (c) Next, the g pooling function compacts all information pertaining to a codeword (the values along rows) into a single scalar (z). The vector $\mathbf{z}$, the BoW image representation, can be represented as $\mathbf{z} = [z_1, z_2, \cdots, z_M]^{\mathrm{T}}$. (d) Classification algorithms (such as SVM classifier) are then trained on the BoW mid-level vectors obtained.

However, soft-assignment results in dense code vectors, which is undesirable, among other reasons, because it leads to ambiguities due to the superposition of the components in the pooling step. Therefore, several intermediate strategies — known as *semi-soft coding* — have been proposed, often applying the soft assignment only to the k nearest neighbors (k-NN) of the input descriptor [Liu et al., 2011a].

Sparse coding (SC) [Yang et al., 2009b; Boureau et al., 2010a] modifies the optimization scheme by jointly considering reconstruction error and sparsity of the code, using the property that regularization with the ℓ_1-norm, for a sufficiently large regularization parameter λ, induces sparsity:

$$\alpha_j = \arg\min_{\alpha} \|\mathbf{x}_j - \mathcal{C}\alpha\|_2^2 + \lambda\|\alpha\|_1. \tag{2.6}$$

Yu et al. [2009] empirically observed that SC results tend to be local — nonzero coefficients are often assigned to bases nearby to the encoded data. They suggested a modification to SC, called Local Coordinate Coding (LCC), which explicitly encourages the coding to be local, and theoretically pointed out that under certain assumptions locality is more essential than sparsity, for successful nonlinear function learning using the obtained codes.

Locality-constrained Linear Coding (LLC) [Wang et al., 2010], which can be seem as a fast implementation of LCC, utilizes the locality constraint to project each descriptor into its local-coordinate system. It is similar to SC, but it adds a penalty for using elements of the codebook that have a large Euclidean distance from the descriptor being coded. Very close to LLC, the Sparse Spatial Coding (SSC) [Oliveira et al., 2012] codes a descriptor using sparse codebook elements nearby in descriptor space. SSC combines a sparse coding codebook learning and a spatial constraint coding stage (the spatial Euclidean similarity).

One of the strengths of those approaches is that one can learn the codebook with the same scheme, but optimizing over $\mathcal{C}$ and α. Efficient tools have been proposed to get tractable solutions [Mairal et al., 2010].

The pooling step has also been subject to extensions and enhancements. The simplest pooling operations, *sum-pooling* and *average-pooling*, have a "blurring" effect due to the averaging of the activations of all elements in the image. That is not always desirable, especially in the presence of very cluttered backgrounds. To overcome that limitation, alternative pooling schemes have been developed, *e.g.*, *max-pooling* [Yang et al., 2009b]. Instead of performing averaging operation, max-pooling computes

the maximum value of each dimension of α_j:

$$\mathbf{z}: \ \forall m, \ z_m = \max_{j \in \{1,\ldots,N\}} \alpha_{m,j}. \tag{2.7}$$

Max-pooling is often preferred[2] when paired with sparse coding and linear classifiers [Yang et al., 2009b; Boureau et al., 2010a].

Recently, Boureau et al. [2010b] conducted a theoretical analysis on feature pooling in image classification. They demonstrated that several factors, including pooling cardinality and sparsity of the features, affect the discriminative powers of different pooling operations. Furthermore, they showed that the best pooling type for a given classification task may be neither average nor max-pooling, but something in between.

Motivated by that consideration, Feng et al. [2011] proposed a *geometric ℓ_p-norm pooling* (GLP) method to perform feature pooling. It utilizes the class-specific geometric information on the feature spatial distributions, providing more discriminative pooling results. However, GLP only works well for datasets with well-positioned foreground objects (*e.g.*, Caltech-101 dataset[3]), restricting its applicability for many challenging datasets, especially those datasets containing large intra-class spatial variances (*e.g.*, PASCAL VOC datasets).

In [Liu et al., 2011a], the authors discussed the probabilistic essence of max-pooling and further developed a *mix-order max-pooling* strategy, which incorporates the occurrence frequency information ignored in simple max-pooling by estimating the probability of the "t-times" presence of a codeword in an image. The authors have shown in their experiments that mix-order max-pooling can lead to better classification performance (at least in classifying scenes and events) than the simple max-pooling.

Another extension to the BoW is to include spatial/layout information. The most popular technique to overcome the loss of spatial information is the Spatial Pyramid Matching (SPM) strategy [Lazebnik et al., 2006]. Inspired by the pyramid match of Grauman and Darrell [2005], Lazebnik et al. proposed to split an image into multiple levels of regular grids, which are described independently (*i.e.*, the pooling is operated over each block of the pyramid) and then concatenated into an image-level histogram.

In [Koniusz and Mikolajczyk, 2011], the authors proposed to include spatial and angular information directly at descriptor level. They used soft-BoW and sparse coding-based signatures, reporting promising results compared to SPM strategy. Jia et al. [2012] introduced spatial regions that do not follow the fixed spatial regions

[2]Depending on the sparse optimization scheme, the $\alpha_{m,j}$ values may be negative. If that occurs, the following pooling is usually applied: $\mathbf{z}: \ \forall m, \ z_m = \max_{j \in \{1,\ldots,N\}} \|\alpha_{m,j}\|$.

[3]Most images in the Caltech-101 dataset have roughly aligned and centered foreground objects.

of SPM and capture better dataset-specific spatial information. Sánchez et al. [2012] proposed to include information about the spatial layout of images in image signatures based on average statistics. Russakovsky et al. [2012] presented an object-centric spatial pooling approach which uses the location information of the objects to pool foreground and background features separately.

Despite recent techniques to include spatial information [Penatti et al., 2011], the simple SPM [Lazebnik et al., 2006] is still by far the most used approach to account for spatial information in BoW-based methods.

Incorporating higher-order statistics is another possible improvement to the classical BoW. By counting the number of occurrences of visual codewords, BoW encodes the zero-order statistics of the distribution of descriptors. The Fisher Vector (FV) [Perronnin et al., 2010c], as well as the related Vector of Locally Aggregated Descriptors (VLAD) [Jégou et al., 2010] and the Super-Vector Coding (SVC) [Zhou et al., 2010], also model the distribution of descriptors assigned to each codeword, encoding higher-order statistics.

Furthermore, Boureau et al. [2011] gives a new perspective to those recent powerful approaches, VLAD and SVC, as specific pooling operations. In those aggregated approaches, locality constraints are incorporated during the pooling step: only descriptors belonging to the same clusters are pooled together.

One of the best mid-level representations currently reported in the literature [Chatfield et al., 2011], the FV [Perronnin et al., 2010c] is based on the use of the Fisher kernel framework popularized by [Jaakkola and Haussler, 1998], with Gaussian Mixture Models (GMM) estimated over the whole set of images. That approach may be viewed as a generalization to the second order of the SVC [Zhou et al., 2010]. Indeed, the final image representation is also a vector concatenating vectors over each mixture term.

Several extensions of the FV have been proposed. Krapac et al. [2011] introduced Spatial Fisher Vector (SFV) to encode spatial layout of features. In SFV, spatial cells are adapted per codeword to the patch positions. While their representation is more compact, their evaluation shows minimal improvement over SPM in terms of classification accuracy. Picard and Gosselin [2011] generalized FV to higher-orders, the so-called Vector of Locally Aggregated Tensors (VLAT). However, its computational complexity, vector size and difficulty in estimating higher-order moments with confidence, limit the practicality of pushing the orders beyond the second. In [Negrel et al., 2012], the authors proposed to reduce the Picard and Gosselin's vector size.

In this dissertation, we propose BossaNova [Avila et al., 2012, 2013], a mid-level image representation for image classification, that enriches the BoW representation.

The fundamental novelty is an enhancement of the pooling operation by considering no more a scalar output for each row as in Equation 2.3, but a vector, summarizing the distribution of the $\alpha_{m,j}$. That strategy allows keeping more information, related to the confidence of the detection of each visual word $\mathbf{c}_m$ in the image.

In order to accomplish that goal, BossaNova departs from the parametric models commonly found in the literature (*e.g.*, [Perronnin et al., 2010c; Krapac et al., 2011]), by employing histograms. That density-based approach allows us to conciliate the need to preserve low-level local descriptor information and keeping the mid-level feature vector at a reasonable size. A preliminary version of the BossaNova representation, the so-called BOSSA [Avila et al., 2011], has allowed us to gain several insights into the benefits of the non-parametric choice and to explore the compromises between the opposite goals of discrimination versus generalization, representativeness versus compactness. BossaNova is introduced in Chapter 4, as well the complementarity of BossaNova and Fisher Vector. The latter one is summarized in the next section.

Fisher Vector Representation

We only provide a brief introduction to the Fisher Vector (FV). More details can be found in [Perronnin and Dance, 2007; Perronnin et al., 2010c].

As mentioned before, BoW encodes the zero-order statistics of the distribution of descriptors by counting the number of occurrences of codewords. The FV extends the BoW by encoding the average first- and second-order differences between the descriptors and codewords. Mathematically speaking, the FV $\mathcal{G}_\lambda^{\mathcal{X}}$ characterizes a sample $\mathcal{X}$ by its deviation from a distribution u_λ (with parameters λ):

$$\mathcal{G}_\lambda^{\mathcal{X}} = L_\lambda G_\lambda^{\mathcal{X}}, \tag{2.8}$$

$G_\lambda^{\mathcal{X}}$ is the gradient of the log-likelihood with respect to λ:

$$G_\lambda^{\mathcal{X}} = \frac{1}{N}\nabla_\lambda \log u_\lambda(\mathcal{X}), \tag{2.9}$$

L_λ is the Cholesky decomposition of the inverse of the Fisher information matrix F_λ, *i.e.* $F_\lambda^{-1} = L'_\lambda L_\lambda$. In [Perronnin and Dance, 2007], $\mathcal{X} = \{\mathbf{x}_j \in \mathbb{R}^D\}$, $j \in \{1, \ldots, N\}$, where $\mathbf{x}_j$ is a local descriptor vector and N is the number of local descriptors in the image and $u_\lambda = \sum_{i=1}^{M} w_i u_i$ is a GMM. We denote $\lambda = \{w_i, \mu_i, \sigma_i\}$, where $i \in \{1, \ldots, M\}$ and w_i, μ_i, σ_i are respectively the mixture weight, mean vector and diagonal covariance matrix of Gaussian u_i.

Let $\gamma_j(i)$ be the soft assignment of descriptor $\mathbf{x}_j$ to Gaussian i. Let $\mathcal{G}^{\mathcal{X}}_{\mu,i}$ and $\mathcal{G}^{\mathcal{X}}_{\sigma,i}$ be the gradient with respect to μ_i and σ_i, respectively. As reported by Perronnin et al. [2010c], the gradient with respect to the weight parameter brings little additional information, hence it is discarded. Then, mathematical derivations lead to:

$$\mathcal{G}^{\mathcal{X}}_{\mu,i} = \frac{1}{N\sqrt{w_i}} \sum_{j=1}^{N} \gamma_j(i) \left(\frac{\mathbf{x}_j - \mu_i}{\sigma_i} \right), \tag{2.10}$$

$$\mathcal{G}^{\mathcal{X}}_{\sigma,i} = \frac{1}{N\sqrt{2w_i}} \sum_{j=1}^{N} \gamma_j(i) \left[\frac{(\mathbf{x}_j - \mu_i)^2}{{\sigma_i}^2} - 1 \right]. \tag{2.11}$$

The FV $\mathcal{G}^{\mathcal{X}}_{\lambda}$ is the concatenation of $\mathcal{G}^{\mathcal{X}}_{\mu,i}$ and $\mathcal{G}^{\mathcal{X}}_{\sigma,i}$ vectors for $i \in \{1, \ldots, M\}$ and is therefore $2MD$-dimensional. An advantage of FV with respect to the BoW is that discriminative signatures can be obtained with small codebooks. On the other hand, the high-dimensional signatures, as FV signatures, come at a high storage/memory cost which poses a challenge to learning, especially on large-scale datasets.

2.3 Feature Normalization

2.3.1 Dimensionality Reduction

Feature descriptors are usually high-dimensional (*e.g.*, SIFT is represented as a 128-dimensional vector). That leads to a high memory usage and an elevated computational time. Moreover, high-dimensional problems are often susceptible to the well-known problem of the *curse of dimensionality* [Bellman, 1961]. To deal with this issue, dimension reduction techniques are often applied as a data preprocessing step. This typically involves the identification of a suitable low-dimensional representation for the original high-dimensional data. By working with this reduced representation, tasks such as classification or clustering can often yield more accurate results, while computational costs may also be significantly reduced [Cunningham, 2008].

Principal Component Analysis (PCA)

One of the most popular techniques for dimensionality reduction is the Principal Components Analysis (PCA). PCA, is a linear technique widely used for dimensionality reduction [Jolliffe, 2002]. It aims to reduce the dimensionality of multivariate data while preserving as much of the relevant information as possible. This method is a

form of unsupervised learning in that it does not take class labels into account. The PCA is also known as the *Karhunen-Loève* transform.

There are two commonly used definitions of PCA that give rise to the same algorithm. PCA can be defined as the orthogonal projection of the data onto a lower dimensional linear space, known as the *principal subspace*, such that the variance of the projected data is maximized [Hotelling, 1933]. Equivalently, it can be defined as the linear projection that minimizes the average projection cost, defined as the mean squared distance between the data points and their projections [Pearson, 1901].

The PCA dimension reduction method has been successfully applied in a large number of domains, such as object recognition [Fei-Fei et al., 2007; Perronnin et al., 2010c; Zhou et al., 2010; Krapac et al., 2011; Sánchez et al., 2012; Avila et al., 2013] and image retrieval [Jégou et al., 2010; Perronnin et al., 2010a; Jégou and Chum, 2012]. The main drawback of PCA is that the size of the covariance matrix is proportional to the dimensionality of the data points. As a result, the computation of the eigenvectors might be infeasible for very high-dimensional; for a matrix of size $D \times D$, the computational cost of computing the full eigenvector decomposition is $O(D^3)$. However, to project the data points onto the first N principal components, we only need to find the first N eigenvalues and eigenvectors. This can be done with more efficient techniques, for example the power method [Golub and Van Loan, 1996], whose the computational complexity is $O(ND^2)$, or the expectation-maximization algorithm.

As mentioned before, the PCA is designed to model linear variabilities in high-dimensional data. However, many high-dimensional datasets have a nonlinear nature. In those cases, the high-dimensional data lie on or near a nonlinear manifold and therefore PCA cannot model the variability of the data correctly. One of the algorithms designed to address the problem of nonlinear dimensionality reduction is the Kernel PCA [Schölkopf et al., 1998; Shawe-Taylor and Cristianini, 2004]. In KPCA, through the use of kernels, principle components can be computed efficiently in high-dimensional feature spaces that are related to the input space by some nonlinear mapping.

2.3.2 BoW Normalization Techniques

One possibility of increasing the systems' performance is to carefully examine the feature normalization techniques. In particular, large margin classifiers are known to be sensitive to the way features are scaled (see, for example [Chang and Lin, 2011], in context of SVM). Therefore, despite the fact it has been neglected so far in most research papers in the literature, the mid-level feature normalization is a crucial step.

In the Bag-of-Words (BoW) model proposed by Sivic and Zisserman [2003], the

vector components are weighted by a *tf-idf* transformation, where *tf* means 'term-frequency' and *idf* means 'inverse document-frequency'. The idea is that word frequency weights words occurring often in a particular document, and thus describe it well, whilst the inverse document frequency downweights words that appear often in the dataset.

A technique usually regarded as part of term weighting is to normalize of the term count by the number of terms in the document (*i.e.*, document length) into a unit-length term frequency vector. This ℓ_1 normalization eliminates the difference between long and short documents with similar word distribution. For images, this means normalizing the count of visual words by the total number of local descriptors in each image, which varies greatly according to the complexity of the image scene.

Recently, the ℓ_1 and ℓ_2 normalizations have been widely used to normalize the BoW-based feature vectors, such as [Nister and Stewenius, 2006; Jégou et al., 2010; Perronnin et al., 2010c; Avila et al., 2011; Chatfield et al., 2011; Picard and Gosselin, 2011; Negrel et al., 2012]. The normalization policy can be driven by the kernel choice, and different kernels lead to different normalization strategies. For example, ℓ_2 normalization is appropriate when using linear kernels, whereas ℓ_1 is optimal when using χ^2 or intersection kernels, see [Vedaldi and Zisserman, 2012]. However, there is no general agreement on the benefit of performing these normalizations, because they can discard relevant information.

Contradicting experimental results have been reported in the literature, and the optimal normalization policy remains largely data-dependent. For example, some authors reported that ℓ_2 normalization negatively impacts performances, and therefore they chose not performing any normalization method (*e.g.*, [Liu et al., 2011a; Boureau et al., 2011]). In [Yang et al., 2007], this normalization factor is evaluated in two benchmarks, PASCAL VOC and TRECVID. The authors have contradicting observations between the two datasets regarding the normalization factor. In PASCAL VOC, normalized features consistently outperforms un-normalized ones. However, in TRECVID the un-normalized features are always better than their normalized counterparts. According to Yang et al. [2007], a plausible explanation is that, PASCAL VOC has images of various sizes, and its classification performance benefits from the normalization factor which eliminates the difference on image sizes. This is not the case with TRECVID, which contains video frames of identical size, and normalization decreases the performance by suppressing the information on the number of visual words in each video frame.

In recent aggregate methods, *e.g.*, Fisher Vector (FV) [Perronnin et al., 2010c], VLAD [Jégou et al., 2010] or Super-Vector Coding (SVC) [Zhou et al., 2010], the cod-

ing step outputs a vectorial representation where the mean cluster value is subtracted from each local descriptor. In [Jégou and Chum, 2012], the favorable impact of this centering on retrieval/classification performances is analyzed. The main claim of the paper is that centering data with linear kernels[4], negative evidence is better taken into account. Negative evidence refers to the joint absence of a given visual word in two image representations. If no centering is performed, negative evidence is encoded similarly (*i.e.*, 0) than when a given word is absent in one image but present in the other, which is not desirable. Another feature highlighted in [Jégou and Chum, 2012] is the ability of whitening to limit the impact of co-occurrences.

Another example of normalization technique is the power-law normalization [Perronnin et al., 2010c]. It is performed by applying the operator $f(z) = sign(z)|z|^{\alpha}$ independently on each component, where $0 \leq \alpha \leq 1$. Perronnin et al. [2010c] empirically observed that the power normalization consistently improves the classification performance. In [Jégou et al., 2012], several complementary interpretations that justify this transform are listed. However, in [Perronnin et al., 2010c; Jégou et al., 2012], the authors have applied the power normalization with $\alpha = 0.5$ for only one representation, the Fisher Vector. Safadi [2012] studied the impact of the α parameter on different representations, including color histograms and BoW approaches. He showed that the optimal value of α varies for each of the representations.

2.4 Machine Learning Algorithms for Image Classification

Machine Learning is concerned with the design and development of algorithms that allow computers to learn based on data. The most fundamental distinction in machine learning is that between supervised and unsupervised learning algorithms.

In *supervised learning* problems, a machine learning algorithm induces a prediction function using a set of examples, called a *training set*. Each example consists of a pair formed by an observation annotated with a corresponding label. The goal of the learned function is to predict the correct label associated with any new observation. When the labels are discrete, the task is referred to as a *classification problem*. Otherwise, for real-valued labels, the task is referred to as a *regression problem*.

The main goal of a machine learning algorithm is to perform correct predictions for previously unknown observations. Therefore, machine learning is not simply a

[4]for translation-invariant kernels, such as radial-basis function kernels, the centering has of course no impact.

question of remembering, but mainly of *generalizing* a model to unknown cases. In practice, a *testing set*, *i.e.* a set of examples never seen by the learning algorithm during the training phase, along with a performance measure are thus employed to evaluate the generalization ability of the learned model.

In *unsupervised learning* problems, one can consider unlabeled training examples and try to uncover regularities in the data. One can also make use of both labeled and unlabeled data for training (typically a small amount of labeled data with a large amount of unlabeled data). This is referred to as *semi-supervised learning* problem.

In this section, we only consider some of the most successful supervised learning algorithms for image classification problems. We start by presenting the Support Vector Machines (Section 2.4.1), a very popular and powerful learning technique for data classification. Next, we discuss ensemble techniques (Section 2.4.2), a strategy which weighs several individual classifiers, and combines them in order to obtain a classifier that outperforms every single one of them. Finally, we approach the k-Nearest Neighbor classifier (Section 2.4.3).

2.4.1 Support Vector Machine

Support Vector Machines (SVMs) [Vapnik, 1998] are supervised learning methods originally used for linear binary classification. They are the successful application of the kernel idea [Aizerman et al., 1964] to large margin classifiers [Vapnik and Lerner, 1963] and have been proved to be powerful tools. In this section, we briefly introduce SVMs. A deep and comprehensive introduction to SVMs can be obtained in [Cristianini and Shawe-Taylor, 2000; Scholkopf and Smola, 2001].

The basic ideas behind the SVM algorithm can be explained by three incremental steps. First, Vapnik and Lerner [1963] proposed to construct the *optimal hyperplane* which maximizes the margin, *i.e.* the minimal distance between the hyperplane separating the training examples into its two classes. Then, Guyon et al. [1993] proposed to construct the optimal hyperplane in the feature space induced by a *kernel function* (a kernel function in the original space is equivalent to a standard scalar product in this feature space). Finally, Cortes and Vapnik [1995] showed that noisy problems are best addressed by allowing some examples to violate the margin constraint.

The SVM optimization problem is equivalent to a quadratic program (QP), that optimizes a quadratic cost function subject to linear constraints. However, this optimization procedure can only be applied to small sized data sets due to its high computational and memory costs. Thus, efficient batch numerical algorithms have been developed to solve the SVM QP problem. One of the best known methods is

the Sequential Minimal Optimization (SMO) [Platt, 1999], which iteratively solves the smallest possible optimization problem each time with two examples. The advantage of SMO is that a QP problem with two example can be solved analytically, and thus a numerical QP solver is avoided. The state-of-the-art implementation of SMO algorithm is the software LIBSVM [Chang and Lin, 2011].

The use of a linear kernel (or a explicit mapping) heavily simplifies the SVM optimization problem. Computing gradients of either the primal or dual cost function is cheap making linear optimization very interesting when one needs to handle large-scale databases. However, this simpler complexity can also result in a loss of generalization power compared to nonlinear kernels [Bordes, 2010].

Recent work exhibits new algorithms scaling linearly in time with the number of training examples. For example, SVM^{perf} [Joachims, 2006] is a simple cutting-plane algorithm for training linear SVM converging in linear time for classification. LIBLINEAR [Hsieh et al., 2008] also reaches very good performances on large scale datasets, converging in linear time with an efficient dual coordinate descent procedure.

Recently, it has experimentally shown that for linear SVMs, stochastic gradient descent (SGD) [Bottou and Bousquet, 2008] approaches in the primal significantly outperform complex optimization methods (for instance, PEGASOS [Shalev-Shwartz et al., 2007], SVMSGD [Bottou, 2007], SGDQN [Bordes et al., 2009]). However, many real-world problems do not generalize well in the original feature space with linear frontiers (hyperplanes). One way to tackle that problem is to approximate nonlinear kernel by linear one [Williams and Seeger, 2001; Vedaldi and Zisserman, 2012]. In most cases, the approximated representations reach about the same level of performances than the exact kernels.

SVM classifiers have been so successful in visual recognition problems, that it is easy to pick dozens of papers that apply them in literature. A few selected ones that apply linear kernels are: [Yang et al., 2009b; Perronnin et al., 2010c; Zhou et al., 2010; Krapac et al., 2011; Sánchez et al., 2012]. A few that explore nonlinear SVM classifiers are [Lazebnik et al., 2006; van Gemert et al., 2010; Guillaumin et al., 2010; Picard and Gosselin, 2011; Avila et al., 2013].

2.4.2 Ensemble Techniques

The main idea behind the ensemble methodology is to weigh several individual classifiers, and combine them in order to obtain a classifier that outperforms every single individual [Rokach, 2010]. That research area is know under different names in the literature: committees of learners, mixtures of experts, classifier ensembles, multiple

classifier systems, consensus theory, etc. [Kuncheva and Whitaker, 2003].

Diversity is a crucial condition for obtaining accurate ensembles [Kuncheva and Whitaker, 2003; Brown et al., 2005]. One way to achieve diversity is to use different training datasets to train individual classifiers. Such datasets are often obtained through re-sampling techniques, such as bootstrapping or bagging, where training data subsets are drawn randomly, usually with replacement, from the entire training data. Another way is to use different training parameters for different classifiers. Adjusting such parameters allows one to control the instability[5] of the individual classifiers, and hence contribute to their diversity. Furthermore, the most popular method to achieve diversity is to train different classifiers on different feature subsets. That is widely used in image classification tasks.

Numerous algorithms have been proposed to construct a good classifier ensemble, improving both the accuracy of the base classifiers and the diversity among them. In the following, we present some ensemble approaches.

Bagging, introduced by Breiman [1996], is one of the most intuitive and perhaps the simplest ensemble based algorithms. Diversity of classifiers in bagging — a name derived from "bootstrap aggregation" — is obtained by using bootstrapped replicas of the training data. That is, different training data subsets are randomly drawn (with replacement) from the entire training dataset. Each training data subset is used to train a different classifier of the same type. Individual classifiers are then combined by taking a simple majority vote of their decisions. For any given instance, the class chosen by the most number of classifiers is the ensemble decision. Since the training datasets may overlap substantially, additional measures can be used to increase diversity, such as using a subset of the training data for training each classifier, or using relatively weak classifiers[6]. In general, bagging improves recognition for unstable classifiers since it effectively averages over their discontinuities [Alpaydin, 2010]. One example of unstable classifier that is rendered useful by bagging are decision trees: they are unstable when trained by greedy algorithms (a slight change in the position of a single training point can lead to a radically different tree), but often present very good performance when used in ensembles.

Examples of bagging in image recognition include a bagging ensemble of Linear Discriminant Analysis used for scene recognition [Lu et al., 2005], and another

[5] A classifier is an unstable algorithm if small changes in the training set causes a large difference in the generated learner.

[6] It suffices that their accuracy on the training set be slightly better than random guessing.

one applied for recognizing different kinds of vegetables and fruits [Rocha et al., 2008]. Zhang and Dietterich [2008] employed bagged Decision Lists for object recognition task.

Boosting also creates an ensemble of classifiers by re-sampling the data, which are then combined by majority voting. However, in contrast to bagging, the re-sampling is applied to provide the most informative training data for each consecutive classifier. The original boosting algorithm [Schapire, 1990] combines three weak classifiers to generate a strong weak classifier. A weak classifier has an accuracy probability slightly over random guess, while a strong classifier has an accuracy probability that can be made arbitrarily close to 100%. The somewhat counterintuitive principle of boosting, is that they require individual weak classifiers in order to guarantee that the entire ensemble will converge to a strong classifier. There are a number of variations on basic boosting. The most widely used form of boosting algorithm is the AdaBoost [Freund and Schapire, 1995], short for "adaptive boosting". It improves the simple boosting algorithm via an iterative process, allowing to choose automatically weak assumptions with adjusted weight. Schapire et al. [1998] explain that the success of AdaBoost is due to its property of increasing the margin. If the margin increases, the training instances are better separated and an error is less likely. That makes AdaBoost's aim similar to the SVM classifier [Alpaydin, 2010].

Over the years, boosting approaches have been proposed to image classification tasks. For instance, the classical AdaBoost algorithm is applied by Opelt et al. [2004] for object recognition. Wolf and Martin [2005] proposed a modified version of the gentleBoost algorithm [Friedman et al., 2000] which enables it to work with only a few examples. They tested their algorithm on Caltech datasets, which have few training examples. Saffari et al. [2008] also proposed a generalization of the gentleBoost algorithm, but to the semi-supervised domain. Gehler and Nowozin [2009] presented a boosting-oriented scheme optimizing alternately the combination weights and the combined kernels. Inspired by the boosting framework, Lechervy et al. [2012] introduced a novel algorithm for image categorization, which designs multi-class kernel functions based on an iterative combination of weak kernels.

Random Forests combine Breiman's bagging idea and the random selection of features, which is an example of the random subspace method introduced by Ho [1995]. A random forest [Breiman, 2001] can be created from individual decision

trees, varying randomly certain training parameters. Such parameters can be bootstrapped replicas of the training data, as in bagging, but they can also be different feature subsets as in random subspace methods.

Their popularity is largely due to the tracking application of Lepetit and Fua [2006]. Random forests have been applied to object recognition problems in [Moosmann et al., 2006; Winn and Criminisi, 2006] but only for a relatively small number of classes. Bosch et al. [2007] increased the number of object categories by an order of magnitude. In [Shotton et al., 2008], randomized decision forests are used for both clustering and classification. Leistner et al. [2009] extended the usage of random forests to semi-supervised learning problems. Yao et al. [2011] proposed a random forest with discriminative decision trees algorithm for fine-grained categorization tasks.

Stacked Generalization (or stacking) [Wolpert, 1992] is a different way of combining multiple models, that introduces the concept of a meta-classifier. In Wolpert's stacking, an ensemble of classifiers is first trained using bootstrapped samples of the training data, and the outputs are then used to train a second-level classifier (meta-classifier). The underlying idea is to learn whether training data have been properly learned. For example, if a particular classifier incorrectly learned a certain region of the feature space, and hence consistently misclassifies instances coming from that region, then the second-level classifier may be able to learn this behavior, and along with the learned behaviors of other classifiers, it can correct such improper training.

Stacking has been exploited in image classification tasks. Tsai [2005] presented a two-level stacked generalization scheme composed of three generalizers (color, texture, and high-level concept) of SVMs for image classification. In [Abdullah et al., 2009], the effectiveness of two different two-level stacking SVMs are compared to the naïve approach, that combines all descriptors in a single input vector for a SVM. They showed that the two-level stacking SVMs outperforms the naïve approach for image classification. In [Znaidia et al., 2012], base classifiers are trained on the considered modalities (visual, contextual and hierarchical) and combined by the stack generalization approach proposed by Wolpert [1992].

The simplest strategy for building an ensemble is bagging, whose diverse component classifiers are built by subsampling the training cases or subsampling the features. Boosting will often produce even better improvements on error than bagging as it has the potential to reduce the bias component of error in addition to the variance com-

ponent. Random forests are an interesting strategy for building ensembles that can provide some useful insights into the data in addition to providing a very effective classifier. Finally, stacked generalization can be seen as a more sophisticated version of cross-validation, exploiting a strategy more useful than cross-validation's crude winner-takes-all for combining the individual generalizers.

2.4.3 k-Nearest Neighbor

Perhaps the most straightforward classifier among machine learning techniques is the Nearest Neighbor Classifier [Duda et al., 2001], where examples are classified based on the class of their nearest neighbors in the descriptor space. It is often useful to take more than one neighbor into account so the technique is more commonly referred to as k-Nearest Neighbor (k-NN) Classification, where k nearest neighbors are used in determining the class [Cunningham et al., 2008].

k-NN classifiers provide good image classification when the query image is similar to one of the labeled images in its class [Boiman et al., 2008]. Indeed, k-NN classifiers have proved to be competitive in restricted image classification domains (*e.g.*, OCR and texture classification [Zhang et al., 2006]), where the number of labeled dataset images is very high relative to the class complexity.

Although k-NN classifiers are extremely simple, easy to implement, and require no learning/training phase, the large performance gap between those classifiers and SVM-based methods (see Section 2.4.1) often renders k-NN classifiers useless. In [Boiman et al., 2008], the authors argued that two practices commonly used in image classification methods (as BoW-based approaches) have led to the inferior performance of k-NN image classifiers: (i) the quantization of local image descriptors (used to generate BoW features, visual codebooks) and (ii) the computation of 'image-to-image' distance (essential to kernel methods, *e.g.*, SVM), instead of 'image-to-class' distance.

Hence, Boiman et al. [2008] proposed the Naive Bayes Nearest Neighbor (NBNN) classifier. Its good performance is mainly due to the avoidance of a vector quantization step, and the use of image-to-class comparisons, yielding good generalization. Nonetheless, NBNN also has its limitations. The computational cost during testing is high, especially when sampling very densely which often seems necessary to obtain good results. Also, the method assumes similar densities in feature space for all classes, which is often violated, resulting in a strong bias towards one or a few object classes.

Behmo et al. [2010] corrected NBNN for the case of unbalanced training sets. They also pointed out that a major practical limitation of NBNN is the time that is needed to perform the NN search. To overcome that limitation, Lowe [2012] proposed

the local NBNN, which merges all of the reference data together into one search structure (instead of maintaining a separate search structure for each class), allowing quick identification of a descriptor's local neighborhood. Recently, Tuytelaars et al. [2011] proposed a kernelized version of the NBNN classifier. Their scheme keeps the image-to-class comparisons, while at the same time fitting it in the kernel-based line of work popular for image classification. A shortcoming of the NBNN kernel is that it does not scale well in the number of classes.

2.4.4 Visual Codebook Learning

Learning a visual codebook is an effective means of extracting the relevant visual content of an image dataset, which is used by most of the classification systems. As mentioned in Section 2.2.2, the codebook is essential for the BoW model, since the representation will be based on the visual codewords. In this section, we explore the most influential visual codebook learning approaches to image classification problems.

The standard pipeline to form the visual codebook consists of (i) collecting a large sample of local descriptors from a representative corpus of images, and (ii) quantizing the descriptor space according to their statistics. Therefore, the choice of quantization algorithm used to construct it is an important concern in learning the visual codebook. Usually, codebooks are constructed by using unsupervised learning algorithms over a sample of local descriptors from the training images.

In that scenario, the k-means clustering algorithm [Duda et al., 2001] is a standard approach applied in many works in the literature [Sivic and Zisserman, 2003; Csurka et al., 2004; Willamowski et al., 2004; Fei-Fei and Perona, 2005; Bosch et al., 2006; Lazebnik et al., 2006; Quelhas et al., 2007; Jégou et al., 2010; Picard and Gosselin, 2011; Avila et al., 2011, 2012, 2013]. Other unsupervised learning algorithms also have been explored, such as agglomerative clustering [Leibe and Schiele, 2003], co-clustering [Liu and Shah, 2007], hierarchical clustering [Nister and Stewenius, 2006; Fulkerson et al., 2008], mean-shift based clustering [Jurie and Triggs, 2005], and Gaussian mixture model [Perronnin and Dance, 2007; Parikh et al., 2009; Perronnin et al., 2010c; Krapac et al., 2011; Sánchez et al., 2012].

An alternative approach to obtain the visual codebook is randomly selecting local descriptors as visual codewords [Nowak et al., 2006; Viitaniemi and Laaksonen, 2008; Penatti et al., 2011]. In [Nowak et al., 2006], the online k-means codebooks are compared with the random ones. Although the former is better than the latter, the randomly selected codebooks produce very respectable results. Those results are also observed by Viitaniemi and Laaksonen [2008].

The visual codebook learning methods mentioned until now, like all unsupervised learning approaches, do not take into account the category labels. Hence, many supervised approaches have been proposed to construct discriminative visual codebooks that explicitly incorporate category-specific information [Farquhar et al., 2005; Winn et al., 2005; Perronnin et al., 2006; Mairal et al., 2008; Moosmann et al., 2008; Larlus and Jurie, 2009; Lazebnik and Raginsky, 2009; Boureau et al., 2010a; Jiang et al., 2011]. For example, Perronnin et al. [2006] characterize images using a set of category-specific histograms, where each histogram describes whether the content can best be modeled by the universal codebook or by its corresponding category codebook. Mairal et al. [2008] propose an algorithm to learn discriminative codebooks for sparse coding, which requires each encoded vector to be labeled. Lazebnik and Raginsky [2009] incorporate discriminative information by minimizing the loss of mutual information between features and labels during the quantization.

Moreover, the visual codebook may be constructed by manually labeling image patches with a semantic label [van Gemert et al., 2006; Vogel and Schiele, 2007; Liu et al., 2009]. For example, Vogel and Schiele [2007] construct a semantic codebook by manually associating the local patches to certain semantic concepts such as "water", "sky", "grass". The idea behind a semantic codebook is that the meaning of an image may be expressed in the meaning of its constituent codewords. The obvious drawback is the large amount of manual labor required, which makes that approach infeasible.

Recently, more sophisticated techniques have been adapted to learn the visual codebook, such as restricted Boltzmann machines (RBM). In [Goh et al., 2012], the visual codebook is trained in two learning phases — unsupervised and supervised. During the unsupervised learning phase, the authors employ a RBM regularization method that enforces selective codewords. For the supervised phase, the codewords are adapted to be discriminative with respect to a local classifier that is concurrently learned. Although the codebooks are compact and inference is fast, the supervised optimization for associating local descriptors to labels deviates from the actual problem of global image classification.

2.5 Other Approaches for Image Classification

To put the Bag-of-Words model (see Section 2.2) in context, we briefly summarize some alternative approaches for image classification.

2.5.1 Biologically-inspired Models

Biologically-inspired computational models for image classification attempt to simulate the process of visual cortex in human vision task [Fukushima and Miyake, 1982; Riesenhuber and Poggio, 1999; Serre et al., 2007; Thériault et al., 2012].

Research on biological visual systems has been an important field of study since the awarded work of Hubel and Wiesel [1959, 1968]. Their studies suggested that the processing in the visual cortex follows a hierarchical structure. Thereafter, various hierarchical image classification approaches have been developed. For example, Fukushima and Miyake [1982] proposed *Neocognitron*, a hierarchical multi-layered network that is capable of merging simple visual features into a more complex whole while retaining some degree of invariance to basic visual transforms.

One biologically-inspired model which has been received attention in recent years comes from the *HMAX model* of Riesenhuber and Poggio [1999], which focuses less on learning and more on designing simple operations inspired by the visual cortex. This model alternates layers of features extraction with layers of maximum pooling.

Several extensions of the HMAX model have been suggested. For instance, Serre et al. [2007] extended the HMAX to add multi-scale representations as well as more complex visual features. Mutch and Lowe [2008] improved the HMAX of Serre et al. [2007] by tuning the complex visual features to the dominant local orientations. Thériault et al. [2011, 2012] proposed to build complex features in terms of the local scales of image structures.

Despite the success of the HMAX model, there are two important limitations of such a model [Han and Vasconcelos, 2010]. First, because the organization of the network lacks a clear computational justification, the HMAX model lacks a principled optimality criterion and training algorithm. That limits its relevance as an explanation for the underlying biological computations. Second, and quite importantly, the HMAX model does not account for the psychophysical and physiological evidence on the important role played by visual attention in processes such as object recognition.

Another biologically-inspired model is the *Convolutional Neural Network* (CNN), introduced by LeCun et al. [1990]. In the original CNN, parameters of the whole network are trained in a supervised manner using the error backpropagation algorithm. For image classification tasks, several variants of CNN have emerged either supervised feature learning [Nebauer, 1998; LeCun et al., 2004, 2010; Sermanet et al., 2012; Krizhevsky et al., 2012] or unsupervised feature learning [Huang and LeCun, 2006; Ranzato et al., 2007b; Kavukcuoglu et al., 2010; Taylor et al., 2010]. Most forms of CNN models, besides being biologically inspired, should also be considered "deep"

models, *i.e.*, models characterized by the presence of several layers of learning nodes ("neurons"), in contrast to the "shallow" models that have at most three layers (input, a single hidden layer, and output). We will explore those "deep" models in more detail in the next section.

2.5.2 Deep Models

Deep models aim at learning feature hierarchies with features from higher levels of the hierarchy formed by the composition of lower level features [Bengio, 2009]. Automatically learning features at multiple levels of abstraction allows a system to learn complex functions mapping the input to the output directly from data, without depending so much on human-crafted features.

Except for the CNN models, mentioned in the previous section, before 2006, deeper architectures were considered "untrainable" for practical purposes. Researchers reported positive experimental results with typically two or three levels (*i.e.*, one or two hidden layers), but training deeper networks consistently yielded poorer results. A breakthrough was brought by the *Deep Belief Network* (DBN) Hinton and Salakhutdinov [2006], which introduced a layer-by-layer unsupervised strategy to pre-train deep models. Unsupervised training learns a good model of the input that allows reconstruction or generation of input data. In [Hinton and Salakhutdinov, 2006], the model is constructed as a stack of Restricted Boltzmann Machines (RBM) [Smolensky, 1986; Hinton, 2002] that are trained in sequence to model the distribution of inputs; the output of each RBM layer is the input of the next layer. The whole network is then trained (or "fine-tuned") with a supervised algorithm.

Other models used as building blocks of deep networks include semi-supervised embedding models (*e.g.*, [Collobert and Weston, 2008; Weston et al., 2008]), denoising auto-encoders (*e.g.*, [Vincent et al., 2008]), and sparse auto-encoders (*e.g.*, [Kavukcuoglu et al., 2008; Jarrett et al., 2009]).

Deep models have been used in image classification tasks [Larochelle et al., 2007; Lee et al., 2008; Ranzato et al., 2007a; Lee et al., 2009a]. A disadvantage of those architectures is that their depth imply a large number of coefficients to be learned and often require to solve complex and highly non-convex optimization problems [Bengio et al., 2007], but they have recently encountered much success for specific tasks that can count on very large training sets [Krizhevsky et al., 2012].

2.5.3 Part-based Category Models

Part-based category models arise from the observation that many objects consist of a set of individual parts that are arranged in some characteristic geometry. Faces, for example, consist of eyes, a nose, and a mouth, while airplanes consist of wings, a fuselage, and a tail. Part-based category models exploit that observation by decomposing an object into its component parts and then modeling the visual appearance of each part individually for each object category. Those models also include some constraints on the relative spatial configuration of the parts (for illustration, see Figure 2.4).

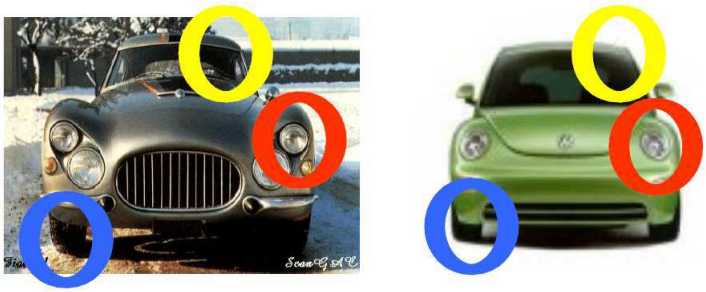
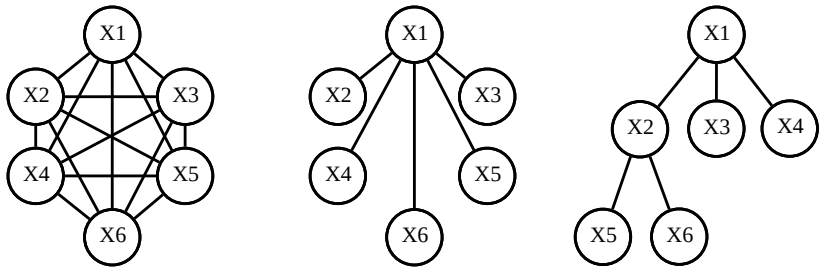


Figure 2.4: Two images containing different cars. The colored circles indicate regions of both instances that are similar in visual appearance and relative position (Figure from [Fergus, 2005]).

The concept of part-based models dates back at least to 1973 in the work of Fischler and Elschlager [1973]. They proposed an appearance model for each individual part, along with a spatial model that intuitively consists of springs connecting some of the parts. That model has been extended in many directions and it has been applied to a number of computer vision problems, in particular, object detection tasks [Leibe et al., 2008; Felzenszwalb et al., 2010; Ott and Everingham, 2011; Zhu et al., 2010; Azizpour and Laptev, 2012].

The part-based approaches can be divided into two categories [Kumar et al., 2009]: generative and discriminative. *Generative models* are typically learnt by maximizing the likelihood of a single class (*i.e.*, it uses only the data of the object class to be modeled) [Burl et al., 1998; Weber et al., 2000; Fergus et al., 2003; Fei-Fei et al., 2007]. By contrast, *discriminative models* attempts to learn a model which utilizes the data from all object classes to discriminate one class from the other (*i.e.*, it creates an explicit decision boundary separating the classes of interest) [Holub and Perona, 2005; Ramanan and Sminchisescu, 2006; Felzenszwalb et al., 2008].

Also, part-based models differ in the way the spatial relations among the individual parts are defined. In Figure 2.5, we give an overview over the most popular



(a) Constellation (b) Star (c) Tree

Figure 2.5: Overview of different spatial configuration of the part-based category models in the literature (Figure adapted from [Carneiro and Lowe, 2006]). Each X_i represent one of the "parts" of the models for one particular object category.

designs. In the following, we approach two popular part-based models for object categorization: the Constellation Model (fully connected model) and the Implicit Shape Model (star-based model).

The *Constellation Model* (CM), proposed by Burl et al. [1998]; Weber et al. [2000]; Fergus et al. [2003], represents objects by estimating a joint appearance and shape distribution of their parts. It has been designed with the goal of learning with 'weak' supervision. That is, neither the part assignments, nor even object bounding boxes are assumed to be known, only the object labels are provided. A drawback of that model is that (as fully connected model) it requires an exponentially growing number of parameters as the number of parts increases, which severely restricts its applicability for complex visual categories.

Fei-Fei and Perona [2005]; Fei-Fei et al. [2007] introduced a hierarchical Bayesian version of the CM to use priors derived from previously learned classes in order to speed up learning of a new class. Also, Fergus et al. [2005] proposed an updated version of the CM incorporating a star topology. In the *Star Model*, each part is only connected to a central reference part. Given this reference position, each part is treated independently of the others.

The idea of the Star Model can be generalized to a *Tree Model*, where each part's location is only dependent on the location of its parent. This type of model is used in the *Pictorial Structures* [Fischler and Elschlager, 1973] and it has become popular for object detection tasks [Felzenszwalb and Huttenlocher, 2005; Felzenszwalb et al., 2010; Ott and Everingham, 2011; Azizpour and Laptev, 2012] and articulated body pose analysis [Ramanan et al., 2007; Ferrari et al., 2008; Yang and Ramanan, 2011].

Learning such models, however, involves the optimization of a non-convex cost function over a set of latent variables standing for image locations of object parts and mixture component assignment. The success of training such models, hence, depends on the good initialization of model parts in general [Parizi et al., 2012].

In contrast to the CM, the *Implicit Shape Model* (ISM) [Leibe et al., 2004, 2008] requires labeled training examples, which should include a bounding box for each training object in order to know the object location and scale. As the name suggests, in the ISM the object shape is only defined implicitly by the information which parts agree on the same reference point.

The basic idea underlying ISM is to perform object category recognition and instances localization based on a non-parametric probability mass function of the position of the object center. Those probability functions come from a probabilistic interpretation of the voting space of a Generalized Hough Transform algorithm. Votes are casted by local descriptors that are matched against a visual codebook learned[7], together with votes, from a set of training examples. The advantage of this model, compared to CM, is its computational efficiency.

Generally speaking, part-based models have the advantage that they can deal with object shapes that are not well-represented by a bounding box with fixed aspect ratio. They have therefore be applied for recognizing deformable object categories [Grauman and Leibe, 2011]. However, the major disadvantage of those models is that the number of parts needed to represent different classes grows linearly with each new class that is added.

It is perhaps worth mentioning that part-based models are somewhat related to the Bag-of-Words model, if we account for the fact that, unlike the orderless BoW, a part-based model learns a deformable *arrangement* of features that represent an *object class*. Similarly to BoW, part-based models often start with local low-level feature detection and description stages. However, diverging from a BoW approach, part-models must account for the arrangement of the parts, and often they do so in a generative way, trying to model separately each object class.

2.6 Conclusion

In this chapter, we introduced the main concepts and techniques applied in this dissertation. We surveyed the three-layer pipeline to visual recognition problem: (i) low-level

[7]ISM codebooks model not only the appearance features of individual parts (as Bag-of-Words codebooks) but also the relative positions among them.

visual feature extraction, (ii) mid-level feature extraction, and (iii) supervised classification. We gave special attention to mid-level image representations, more specifically the Bag-of-Words models.

We observed that a large number of novel mid-level representations based on the BoW model have proposed in the past three years, and both steps of coding and pooling have been subject of important improvements. Briefly, these ameliorations have been done in two ways: (i) by expressing features as combinations of codewords (*e.g.*, soft assignment [van Gemert et al., 2010; Liu et al., 2011a]), or/and (ii) by preserving the difference between the features and the codewords (*e.g.*, Fisher Vector [Perronnin et al., 2010c], Super-Vector Coding [Zhou et al., 2010], VLAT [Picard and Gosselin, 2011]). The latter generates the steady inflation of feature vector sizes.

In Chapter 4, we introduce our BoW-based image representation, which takes into account SIFT descriptors densely sampled at multiple scales. As the low-level feature extraction has a big influence on the quality of the results, to make the comparisons fair, we apply the same descriptors for all techniques evaluated in this dissertation. Most importantly, SIFT descriptors were used by those techniques in their original papers. Additionally, SIFT still seems the most appealing descriptor for practical uses, and also the most widely used nowadays.

Also, our image representation relies on a soft-based coding, which is is conceptually simpler and computationally more efficient compared with existing coding schemes. It involves no optimization and only needs to compute the distance of a local feature to each word. Furthermore, we notice that by carefully adjusting the pooling step, relatively simple systems of local descriptors and classifiers can become competitive with respect to more complex ones. In this dissertation, we propose a new pooling operation.

In addition, to account for spatial information in our BoW-based method, we employ the spatial pyramids approach. It provides a reasonable coverage over the image space with scale information, and most existing classification methods either use them directly, or use slightly modified/simplified versions.

Finally, to train our mid-level features vectors, we choose to apply the popular and efficient SVM classifier using kernel similarity function adapted to the image signature.

Chapter 3

Challenges and Benchmarks Addressed

Over the last decade, progress in image classification has been quantifiable thanks to the availability of benchmark image datasets with ground truth labels and standard evaluation protocols. Those benchmarks provide a common ground for researchers to compare their methods, besides provoking discussion in the community about the types of imagery and annotation on which we should focus. In addition, in recent years, dedicated workshops have been held at major vision meetings for groups to compete with their algorithms on novel test sets. Organized annually from 2005 to present, the PASCAL Visual Object Classes challenge is a prime example.

In order to evaluate our approach, we use a wide range of datasets. This chapter gives details about each dataset and discusses how they differ among themselves. We review each dataset, commenting on its relative challenges, such as intra-class variability, viewpoint changes, occlusion, amount of training data, background clutter. We also report the best published results for each dataset, restricting ourselves to results that employ only visual information, since some tasks allow the use of other types of media or metadata.

3.1 MIRFLICKR Challenge

The MIRFLICKR dataset (or MIRFLICKR-25000) [Huiskes and Lew, 2008] contains 25,000 images collected from the Flickr photo-sharing social network[1], with associate labels and tags. In our experiments, we only consider as features the visual image content. The dataset is split into a collection of 15,000 training images and 10,000 test images, as defined by the standard challenge "Visual Concept/Topic Recognition" [Huiskes and Lew, 2008]. Example images are shown in Figure 3.1.

[1] http://www.flickr.com

Figure 3.1: Example images from MIRFLICKR dataset [Huiskes and Lew, 2008] with their associated concepts labels. The images are annotated for 24 potential concepts, and 14 relevant concepts (marked with (r)), see the text for more details.

All images are manually annotated for 24 concepts, including categories that describe the presence of specific object (*car*, *bird*, *dog*), categories that are concrete but less spatially localized (*clouds*, *night*, *sky*) and more abstract categories (*indoor*, *food*, *structures*, *transport*).

The annotation process is divided into two main stages. First, people are asked whether the image is potentially relevant to the concept: to have a positive annotation, the concept must be at least visible or recognizable in a given image. In a second stage, the annotation is applied for a subset of 14 concepts by selecting only images in which the topic is considered to be present with a strong evidence (*e.g.*, object that are large or clearly visible in the image). Finally, each image is thus annotated for 38 concepts. We use a "(r)" for concepts to refer to the latter annotation.

Table 3.1 summarizes, for each concept, the number of images in the training and test sets. The amount of training images varies greatly from concept to concept. For instance, while the *plant life* concept has 5,259 training images, the *baby(r)* concept contains only 71 training images. Also, MIRFLICKR images display different levels of difficulty, including reasonable levels of occlusion and viewpoint variation, a higher degree of intra-class variability, and concepts embedded in complex background clutter.

It is worth noting that the MIRFLICKR collection is a multi-label image classification dataset, which means multiple concepts may occur in the same image. In the past, multi-label annotation was rare in Computer Vision challenges, but nowadays it is becoming more common, since it reflects the real nature of images, that may represent more than one concept.

Table 3.1: Number of images for each concept in MIRFLICKR dataset.

Concepts	#train	#test	Concepts	#train	#test
1: animals	1950	1266	20: male(r)	2194	1453
2: baby	152	107	21: night	1593	1118
3: baby(r)	71	45	22: night(r)	392	277
4: bird	439	303	23: people	6213	4160
5: bird(r)	288	196	24: people(r)	4685	3164
6: car	719	458	25: plant life	5259	3504
7: car(r)	232	148	26: portrait	2333	1598
8: clouds	2250	1450	27: portrait(r)	2270	1559
9: clouds(r)	813	537	28: river	540	354
10: dog	418	266	29: river(r)	88	61
11: dog(r)	359	231	30: sea	806	516
12: female	3682	2502	31: sea(r)	131	83
13: female(r)	2363	1619	32: sky	4731	3181
14: flower	1132	691	33: structures	5964	4028
15: flower(r)	677	400	34: sunset	1303	832
16: food	591	399	35: transport	1736	1159
17: indoor	4978	3335	36: tree	2762	1921
18: lake	479	312	37: tree(r)	396	272
19: male	3656	2425	38: water	1988	1343

The classification performance is evaluated using the standard metric for this dataset, which is the Mean Average Precision (mAP). Precision-recall graphs can also be used to report in more detail the performance on the test set.

The baseline MIRFLICKR dataset result [Huiskes et al., 2010] is 37.0% mAP. The authors employed only global descriptors and the SVM classifier. Moreover, Guillaumin et al. [2010] reported 53.0% mAP applying many (local and global) descriptors. Again, classification is performed with a SVM classifier.

The MIRFLICKR image collection can be downloaded and redistributed without fees or registration.

3.2 ImageCLEF Evaluation Campaign

The ImageCLEF Evaluation Campaign was introduced in 2003 as part of CLEF (Cross Language Evaluation Forum). Motivated by the need to support multilingual users from a global community accessing the ever growing body of visual information, the main aims of ImageCLEF were: (i) to develop the necessary infrastructure for the evaluation of visual information systems, (ii) to provide reusable resources for such

benchmarking purposes, and (iii) to promote the exchange of ideas towards the further advancement of the field of visual media analysis, indexing, classification, and retrieval.

Since 2011, the Photo Annotation task has been based on various subsets of the MIRFLICKR-1M collection [Huiskes et al., 2010]. Every year the list of concepts to detect has been updated in order to cover a wider selection of concept types and to make the task more challenging. There are three subtasks that allow the use of different information (i) visual information only; (ii) Flickr user tags; (iii) multi-modal information, considering both visual information and Flickr user tags, in addition (optionally) to EXIF metadata contained in the images.

In this review, and in our experiments (Chapter 5), we consider only the visual-only subtask. This task aims at the automated annotation of consumer photos with multiple concepts. Further information can be found in the ImageCLEF book [Müller et al., 2010], which describes the formation, growth, resources, tasks, and achievements of ImageCLEF.

3.2.1 ImageCLEF 2011 Photo Annotation Challenge

The ImageCLEF 2011 Photo Annotation task [Nowak et al., 2011] poses the challenge of an automated annotation of Flickr images with 99 visual concepts. The dataset consists of 18,000 images, split into 8,000 training images and 10,000 test images. The image set is annotated with concepts that describe the scene (*e.g.*, *indoor*, *outdoor*, *landscape*), depicted objects (*e.g.*, *car*, *animal*, *person*), the representation of image content (*e.g.*, *portrait*, *graffiti*, *art*), events (*e.g.*, *travel*, *work*), quality issues (*e.g.*, *overexposed*, *underexposed*, *blurry*) or even sentiment concepts (*e.g.*, *happy*, *euphoric*, *melancholic*, *scary*). Example images are shown in Figure 3.2.

The relevance assessments for the annotation task was acquired by crowdsourcing, using the Amazon Mechanical Turk (AMT). AMT is an online work marketplace which distributes mini-jobs, called HITs (Human Intelligence Tasks), among a crowd of people, the "turkers". Those turkers can choose the HITs they would like to perform (in exchange for small amounts of money, usually between 0.10 and 1.00 American dollars per task) and submit the results to AMT. The proposer of the work collects the results from AMT and can approve or reject the work of each turker.

The construction of the ground truth considers the majority vote for each image. However, in some cases, no clear answer is obtained, in particular for the sentiment concept annotation, which is very subjective. In that case, the annotation for that kind of concept is discarded for that particular image [Nowak et al., 2011]. The authors report that this situation occurred in about 14–15% of the images. Figure 3.3 illustrates

Figure 3.2: Example images from ImageCLEF 2011 Photo Annotation dataset [Nowak et al., 2011] with their associated concepts labels.

Figure 3.3: Example images from ImageCLEF 2011 Photo Annotation dataset [Nowak et al., 2011] that have no sentiment information.

some of those problematic images without sentiment concept annotation.

Table 3.2 summarizes, for each concept, the number of images in the training and test sets. As is often the case on this kind of dataset, the amount of training images varies greatly from concept to concept. However, and more importantly, we can notice that the distribution of concepts is not the same between the training and test sets: some concepts are underrepresented in the training set and overrepresented in the test set, or vice versa (*e.g.*, *male*, *beach holidays*). That divergence is problematic for learning schemes that assume that the training set distribution of concepts is a prior about the relative abundance of those concepts "in the world".

In the ImageCLEF 2011 dataset, there are huge variations in viewpoint, illumination and occlusions. The cluttered backgrounds, the large intra-class variability and sometimes small inter-class variability also makes this dataset very challenging.

In order to evaluate the quality of the annotations, three measures are applied: one for the evaluation per concept and two for the evaluation per photo. The evaluation per concept is performed with the Mean interpolated Average Precision (mAP). The evaluation per example is performed with the example-based F-Measure (F-ex) and the Semantic R-Precision (SR-Precision).

Table 3.2: Number of images for each concept in ImageCLEF 2011 dataset.

Concepts	**#train**	**#test**	**Concepts**	**#train**	**#test**
1: party life	293	414	51: street	715	624
2: family friends	1109	1525	52: church	81	78
3: beach holidays	154	279	53: bridge	105	98
4: buildings sights	888	1088	54: park garden	621	569
5: snow	127	128	55: rain	37	25
6: city life	1142	1578	56: toy	206	297
7: landscape nature	1362	1661	57: musical instrument	87	72
8: sports	145	241	58: shadow	397	393
9: desert	30	27	59: body part	538	613
10: spring	105	56	60: travel	415	446
11: summer	887	764	61: work	237	249
12: autumn	153	182	62: birthday	43	35
13: winter	208	197	63: visual arts	3346	3058
14: indoor	1894	2228	64: graffiti	124	99
15: outdoor	4173	5032	65: painting	325	319
16: plants	1865	2642	66: artificial	862	906
17: flowers	367	413	67: natural	4594	5944
18: trees	890	1252	68: technical	533	392
19: sky	1977	2692	69: abstract	376	124
20: clouds	1104	1425	70: boring	483	567
21: water	761	1130	71: cute	3910	4932
22: lake	89	134	72: dog	211	238
23: river	130	171	73: cat	61	53
24: sea	222	324	74: bird	183	199
25: mountains	230	382	75: horse	28	34
26: day	4199	5049	76: fish	25	36
27: night	530	661	77: insect	91	88
28: sunny	1155	1545	78: car	268	380
29: sunset sunrise	362	404	79: bicycle	61	126
30: still life	651	839	80: ship	79	118
31: macro	705	1495	81: train	59	78
32: portrait	984	1533	82: airplane	41	50
33: overexposed	91	160	83: skateboard	12	6
34: underexposed	425	444	84: female	1254	1147
35: neutral illumination	7484	9396	85: male	2178	933
36: motion blur	244	332	86: baby	68	90
37: out of focus	150	164	87: child	176	270
38: partly blurred	2366	2676	88: teenager	413	270
39: no blur	5240	6828	89: adult	1461	2006
40: single person	1573	1955	90: old person	144	185
41: small group	723	886	91: happy	1402	1858
42: big group	236	349	92: funny	1012	1543
43: no persons	5468	6810	93: euphoric	280	193
44: animals	739	838	94: active	1242	1216
45: food	264	365	95: scary	443	419
46: vehicle	594	899	96: unpleasant	856	1103
47: aesthetic impression	1399	1771	97: melancholic	1226	1217
48: overall quality	1677	1405	98: inactive	1639	2752
49: fancy	1154	1284	99: calm	2045	2468
50: architecture	1135	955			

Most approaches with reported results on the ImageCLEF 2011 Photo Annotation dataset employ complex combinations of several low-level features to achieve good results. The best system during the competition (and the best published result to date) [Binder et al., 2011] reported 38.8% mAP, employing nonsparse multiple kernel learning and multi-task learning. They apply SIFT and color channel combinations to build different extensions of the BoW models. The system of Su and Jurie [2011] uses many features aggregating them into a global histogram using a BoW approach. In addition, Fisher Vectors were used as enhancement of the BoW model. They achieved 38.2% mAP. The method of van de Sande and Snoek [2011] reported 36.7% applying several color SIFT features with Harris-Laplace and dense sampling.

The ImageCLEF 2011 Photo Annotation dataset is not publicly available, but access to the data can be granted after a license agreement is signed[2].

3.2.2 ImageCLEF 2012 Photo Annotation Challenge

The ImageCLEF 2012 Photo Annotation dataset [Thomee and Popescu, 2012] consists of 25,000 images, split into 15,000 training images and 10,000 test images. In this edition, the Photo Annotation task continued along the lines of previous years in terms of concepts. In total, the dataset contains 94 concepts, categorized as natural elements (*e.g.*, *day*, *snow*, *fire*), environment (*e.g.*, *coast*, *plant*, *bird*), people (*e.g.*, *baby*, *female*, *small group*), image elements (*e.g.*, *in focus*, *home life*, *happy*), and human elements (*e.g.*, *car*, *bicycle*, *air vehicle*).

In comparison with the ImageCLEF 2011 Photo Annotation dataset, a few concepts were removed (*e.g.*, *beach holidays*, *neutral illumination*, *aesthetic impression*) because they were not sufficiently present in the dataset, or it was decided they were ambiguously defined, based on feedback given by former participants. Furthermore, in order to provide a more realistic context for the task, several new concepts were added, inspired by popular queries issued to the Yahoo! image search engine[3].

The ground truth for the newly defined concepts, as well as for the concepts reused from the ImageCLEF 2011 Photo Annotation dataset, was also acquired with the Amazon Mechanical Turk, a crowdsourcing platform (see the previous section). However, due to the experience with turkers without genuine interest in performing well the requested service, Thomee and Popescu [2012] used the intermediary service of CrowdFlower[4] to obtain the relevance judgments. This service automatically performs the filtering of the workers based on the quality of the work they perform by validating

[2]http://imageclef.org/2011/Photo
[3]http://images.yahoo.com
[4]http://crowdflower.com/

it against specific examples for which the correct answer is known. Such examples are commonly referred to as *gold* and need to be supplied in addition to the job.

Table 3.3 summarizes, for each concept, the number of images in the training and test sets. The amount of training images in the ImageCLEF 2012 Photo Annotation dataset still varies considerably from concept to concept. However, this year, the relative abundance of a concept is roughly the same between the training and the test sets. Thomee and Popescu [2012] decided to be of paramount importance to assure that concepts with few images are sufficiently present in both sets and in balance with each other. Other than that, the challenges in the ImageCLEF 2012 dataset are similar to the previous collection: variations in viewpoint, illumination and occlusions, cluttered backgrounds, large intra-class variability and small inter-class variability.

In order to evaluate the quality of the annotations, three measures are applied: Mean Average Precision (mAP), Geometric Mean Average Precision (GmAP), and F-Measure (F-ex). The evaluation per concept is performed with mAP and GmAP. The evaluation per example is performed with F-ex.

The best performance during the competition (and the best published result to date) was the one obtained by Liu et al. [2012a], who reported 34.8% mAP, applying a combination of the top 5 features among the 24 visual features for each concept based on a late fusion scheme. Moreover, they applied BoW models and soft assignment. The method of Ushiku et al. [2012] also uses numerous descriptors and Fisher Vectors to achieve 32.4% mAP. The approach of Xioufis et al. [2012] reported 31.8% mAP employing several descriptors which are used by different visual representations (BoW, VLAD and VLAT). Furthermore, the authors applied a late fusion scheme.

Like the previous year, the ImageCLEF 2012 Photo Annotation dataset is not available to the general public, but can be download after a license agreement is signed[5].

The results in the ImageCLEF 2012 Photo Annotation task are particularly important for this dissertation, since we were ranked at the 2nd place out of 13 participants and 28 submissions, considering only visual-based approaches (see Section 5.4).

3.3 PASCAL VOC Challenge

The PASCAL Visual Object Classes (VOC) challenge is a benchmark in visual object category recognition and detection, providing the machine learning community a standard dataset of images and annotation, in addition to standard evaluation procedures. Organized annually from 2005 to present, the goal of the PASCAL VOC challenge is

[5]http://imageclef.org/2012/Photo

Table 3.3: Number of images for each concept in ImageCLEF 2012 dataset.

Concepts	#train	#test	Concepts	#train	#test
1: timeofday_day	4897	3325	48: quantity_two	682	432
2: timeofday_night	685	431	49: quantity_three	213	127
3: timeofday_sunrisesunset	508	348	50: quantity_smallgroup	313	239
4: celestial_sun	363	224	51: quantity_biggroup	383	223
5: celestial_moon	101	68	52: age_baby	81	81
6: celestial_stars	44	25	53: age_child	400	256
7: weather_clearsky	1105	705	54: age_teenager	313	220
8: weather_overcastsky	694	433	55: age_adult	3536	2306
9: weather_cloudysky	1196	812	56: age_elderly	225	127
10: weather_rainbow	33	18	57: gender_male	2484	1660
11: weather_lightning	167	125	58: gender_female	2619	1721
12: weather_fogmist	168	100	59: relation_familyfriends	816	563
13: weather_snowice	100	91	60: relation_coworkers	239	136
14: combustion_flames	68	35	61: relation_strangers	335	212
15: combustion_smoke	71	47	62: quality_noblur	9639	6421
16: combustion_fireworks	54	18	63: quality_partialblur	3549	2293
17: lighting_shadow	861	576	64: quality_completeblur	100	83
18: lighting_reflection	448	273	65: quality_motionblur	287	176
19: lighting_silhouette	475	314	66: quality_artifacts	318	199
20: lighting_lenseffect	530	344	67: style_pictureinpicture	113	64
21: scape_mountainhill	295	218	68: style_circularwarp	167	141
22: scape_desert	73	36	69: style_graycolor	306	219
23: scape_forestpark	451	303	70: style_overlay	567	371
24: scape_coast	766	436	71: view_portrait	1533	1069
25: scape_rural	361	237	72: view_closeupmacro	2340	1589
26: scape_city	906	572	73: view_indoor	2061	1399
27: scape_graffiti	324	184	74: view_outdoor	4856	3259
28: water_underwater	53	44	75: setting_citylife	1676	1128
29: water_seaocean	369	197	76: setting_partylife	368	256
30: water_lake	135	75	77: setting_homelife	945	645
31: water_riverstream	181	115	78: setting_sportsrecreation	506	283
32: water_other	399	255	79: setting_fooddrink	626	430
33: flora_plant	419	262	80: sentiment_happy	1146	840
34: flora_tree	2129	1343	81: sentiment_calm	2119	1441
35: flora_flower	719	508	82: sentiment_inactive	1262	877
36: flora_grass	859	548	83: sentiment_melancholic	880	594
37: fauna_cat	106	72	84: sentiment_unpleasant	623	447
38: fauna_dog	361	267	85: sentiment_scary	377	278
39: fauna_horse	64	40	86: sentiment_active	1087	735
40: fauna_fish	49	39	87: sentiment_euphoric	189	140
41: fauna_bird	352	219	88: sentiment_funny	765	557
42: fauna_insect	137	114	89: transport_cycle	220	142
43: fauna_spider	16	11	90: transport_car	500	321
44: fauna_amphibianreptile	40	27	91: transport_truckbus	69	44
45: fauna_rodent	59	46	92: transport_rail	93	61
46: quantity_none	10335	6989	93: transport_water	187	127
47: quantity_one	3084	1990	94: transport_air	89	50

to investigate the performance of recognition methods on a wide spectrum of natural images.

The PASCAL VOC 2007 dataset [Everingham et al., 2007] consists of annotated consumer photographs collected from the Flickr photo-sharing website. The goal of this challenge is to recognize 20 visual object classes in realistic scenes (*i.e.*, not pre-segmented objects). Those object classes are categorized as person (*person*), animal (*bird*, *cat*, *cow*, *dog*, *horse*, *sheep*), vehicle (*aeroplane*, *bicycle*, *boat*, *bus*, *car*, *motorbike*, *train*), and indoor objects (*bottle*, *chair*, *dinning table*, *potted plant*, *sofa*, *tv/monitor*). In total, there are 9,963 images. Some example images are shown in Figure 3.4.

Figure 3.4: Example images from PASCAL Visual Object Classes 2007 dataset [Everingham et al., 2007] with their associated class labels.

The VOC 2007 challenge contains two main tasks: classification and detection. In our experiments, we show the results for the classification task. The dataset is split into three subsets: training (2,501 images), validation (2,510 images) and test (4,952 images). Our experimental results are obtained on 'trainval'/test sets (see Chapter 5).

In order to evaluate the classification challenge, the image annotation includes (in addition to class labels) the attribute 'difficult' for every object in the target set of object classes. An object is marked as 'difficult' when it is hard to recognize, for example, when it is very small, or considerably occluded, so it is hard to identify without substantial use of context. Objects marked as difficult are currently ignored in the evaluation of the challenge [Everingham et al., 2007]. We, too, have opted to ignore difficult objects.

Table 3.4 summarizes the number of images (containing at least one object of a

Table 3.4: Number of images for each class in PASCAL VOC 2007 dataset.

Class	**#train**	**#val**	**#test**	**Class**	**#train**	**#val**	**#test**
1: aeroplane	112	126	204	11: dining table	97	103	190
2: bicycle	116	127	239	12: dog	203	218	418
3: bird	180	150	282	13: horse	139	148	274
4: boat	81	100	172	14: motorbike	120	125	222
5: bottle	139	105	212	15: person	1025	983	2007
6: bus	97	89	174	16: potted plant	133	112	224
7: car	376	337	721	17: sheep	48	48	97
8: cat	163	174	322	18: sofa	111	118	223
9: chair	224	221	417	19: train	127	134	259
10: cow	69	72	127	20: tv/monitor	128	128	229

given class) for each class in the training, validation and test sets. The data has been split into 50% for training/validation and 50% for test. The distributions of images by class are approximately equal across the training/validation and test sets.

The PASCAL VOC 2007 dataset is an image classification benchmark, which contains significant variability in terms of object size, orientation, pose, illumination, position and occlusion. Moreover, the VOC 2007 annotation procedure was designed to be consistent, accurate and exhaustive for the classes. The dataset is freely available[6].

The classification performance is measured by the precision/recall curve. The principal quantitative measure used is the Mean Average Precision (mAP).

The best system during the competition [Everingham et al., 2007] reported 59.4% mAP using multiple feature channels and non-linear SVMs. van Gemert et al. [2010] reported 60.5% mAP employing many channels and and soft assignment. Yang et al. [2009a] also use many feature channels and multiple kernel learning to achieve 62.2% mAP. Using only SIFT descriptors, Zhou et al. [2010] reported 64.0% mAP employing the Super Vector (SV) coding. However, Chatfield et al. [2011] showed that the best reproducible result for SV coding is 58.2%. Moreover, Chatfield et al. achieved 61.7% for Fisher Vector representation, using SIFT descriptors extremely dense. The best published result for the PASCAL VOC 2007 dataset is 68.3% reported by Znaidia et al. [2012], but employing as features the image tags. Without access to that information, their Bag-of-Words baseline drops to 52.1%.

[6]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

3.4 15-Scenes Dataset

The 15-Scenes dataset [Lazebnik et al., 2006] contains 4,485 images of 15 natural scene categories, in which 8 categories (*highway*, *inside city*, *tall building*, *street*, *forest*, *coast*, *mountain*, *open country*) are provided by Oliva and Torralba [2001], 5 categories (*suburb*, *bedroom*, *kitchen*, *living room*, *office*) are provided by Fei-Fei et al. [2004] and 2 categories (*industrial*, *store*) are provided by Lazebnik et al. [2006]. Each category has 210 to 410 images, and the average image size is 300×250 pixels. The major sources of images in the dataset include the COREL collection, personal photographs, and Google image search. Example images are shown in Figure 3.5.

Figure 3.5: Example images from 15-Scenes dataset [Lazebnik et al., 2006] with their associated class labels.

Table 3.5 summarizes for each category the number of images. The standard benchmarking protocol consists in randomly selecting 100 training images per category and using the remaining ones for the test. The classification performance is measured by the average recognition rates over N random training/test splits. The final result is reported as the mean and standard deviation of the results from the individual splits. Usually, a confusion table is used to illustrate the results.

The 15-Scenes is a single-label image classification dataset, unlike the previous multi-label datasets (MIRFLICKR [Huiskes and Lew, 2008], ImageCLEF Photo Annotation [Nowak et al., 2011; Thomee and Popescu, 2012] and VOC 2007 [Everingham et al., 2007]). Also, although the 15-Scenes dataset contains less classes and intra-class variation is smaller, this dataset is relatively widely used within the community for evaluating image classification. It is publicly available[7].

The authors of the 15-Scenes dataset [Lazebnik et al., 2006] reported 81.4% performance accuracy employing the spatial pyramid approach. Recently, Krapac et al.

[7]http://www-cvr.ai.uiuc.edu/ponce_grp/data/

Table 3.5: Number of images for each class in 15-Scenes dataset.

Class	#train	#test	Class	#train	#test
1: bedroom	100	116	9: inside city	100	208
2: suburb	100	141	10: mountain	100	274
3: industrial	100	211	11: open country	100	310
4: kitchen	100	110	12: street	100	192
5: living room	100	189	13: tall building	100	256
6: coast	100	260	14: office	100	115
7: forest	100	228	15: store	100	215
8: highway	100	160			

[2011] achieved 88.2% by using the Spatial Fisher Vector representation. The best result published is 89.8% [Gao et al., 2010] for a Laplacian sparse coding method.

3.5 Oxford Flowers Dataset

The Oxford Flowers dataset [Nilsback and Zisserman, 2006] contains 1,360 images of 17 different flower species (80 images per category). The images were acquired by searching the web and taking pictures. The dataset is separated into three different folds, each with its own training (17 × 40 images), validation (17 × 20 images) and test sets (17 × 20 images). Example images are shown in Figure 3.6.

Figure 3.6: Example images from Oxford Flowers dataset [Nilsback and Zisserman, 2006] with their associated class labels.

Classifying flowers is a difficult task even for humans. In the Oxford Flowers images, there are large variations in viewpoint and scale, illumination, partial occlusions, cluttered backgrounds. The large intra-class variability and the sometimes small

inter-class variability makes this dataset very challenging. The flower categories are deliberately chosen to have some ambiguity on each aspect. For example, some classes cannot be distinguished on color alone (*e.g.*, *dandelion* and *colt's foot*), others cannot be distinguished on shape alone (*e.g.*, *sunflower* and *daisy*).

The accuracy rate is reported by the average scores of the three folds. The final result is reported as the mean and standard deviation of the three folds.

The authors of the Oxford Flowers [Nilsback and Zisserman, 2006] have employed a BoW scheme (with 800 codewords), using SIFT descriptors and k-Nearest Neighbor classifier. They have reported 71.8% performance accuracy. Lechervy et al. [2012] has proposed a linear combination of base kernels using the boosting paradigm to achieve 88.3%. The best published result, as far as we know, is 95.2%, reported by Koniusz and Mikolajczyk [2011]. They have applied soft-BoW and sparse coding-based signatures, combined with color SIFT at kernel level. Also, they have used a kernel discriminant analysis (KDA) classifier. The dataset is freely available[8].

3.6 Conclusion

Datasets and challenge are an integral part of contemporary image recognition research. They have been one important factor in the considerable progress in the field, not just as source of large amounts of training data, but also as means of measuring and comparing performance of competing algorithms. Such databases allow recognition systems exercising the ability to handle intra-class variability, varying size and pose, partial occlusion, contextual cues, cluttered backgrounds.

Table 3.6: Summary of all datasets used in this dissertation.

Dataset	**#images**	**#class**	**#train**	**#test**	**classification measure**	**publicly available?**
MIRFLICKR	25,000	38	15,000	10,000	mAP	yes
ImgCLEF 2011	18,000	99	8,000	10,000	mAP	no
ImgCLEF 2012	25,000	94	15,000	10,000	mAP	no
VOC 2007	9,963	20	5,011	4,952	mAP	yes
15-Scenes	4,485	15	1,500	2,985	Accuracy	yes
Oxford Flowers	1,360	17	680	340	Accuracy	yes

[8]http://www.robots.ox.ac.uk/∼vgg/data/flowers/

The 6 different datasets used in this dissertation offer different challenges. The 15-Scenes is easier than others. The Oxford Flowers is more challenging. The MIR-FLICKR and PASCAL VOC 2007 are a very challenging datasets. Finally, the ImageCLEF Photo Annotation datasets offers a hard test. To summarize, statistics of all datasets mentioned are listed in Table 3.6.

Chapter 4

BossaNova Representation

The last decade has witnessed two important breakthroughs in the field of image classification: (i) the development of very discriminant *low-level* local descriptors (such as SIFT descriptors [Lowe, 2004]); and (ii) the emergence of *mid-level* aggregate representations, based on the quantization of those features, in the so-called *Bag-of-Words* (BoW) model [Sivic and Zisserman, 2003]. Those advances in feature extraction and representation have closely followed a previous turning point on statistical learning, represented by the maturity of kernel methods and support vector machines [Sebe et al., 2005; Cord and Cunningham, 2008].

BoW models can be obtained by a succession of two steps [Boureau et al., 2010a]: coding and pooling. Traditionally, the *coding* step simply associates the image local descriptors to the closest element in the codebook, and the *pooling* takes the average of those codes over the entire image. Since the pooling operation compacts all the information contained in the individually encoded local descriptors into a single feature vector, that step is critical for BoW-based representations.

In this chapter, we introduce our BossaNova representation, which is based on a new pooling strategy. Therefore, we open this chapter by reintroducing the coding/pooling formalism, using a novel matrix formalism (Section 4.1). Then, we expose how our early ideas of extending BoW pooling came to light (Section 4.2). We then introduce the complete BossaNova pooling formalism (Section 4.3). Next, we detail the computational steps of our representation: the semi-soft coding scheme (Section 4.4) and the normalization strategy (Section 4.5). Then, we describe the proposed BossaNova algorithm in pseudo-code, followed by an analysis of its computational complexity (Section 4.6). After, we analyze how BossaNova and Fisher Vector representations can be expected to complement each other well (Section 4.7). Finally, we present a generative formulation of our BossaNova strategy (Section 4.8).

4.1 Coding & Pooling Matrix Representation

As discussed in Section 2.2, the classical BoW model can be interpreted as an occurrence histogram of visual words, where the visual codebook has been trained from a set of local descriptors. The mapping of the visual codebook into image descriptors can be decomposed into a coding step followed by a pooling step, as formalized by [Boureau et al., 2010a]. In the original BoW model [Sivic and Zisserman, 2003], a vector quantization stage is applied for coding, and the codes are aggregated with an average-pooling strategy. In this section, we rediscuss those steps, using a novel matrix formalism.

Let $\mathcal{X}$ be an unordered set of local descriptors extracted from an image. $\mathcal{X} = \{\mathbf{x}_j\}$, $j \in \{1, \ldots, N\}$, where $\mathbf{x}_j \in \mathbb{R}^D$ is a local descriptor vector and N is the number of local descriptors in the image. In the BoW model, let $\mathcal{C}$ be a visual codebook obtained by an unsupervised learning algorithm (*e.g.*, k-means clustering algorithm). $\mathcal{C} = \{\mathbf{c}_m\}$, $m \in \{1, \ldots, M\}$, where $\mathbf{c}_m \in \mathbb{R}^D$ is a codeword and M is the number of visual codewords. $\mathbf{z} \in \mathbb{R}^M$ is the final vectorial representation of the image used for classification.

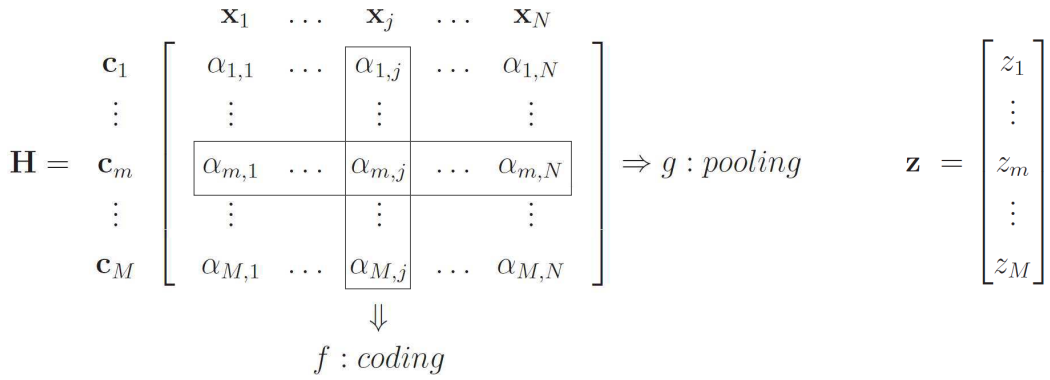
Therefore, in order to represent the coding and pooling functions, we introduce the matrix representation $\mathbf{H}$ of the BoW model, with columns $\mathcal{X}$ and rows $\mathcal{C}$. As illustrated in Figure 4.1a, the coding function f for a given descriptor $\mathbf{x}_j$ corresponds to the j^{st} column, and may be interpreted as an activation function for the codebook, activating each of the codewords according to the local descriptor. The pooling function g for a given visual codeword $\mathbf{c}_m$ corresponds to the m^{st} row, and may be understood as the aggregation of the activations of that codeword.

In the classical BoW model, the coding function activates only the $\mathbf{c}_m$ codeword closest to the local descriptor $\mathbf{x}_j$ (i.e, $\alpha_{m,j} = 1$), assigning zero weight to all others. The pooling function computes the average value of each dimension of α_j pertaining to a codeword $\mathbf{c}_m$, compacting all information into a single scalar z_m (*i.e.*, $z_m = \sum_{j=1}^{N} \alpha_{m,j}$).

As illustrated in Figure 4.1b, the vector $\mathbf{z}$ representing the whole image is given by $\mathbf{z} = [z_1, z_2, \cdots, z_M]^{\mathrm{T}}$.

4.2 Early Ideas

Our goal in this dissertation is to address the problem of classifying images based on their visual content. Nowadays, most state-of-the-art image classification systems are based on the BoW representation. Therefore, we have explored the literature of



(a) Matrix representation of the BoW model. (b) Final representation.

Figure 4.1: (a) Matrix representation $\mathbf{H}$ of the BoW model illustrating coding and pooling functions, with columns $\mathcal{X}$ related to the low-level local descriptors, and rows $\mathcal{C}$ related to the visual codewords. The coding function f for a given descriptor $\mathbf{x}_j$ corresponds to column j, and may be interpreted as how much that descriptor activates each visual codeword. The pooling function g for a given visual codeword $\mathbf{c}_m$ corresponds to a summarization of row m and may be interpreted as the aggregation of the activations of that codeword. (b) The final representation is a vector $\mathbf{z}$, containing those aggregated activations, for each visual codeword.

BoW model to be thoroughly familiar with, to identify possible shortcomings and to determine the variety of BoW-based approaches (see Section 2.2).

As we have observed in the literature, recent research has been mostly focused on coding to improve the BoW representation (*e.g.*, FV [Perronnin et al., 2010c], VLAD [Jégou et al., 2010], SVC [Zhou et al., 2010], SFV [Krapac et al., 2011], VLAT [Picard and Gosselin, 2011]). The focus on coding functions that preserve more information has been resulting the steady inflation of vector sizes.

By contrast, we have pointed out the weakness in the standard pooling operation used in the BoW signature generation: it compacts all information pertaining to a codeword into a single scalar. In general terms, the objective of pooling is to summarize the information contained in the individually encoded descriptors into a single feature vector, preserving important information while discarding irrelevant details.

From this perspective, instead of averaging all the values from one row in the $\mathbf{H}$ matrix, we propose to describe their distribution. The representation can be seen as a histogram of distances between the descriptors found in the image and each codebook element. BOSSA (Bag Of Statistical Sampling Analysis) is, therefore, an extension to the BoW approach, resulting in a new representation that better preserves the information from the encoded local descriptors, by using a density-based pooling description.

BOSSA is an early work that presents a proof-of-concept of our strategy (the achieved representation, which incorporates all enhancements, was named BossaNova and is presented in the next sections). BOSSA was first evaluated in basic experiments using the Oxford Flowers dataset (Section 3.5), which we have published in [Avila et al., 2011], and which we reproduce next.

We have compared the performance of the BOSSA representation with BoW. We have implemented a simple BOSSA strategy with an hard coding from the **H** matrix and few bins to quantify the distance-to-codeword distribution. Table 4.1 reports the classification performances for BOSSA and BoW (using their best tested configuration parameters[1]).

Table 4.1: BOSSA and BoW classification performances on the Oxford Flowers dataset [Nilsback and Zisserman, 2006]. The table shows the means and standard deviations over three accuracy measures.

	Accuracy (%)
BOSSA [Avila et al., 2011]	64 ± 2
BoW [Sivic and Zisserman, 2003]	59 ± 1

This basic experiment shows that BOSSA outperforms BoW with 8.5% relative improvement. That highlights the relevance of such a pooling strategy. It is also important to point out that, in order to better isolate the improvement due to our pooling, we do not have considered in those experiments extended representations of the BoW, like the spatial pyramid of Lazebnik et al. [2006] or others.

To provide a more comprehensive analysis of our representation, we need to further investigate all its facets. Typically, we need to analyze the range of distances used to compute each codeword histogram, and the ways to encode this histogram (number of bins). We also need to explore normalization aspects. In the BOSSA representation, the final representation merges all the local histograms computed per codeword. We already suspected that the global normalization would not be sufficient in order to ex-

[1] As a low-level local descriptor, we employed HueSIFT [van de Sande et al., 2010], a SIFT variant that includes color information. The 165-dimensional HueSIFT descriptors are extracted from 21 $\times$ 21 pixel patches on regular grids (every 6 pixels). As a result, roughly 8,500 descriptors are extracted from each image of Oxford Flowers. The codebooks are learnt by k-means clustering algorithm with Euclidean distance over one million randomly sampled descriptors. For classification, we have applied the SVM classifier, specifically a χ^2 kernel and the one-versus-all approach for multi-class approach. Kernel matrices are computed as $exp(-\gamma d(x, x'))$ with d being the distance and γ being fixed to the inverse of the pairwise distances mean.

ploit the local structure of such feature space, so we wanted to explore more powerful normalization strategies.

In short, we proposed the BOSSA representation to introduce our density function-based pooling strategy in order to keep more information than the BoW during the pooling. Preliminary results have shown the significance of such a pooling. We propose now to explore and optimize the whole image representation scheme: the local feature extraction, the extended coding techniques and the BOSSA pooling. The resulting scheme, called BossaNova, which also integrates parametrization and normalization, is presented next.

4.3 BossaNova Pooling Formalism

Our approach follows the BoW formalism, but proposes a new image representation which keeps more information than BoW during the pooling step. Thus, our pooling estimates the distribution of the descriptors around each codeword. We choose a non-parametric, density-based estimation of the descriptors distribution, by computing a histogram of distances between the descriptors found in the image and each codeword.

The proposed pooling function g estimates the probability density function of α_m: $g(\alpha_m) = \text{pdf}(\alpha_\mathbf{m})$, by computing the following histogram of distances $z_{m,b}$:

$$\begin{aligned} g : \mathbb{R}^N &\longrightarrow \mathbb{R}^B, \\ \alpha_\mathbf{m} &\longrightarrow g(\alpha_m) = z_m, \\ z_{m,b} &= \text{card}\left(\mathbf{x}_j \mid \alpha_{m,j} \in \left[\frac{b}{B}; \frac{b+1}{B}\right]\right), \\ & \quad \frac{b}{B} \geq \alpha_m^{min} \text{ and } \frac{b+1}{B} \leq \alpha_m^{max}. \end{aligned} \tag{4.1}$$

where B denotes the number of bins of each histogram z_m, and $[\alpha_m^{min}; \alpha_m^{max}]$ limits the range of distances for the descriptors considered in the histogram computation. We have observed that, due to the "curse of dimensionality" [Bellman, 1961], distances between descriptors seldom fall below a certain range, making some bins of the histograms always zero (see Figure 4.2 for illustration). The double range makes better use of the representation space.

The function g represents the discrete (over B bins) density distribution of the distances $\alpha_{m,j}$ between the codeword $\mathbf{c}_m$ and the local descriptors of an image. That is illustrated in Figure 4.3. For each center $\mathbf{c}_m$, we obtain a local histogram z_m. The colors (green, yellow and blue) indicate the discretized distances from the center $\mathbf{c}_m$

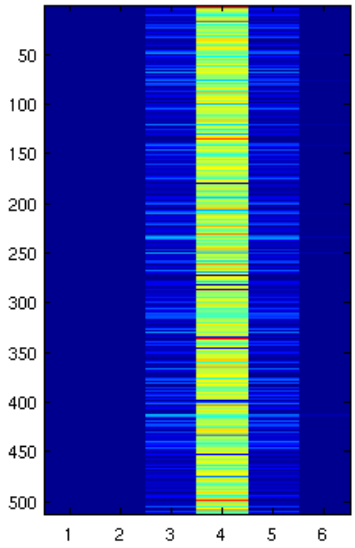


Figure 4.2: Number of SIFT descriptors assigned to each codeword at each bin in the Oxford Flowers dataset [Nilsback and Zisserman, 2006]. The graphical representation is obtained for 512 codewords and 6 bins ($[\alpha_m^{min}, \alpha_m^{max}] = [0 \cdot \sigma_m,\ 2 \cdot \sigma_m]^a$ for each visual codeword $\mathbf{c}_m$); it is coded according to a color scale, which ranges from blue (the number of SIFT descriptors is zero) to red (many numbers of SIFT descriptors).

[a] σ_m is the standard deviation of each cluster $\mathbf{c}_m$ obtained by k-means clustering algorithm.

to the local descriptors shown by the black dots. For each colored bin $z_{m,b}$, the height of the histogram is equal to the number of local descriptors $\mathbf{x}_j$, whose discretized distance to codeword $\mathbf{c}_m$ fall into the b^{th} bin. In Figure 4.3, $B = 3$. We can note that if $B = 1$, the histogram z_m reduces to a single scalar value counting the number of feature vectors $\mathbf{x}_j$ falling into center $\mathbf{c}_m$. Therefore, the proposed histogram representation can be considered as a generalization of BoW pooling step.

Note that $\alpha_{m,j}$, introduced in Figure 4.1, traditionally quantifies a similarity between the descriptor $\mathbf{x}_j$ and the codeword $\mathbf{c}_m$, while in our pooling formalism, it represents a dissimilarity (indeed, a distance). That choice makes illustrations clearer and more intuitive, and no generality is lost, since estimating a similarity probability density function for $\alpha_{m,j}$ from our model is straightforward.

After computing a local histogram z_m for all the $\mathbf{c}_m$ centers, we concatenate them to form the whole image representation. In addition, since the occurrence rate of each codeword $\mathbf{c}_m$ in the image is lost, we incorporate in our image representation $\mathbf{z}$ an additional scalar value t_m for each codeword, counting the number of local descriptors

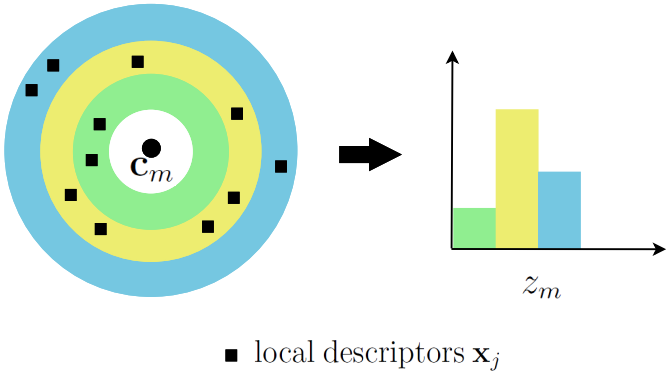


Figure 4.3: Illustration of a local histogram z_m. For each center $\mathbf{c}_m$, we obtain a local histogram z_m. The colors indicate the discretized distances from the center $\mathbf{c}_m$ to the local descriptors shown by the black dots. For each colored bin $z_{m,b}$, the height of the histogram is equal to the number of local descriptors $\mathbf{x}_j$, whose discretized distance to codeword $\mathbf{c}_m$ fall into the b^{th} bin.

$\mathbf{x}_j$ close to that codeword. That value corresponds to a classical BoW term, accounting for a raw measure of the presence of the codeword $\mathbf{c}_m$ in the image. Also, we apply a weight factor s to each t_m value[2]. Thus, our image representation $\mathbf{z}$ can be written as:

$$\mathbf{z} = \left[\left[z_{m,b}\right], st_m\right]^{\mathrm{T}}, \quad (m,b) \in \{1,\ldots,M\} \times \{1,\ldots,B\}, \tag{4.2}$$

where $\mathbf{z}$ is a vector of size $M \times (B+1)$ and M is the codebook size. The Equation 4.2 lets us interpret BossaNova as an improvement over the BoW representation, through the use of an additional term coming from the more informative pooling function.

Recently, that idea of enriching BoW representations with extra knowledge from the set of local descriptors has been explored on several representations. It can be found, for example, on Fisher Vector [Perronnin et al., 2010c] and Super-Vector Coding [Zhou et al., 2010]. Those works, however, opt by parametric models that lead to very high-dimensional image representations. By using a simple histogram of distances to capture the relevant information, our approach remains very flexible and keeps the representation compact.

Our BossaNova representation is defined by the three followings parameters: the number of codewords M, the number of bins B in each histogram z_m, and the range of distances $[\alpha_m^{min}, \alpha_m^{max}]$ – the minimum distance α_m^{min} and the maximum distance α_m^{max} in the $\mathbb{R}^D$ descriptor space that define the bounds of the histogram. In Section 5.2, we

[2]The factor s may be manually set or learned via cross-validation on a training/validation sub-set.

evaluate the key aspects of the parametric space of our representation.

In Figure 4.4, we illustrate the overview of BossaNova image classification pipeline. For low-level visual feature extraction, we extract SIFT descriptors [Lowe, 2004] on a dense spatial grid at multiple scales. As we have discussed in Section 2.1.2, that setup for low-level visual feature extraction proves to give very good performances in standard image datasets (*e.g.*, see Chatfield et al. [2011]). Next, the dimensionality of the SIFT descriptors is reduced by using PCA. It was observed [Jégou et al., 2012] that the performance of BoW (and consequently, BoW-based approaches) is improved by PCA, while, by working with the reduced representation, computational costs may also be significantly reduced.

After, our BossaNova mid-level feature vector is obtained by a succession of four steps: coding, pooling, normalization and weighting. In the coding step, instead of using hard-assignment (which introduces coding errors induced by the descriptor space quantization), we propose applying a localized soft-assignment coding (see Section 4.4), employing the soft assignment only to the n closest codewords in a visual codebook (obtained by clustering a large set of descriptors with k-means). That coding scheme achieves comparable or even better performance than existing sparse or local coding schemes [Liu et al., 2011a]. In the pooling step, we apply our density function-based pooling strategy, which computes a histogram of distances between the descriptors found in the image and each codeword. As we have shown in Section 4.2, the preliminary results demonstrated the significance of our pooling. Additionally, we compute the occurrence rate of each codeword in the image (*i.e.*, BoW term), accounting for a raw measure of the presence of each codeword in the image. In the normalization step, we propose applying a two step normalization (see Section 4.5). We employ a power-law normalization, which consistently improves the classification performance [Perronnin et al., 2010c], followed by ℓ_2-normalization, that has widely been used to normalize the BoW-based feature vectors (as we have observed in Section 2.3.2). Those normalizations are applied separately to the local histograms and BoW histogram. Finally, in the weighting step, the local histograms and the BoW terms are concatenated by applying a weight factor on the latter in order to set the relevance of each term in BossaNova mid-level feature vector.

Once we obtained the BossaNova vectors, SVM classifiers are applied by using a nonlinear Gauss-ℓ_2 kernel, since linear SVMs have been repeatedly reported to be inferior to nonlinear SVMs on BoW-based representation [Perronnin et al., 2010b; Vedaldi and Zisserman, 2012].

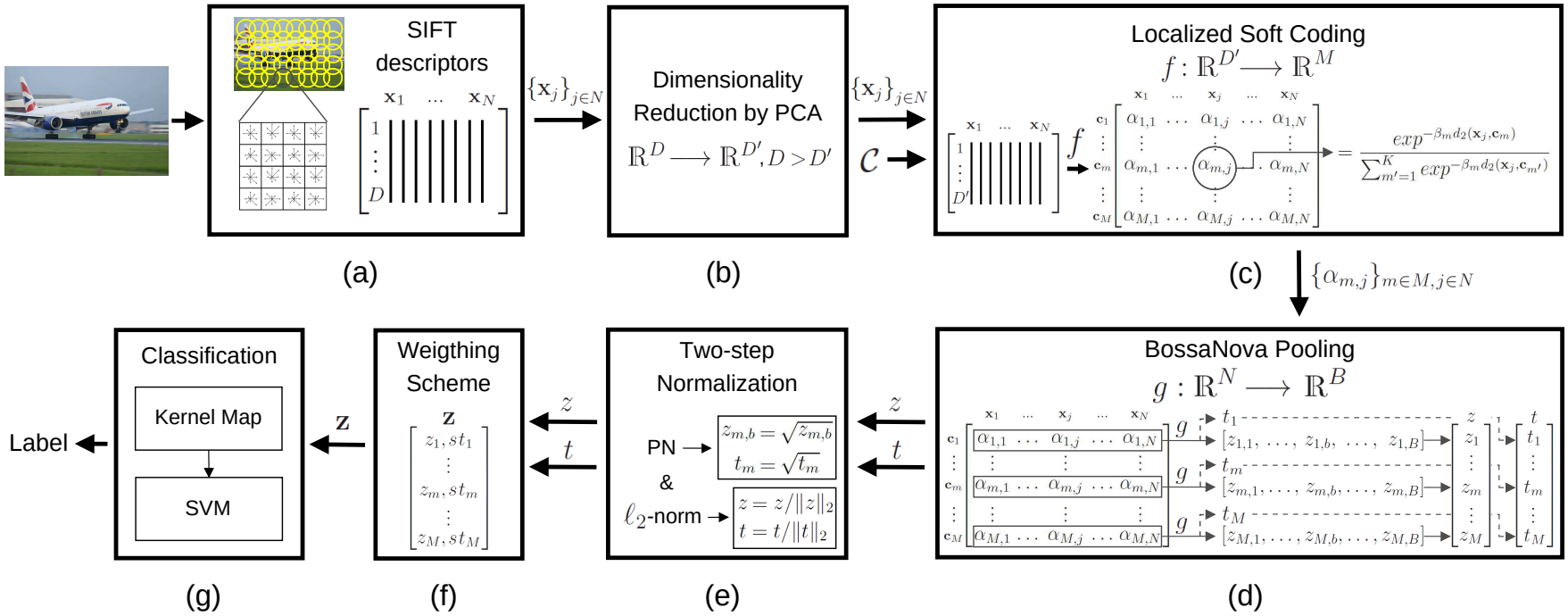


Figure 4.4: Overview of BossaNova image classification pipeline. (a) Extraction of the $\{\mathbf{x}_j\}$ low-level local descriptors (SIFT) from an image. (b) Dimensionality reduction by applying Principal Component Analysis (PCA). (c) At the coding step, the f coding function activates n closest codewords to the descriptor, which corresponds to a localized soft coding over the $\mathcal{C}$ visual codebook. M is the number of codewords. (d) Our pooling strategy: the g pooling function computes histograms of distances z_m for each $\mathbf{c}_m$ codeword. $\alpha_{m,j}$ represents a dissimilarity (*i.e.*, a distance) between $\mathbf{c}_m$ and $\mathbf{x}_j$. B is the number of bins. t_m corresponds to a classical BoW term, accounting for a raw measure of the presence of the codeword $\mathbf{c}_m$ in the image. (e) Two-step normalization: power normalization followed by ℓ_2-normalization. (f) Weighting of the histogram (z_m) and counting components (t_m), by applying a weight factor s on the latter. The vector $\mathbf{z}$, the BossaNova image representation, can be represented as $[[z_{m,b}], st_m]^{\mathrm{T}}$, where $m \in M$ and $b \in B$. (g) Classification algorithms (such as SVM classifier) are then trained on the BossaNova vectors obtained.

4.4 Localized Soft-Assignment Coding

In BossaNova coding, we propose a soft-assignment strategy, for both the local histograms and the raw counts in the feature vector. Soft assignment is chosen because it has been shown to considerably enhance the results over hard-assignment, without incurring the computational costs of sparse coding [Yang et al., 2009b; Boureau et al., 2010a]. In addition, a recent evaluation [Liu et al., 2011a] reveals that well-designed soft coding can perform as well or even better than sparse coding.

Soft-assignment coding attenuates the effect of coding errors induced by the quantization of the descriptor space. Different soft coding strategies have been presented and evaluated by van Gemert et al. [2010], the most successful approach being the one they call "codeword uncertainty". Other authors [Wang et al., 2010; Liu et al., 2011a; Boureau et al., 2011] point out the importance of locality in the coding, which leads us to a localized "semi-soft" coding scheme.

Thus, like [Liu et al., 2011a], we consider only the K-nearest codewords in coding a local descriptor, and we perform for those neighbors a "codeword uncertainty" soft assignment. Let us consider a given local descriptors $\mathbf{x}_j$, and its K closest codewords $\mathbf{c}_m$. The soft-assignment $\alpha_{m,j}$ to the codeword $\mathbf{c}_m$ is computed as follows:

$$\alpha_{m,j} = \frac{exp^{-\beta_m d_2(\mathbf{x}_j, \mathbf{c}_m)}}{\sum_{m'=1}^{K} exp^{-\beta_m d_2(\mathbf{x}_j, \mathbf{c}_{m'})}}, \tag{4.3}$$

where $d_2(\mathbf{x}_j, \mathbf{c}_m)$ is the (Euclidean) distance between $\mathbf{c}_m$ and $\mathbf{x}_j$. The parameter β_m regulates the softness of the soft-assignment (the bigger it is, the hardest the assignment). The main difference between our approach and the one of Liu et al. [2011a] is that we allow β_m to vary for each codeword, while they use a global β parameter, determined by cross-validation. Since our codewords $\mathbf{c}_m$ correspond to cluster centers obtained by a k-means algorithm, we take advantage of the standard deviation σ_m of each cluster $\mathbf{c}_m$ to setup $\beta_m = \sigma_m^{-2}$.

4.5 Normalization Strategy

In BossaNova normalization, we propose a two-step signature normalization. The first step in that normalization is motivated by the following observation: as the number of codewords increases, the local histogram becomes sparser. That is also the case for most BoW-like representations: Perronnin et al. [2010c] have also observed that effect, which is indeed a direct consequence of the ratio between the number of local

descriptors and the mid-level representation vector size. They observe that similarities become less reliable when the vector signatures become too sparse, proposing a power-law normalization to alleviate that drawback. Therefore, we choose to incorporate that normalization into the BossaNova representation.

Formally, the power normalization consists of applying the following operator in each histogram bin $z_{m,b}$:

$$f(z_{m,b}) = \text{sign}(z_{m,b})|z_{m,b}|^{\delta}, \quad 0 < \delta \leq 1. \tag{4.4}$$

In our experiments, we consider $\delta = 0.5$, which has shown in preliminary experiments to provide better performance.

The second step is an ℓ_2-normalization applied to the final vector. We apply the power-law normalization first and then the ℓ_2-normalization.

4.6 Computational Complexity

In Algorithm 1, we formally describe the BossaNova algorithm. In this section, we analyze our algorithm in terms of computational complexity.

Let $\mathcal{X}$ and $\mathcal{C}$ be the input for the algorithm. $\mathcal{X} = \{\mathbf{x}_j\}$, $j \in \{1, \ldots, N\}$ is an unordered set of local descriptors extracted from an image, where $\mathbf{x}_j \in \mathbb{R}^D$ is a local descriptor vector and N is the number of local descriptors in the image. $\mathcal{C} = \{\mathbf{c}_m\}$ is a codebook, where $\mathbf{c}_m \in \mathbb{R}^D$, $m \in \{1, \ldots, M\}$ and M is the number of codewords.

The algorithm is composed of three consecutive steps: (i) the localized soft coding and pooling scheme (lines **1:12**), (ii) the two-step normalization (lines **13:16**), and (iii) the weighting scheme (line **17**). We analyze the algorithm for each step.

Step (i): The loop of lines **1:12** is executed N times. The loop in line **2** is executed M times. The loop of lines **3:11** is executed k times (the k parameter refers to k nearest codewords), and it has an (implicit) inner loop in line **5** that executes k times. Then, the outer loop of line **1** is executed $O(N \cdot M \cdot k)$ times, where $k \ll M < N$.

Step (ii): The loop of lines **13:15** is executed $O(M \cdot B)$ times. The instruction in line **16** is also executed $O(M \cdot B)$ times, where $B \ll M$.

Step (iii): The instruction in line **17** is executed $O(M)$ times, because it involves M multiplication operations $(s \cdot t_m)$.

The running time of BossaNova algorithm is $O(N \cdot M \cdot k) + O(M \cdot B) + O(M)$. Considering the summation rule to find a total running time for the entire algorithm, we can conclude that the running time of BossaNova algorithm is $O(N \cdot M \cdot k)$.

Algorithm 1 BossaNova algorithm in pseudo-code.

Input: $\mathcal{X} = \{\mathbf{x}_j\}$, $\mathcal{C} = \{\mathbf{c}_m\}$.
Output: BossaNova representation $\mathbf{z}$.

for all $\mathbf{x}_j$ **do**
  $\forall \mathbf{c}_m$ compute $d_2(\mathbf{x}_j, \mathbf{c}_m) = \|\mathbf{x}_j - \mathbf{c}_m\|_2$
  for $i \leftarrow 0, k$ **do**
    Let $\mathbf{c}_m$ be the i nearest codeword to $\mathbf{x}_j$
    Compute $\alpha_{m,j}$ with Equation 4.3
    if $d_2(\mathbf{x}_j, \mathbf{c}_m) \in [\alpha_m^{min},\ \alpha_m^{max}]$ **then**
      $b \leftarrow \lfloor B \cdot\ (d_2(\mathbf{x}_j, \mathbf{c}_m) - \alpha_m^{min})/(\alpha_m^{max} - \alpha_m^{min}) \rfloor$
      $z_{m,b} \leftarrow z_{m,b} + \alpha_{m,j}$ {*Computation of the local histogram* z_m}
    end if
    $t_m \leftarrow t_m + \alpha_{m,j}$ {*Computation of the BoW term* t_m}
  end for
end for
for all $z_{m,b}, t_m$ **do**
  $z_{m,b} \leftarrow \sqrt{z_{m,b}}$, $t_m \leftarrow \sqrt{t_m}$ {*Power Normalization*}
end for
$z \leftarrow z/\|z\|_2$, $t \leftarrow t/\|t\|_2$ {ℓ_2*-normalization*}
$\mathbf{z} \leftarrow [[z_{m,b}], st_m]^{\mathrm{T}}$ {*Weighting* z_m *and* t_m}
return $\mathbf{z}$

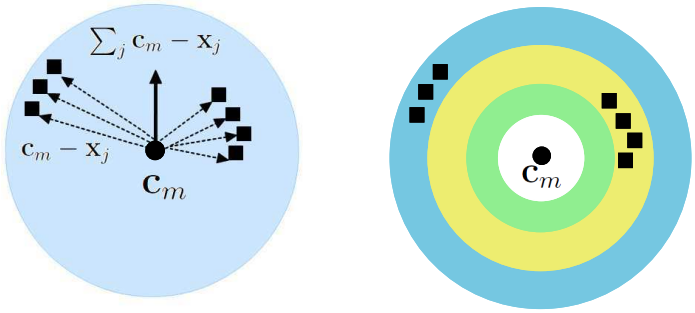
4.7 BossaNova & Fisher Vector: Pooling Complementarity

Although alternative pooling strategies have recently been explored (*e.g.*, max-pooling [Yang et al., 2009b]), average-pooling remains the most commonly employed scheme for aggregating local descriptors. As pointed out by Boureau et al. [2011], incorporating locality constraints during coding or pooling is mandatory for extracting a meaningful image representation when using average-pooling. That is especially the case for state-of-the-art local descriptors such as SIFT [Lowe, 2004] or HOG [Dalal and Triggs, 2005] that cannot be averaged without considerably loosing information. For example, if we do not consider any coding step (*i.e.*, $M = D$, $f = I_D$ in Figure 4.1), aggregating SIFT or HOG descriptors with average-pooling would pro-

duce a global histogram of gradient orientation for the image. Thus, if care is not taken, the pooling step makes the representation uninformative for classification.

In aggregated methods, such as VLAD [Jégou et al., 2010], Fisher Vector [Perronnin et al., 2010c] or Super-Vector Coding [Zhou et al., 2010], the locality constraints are mainly incorporated during the pooling step. In that class of methods, since the coding step is much more accurate (for each codeword, a vector is stored instead of a simple scalar value with standard BoW coding schemes), the authors often claim that they can afford to use a codebook of limited size (*e.g.*, $M \sim 100$) and get very good performances. However, reducing the codebook size intrinsically increases the hypervolume of each codeword in the descriptor space. That naturally decreases the range of the locality constraints that can be incorporated during pooling: all local descriptors falling into a (now larger) codeword are averaged together.

Therefore, we argue that average-pooling used in aggregate methods may lack locality, as soon as the distribution of local descriptors becomes multi-modal inside a codeword. For example, Fisher Vectors model the distribution of local descriptors in each codeword with a single Gaussian. When that Gaussian assumption does not hold, the pooled representation may be unrepresentative of the local descriptor statistics. This is illustrated in Figure 4.5.



a) Aggregate methods b) BossaNova

Figure 4.5: Aggregated methods, *e.g.* Fisher Vector [Perronnin et al., 2010c], may lack locality during pooling for small codebooks, whereas BossaNova does not. In counterpart, aggregated methods are more accurate during the coding steps, making the two representation complementary. See discussion in Section 4.7.

Figure 4.5a shows an illustration of a cluster around codeword $\mathbf{c}_m$ with local descriptors $\mathbf{x}_j$ having two different modes (*i.e.*, sub-clusters). When averaging the codes during pooling, we get for $\mathbf{c}_m$ a pooled vector $\sum_j \mathbf{c}_m - \mathbf{x}_j$ that is far away from any local descriptors $\mathbf{x}_j$. In contrast to that, BossaNova representation uses

additional locality constraints during the pooling, since only the feature vectors $\mathbf{x}_j$ that are close to the codewords $\mathbf{c}_m$ are pooled together, as shown in Figure 4.5b. The pooled representation is thus able to capture the statistics of the local descriptors.

On the other hand, when the Gaussian assumption is fulfilled, aggregated methods provide powerful signatures thanks to the improved accuracy of the coding step. The two mid-level representations are thus complementary, and we can expect improving performances by combining them. In a supervised learning task, the classifier is supposed to select the most relevant pooling strategy for each cluster, in a discriminative manner.

We propose combining those two mid-level representations by using a kernel combination or by applying a late fusing strategy. The former can take advantage of choosing the appropriate kernel functions according to the mid-level representation, while the latter allows the use of a specific method of classification for each mid-level representation. For the kernel combination, we first compute the individual kernels: a linear kernel for Fisher Vector and a nonlinear Gauss-ℓ_2 kernel for BossaNova. Then, we apply a linear combination of those kernels as follows:

$$K = \varphi \cdot K_{BN} + (1 - \varphi) \cdot K_{FV}. \tag{4.5}$$

The weighting coefficient φ represents the relative importance given to the two mid-level representations. It can be fixed heuristically, or learned by cross-validation.

Our late fusion strategy is done by a linear combination (as the kernel fusion) of classification scores of the two mid-level representations.

As shown in the experiments (Chapter 5), we report that combining BossaNova with Fisher Vector indeed boosts the classification performances.

4.8 BossaNova as a Fisher Kernel Formalism

In order to further comment on the link with Fisher-based approaches, we present a generative formulation of our strategy. Indeed, we propose to derive our strategy as a Fisher Kernel on a generative model, called the (Fisher) BossaNova.

Let us consider the underlying distribution of the local features x as a mixture of several (basic) distribution functions $p_k(x)$:

$$p(x|\theta) = p_\theta(x) = \sum_{k=1}^{K} w_k p_k(x), \tag{4.6}$$

where $M = \{p_\theta(x) : \theta \in \Theta \subseteq \mathbb{R}^N\}$ is a parameterized function set, with $\sum_{k=1}^{K} w_k = 1$. For instance, if $p_k(x)$ are Gaussian functions, we recognize the classical Gaussian Mixture Model (GMM).

When considering a probability function $p_k(x)$ constant on a limited support of the function domain, the model is a multinomial law on the weights. If the parameters of this multinomial are fitted from the data used to learn the k-means quantizer, and are then simply given by the fraction of the local features assigned to each visual word, one can recognize the basic BoW strategy. The corresponding graphical representation is given in Figure 4.6 following notations of Krapac [2011].

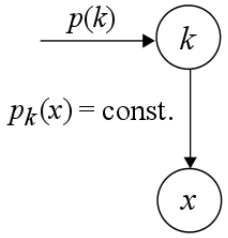


Figure 4.6: Graphical representation of the BoW model.

In BossaNova, we are considering not a constant but a slightly more complex density function for $p_k(x)$: a mixture of B constant non overlapping radial-based functions $p_b(x|k)$ between α_k^{min} and α_k^{max} to each visual word c_k. We have:

$$p_k(x) = \sum_{b=1}^{B} w_{(b,k)} p_b(x|k), \tag{4.7}$$

where $p_b(x|k)$ may be expressed with $\Delta_k = \frac{1}{B}(\alpha_k^{max} - \alpha_k^{min})$ and the indicator function $\mathbb{1}$ as:

$$p_b(x|k) = \mathbb{1}_{\alpha_k^{min} + (b-1)\Delta_k \leq ||x-c_k|| \leq \alpha_k^{min} + b\Delta_k}.$$

Let the normalization term to guarantee probabilities be contained in the weights $w_{(b,k)}$. Note that a straightforward extension of our strategy (not explored in this dissertation) comes from choosing overlapping supports. $p_k(x)$ acts as sum of ring functions in the feature space. Because our function is only dependent of the distance of a point to the visual word, it is no more a quantification problem of the feature space, at least in its classical formulation. Finally, by combining Equations 4.6 and

4.7, the generative model is:

$$p(x|\theta) = p_\theta(x) = \sum_{k=1}^{K} w_k \left(\sum_{b=1}^{B} w_{(b,k)} p_b(x|k) \right). \tag{4.8}$$

The resulting graphical models are given in Figure 4.7.

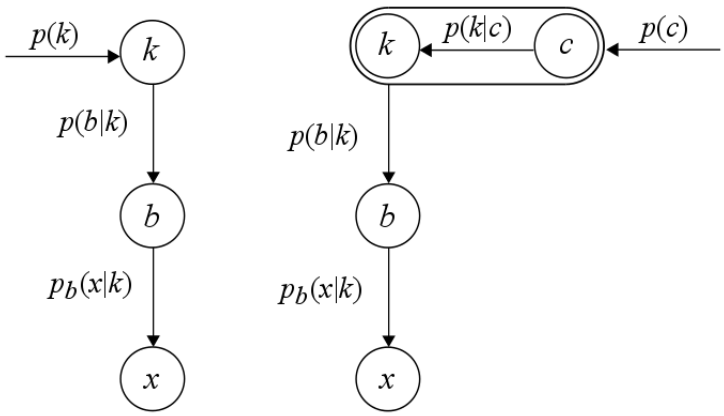


Figure 4.7: Graphical representation of our generative BossaNova model (left), and with Spatial Pyramid (right).

This kind of models $p_\theta(x)$ may be trained in different ways. Jaakkola and Haussler [1998] define a very poor model for DNA splice site classification problem that assigns the same probability to all examples, while Holub et al. [2005] create underlying generative models from categories on Caltech101 dataset. Alternatively, a fully unsupervised strategy using all images without any class label information may be used to learn the model parameters θ by maximizing the likelihood over the whole dataset of local descriptors. Perronnin and Dance [2007] have shown that there is no significant differences with supervised approaches.

Let θ^0 be the learnt model. To derive a kernel from our generative model, many marginalization kernels may be used [Tsuda et al., 2002]. Following the Fisher kernel strategy introduced in Chapter 2, the likelihood $\mathcal{L}_\theta(X)$ for one image noted $X = \{x_t, t = 1 \dots T\}$ (supposing the points generated independently) is given by $\mathcal{L}_\theta(X) = \prod_{t=1}^{T} p_\theta(x_t)$. The gradient with respect to the parameters θ is:

$$g(\theta, X) = \left(\partial \frac{1}{T} log \mathcal{L}_\theta(X) / \partial \theta_i \right)_{i=1}^{N}.$$

The Fisher score of X with respect to the learnt model is $g(\theta^0, X)$.

To compute g, we parameterize the multinomial laws using softmax:

$$w_k = exp(\alpha_k) / \sum_j exp(\alpha_j),$$

and

$$w_{(b,k)} = exp(\beta_{(b,k)}) / \sum_j exp(\beta_{(j,k)}).$$

We note $\gamma_i(x_t) = p_i(x_t)w_i/p_\theta(x_t)$ the occupancy probability, which represents the probability that any observation x_t has been generated by i-th mixture term, and $\gamma_{(b,k)}(x_t)$ the probability that x_t has been generated by the b-th ring related to the k-th visual word in the image.

The resulting scores are given below (a detailed derivation is given in Appendix A):

$$g(\alpha_k, X) = \frac{1}{T} \sum_{t=1}^{T} \gamma_k(x_t) - w_k, \tag{4.9}$$

$$g(\beta_{(b,k)}, X) = \frac{1}{T} \sum_{t=1}^{T} \left(\gamma_{(b,k)}(x_t) - w_{(b,k)}\right) \gamma_k(x_t). \tag{4.10}$$

The Fisher kernel is the dot product $\kappa(X, Z) = g(\theta^0, X)' I_M^{-1} g(\theta^0, Z)$ weighted by the inverse of the Fisher information matrix I_M^{-1} with respect to the setting θ^0, but other kernels may be considered: often the simple dot product $\kappa(X, Z) = g(\theta^0, X)' g(\theta^0, Z)$, called the "practical" Fisher kernel is employed [Shawe-Taylor and Cristianini, 2004]. When using a Gauss-ℓ_2 kernel in the Fisher score space, we have: $\kappa(X, Z) = exp\left(-||g(\theta^0, X) - g(\theta^0, Z)||^2 / 2\sigma^2\right)$.

Finally, the Fisher score $g(\theta^0, X)$ is easy to compute for the (Fisher) BossaNova model and much more compact than many other Fisher-based representations provided that b is small. In Equations 4.9 and 4.10, we see that this expression is close to the formulation of the BossaNova. The contribution of each x_t to the final vector score is very similar to BossaNova because usually $p_i(x_t) \approx 0$ and $\gamma_i(x_t) \approx 0$ if the cluster i is not in the K-nearest neighbors of x_t (as the localized soft assignment of BossaNova in Section 4.4). The main difference is that it is no more counting the number of points in a ring but the difference between this number and the "mean" number $w^0_{(b,k)}$ (or w^0_i) estimated over the image dataset. Note that some kernels (as the Gauss-ℓ_2) are invariant to global shift. Even if some preliminary tests did not indicate significant performance difference with the original BossaNova, further investigation

on this formulation and comparison with BN could reveal deeper relationships.

In contrast with Perronnin and Dance's, whose GMM Fisher Kernels are not designed with any pooling process in mind, in our proposition the pooling operation is central: our Fisher score is computed over a generative mixture model which represents the information obtained during the pooling step. Krapac et al. [2011], have also investigated alternatives to GMM generative model, but in the context of incorporating spatial information to the generative model. Our aim is different: enhancing the representation by building a mixture model less constrained by the Gaussianity hypotheses of GMM and more in touch with the needs of low-level feature representativeness.

4.9 Conclusion

In image classification, most of the highest-performing statistical learning approaches are based on the Bag-of-Words model. In this chapter, we proposed an extension of this formalism. Considering the Bag-of-Features, coding and pooling steps, we aim to advance the state-of-the-art by introducing a density function-based pooling strategy. Our hypothesis is that a well-chosen pooling strategy allows us to better represent the links between codewords and local descriptors in the image signature.

Our proposed BossaNova representation [Avila et al., 2012, 2013] is based on that novel pooling strategy, enhancing the Bag-of-Words model. The idea is to estimate the distribution of the descriptors around each codeword, by computing a histogram of distances between those descriptors and each codeword. The core of that idea has been introduced in our BOSSA representation [Avila et al., 2011], as a "proof-of-concept".

Therefore, in addition to that pooling strategy, BossaNova integrates three well-motivated computational steps over the BOSSA representation: the weighting scheme to balance the BoW term and the histogram of distances (BOSSA implicitly assigns equal importance to the both terms), the semi-soft coding scheme (BOSSA applies a hard coding) and the two-step normalization (BOSSA does not implement the power normalization and employs an ℓ_1-block normalization strategy instead of the ℓ_2).

Moreover, the BossaNova representation is interesting from a technical point of view: the simple vector computation, the ease of implementation and the relatively compact feature vector are non-negligible advantages, especially when tackling datasets which are becoming progressively larger in scale and scope. Also, BossaNova geometric properties lead us to predict an interesting complementarity with the Fisher Vector representations, which is confirmed empirically in the next chapter. That complementarity can also be understood in a generative model of BossaNova, since the density-based

model of our pooling "onion-rings" can be modeled conveniently in a likelihood model. That generative model can be employed in a Fisher kernel approach to BossaNova.

Chapter 5

Experimental Results

In this chapter, we present our empirical results. We choose five challenging benchmarks to perform our experiments: MIRFLICKR [Huiskes and Lew, 2008], ImageCLEF 2011/2012 Photo Annotation [Nowak et al., 2011; Thomee and Popescu, 2012], PASCAL VOC 2007 [Everingham et al., 2007] and 15-Scenes [Lazebnik et al., 2006].

After describing our experimental setup, we report the results of the BossaNova representation, the proposed in this dissertation. Those results are organized in three groups. First, we evaluate the impact of the three proposed improvements of BossaNova over BOSSA (Section 5.1), analyzing the isolated and joint impact of each enhancement on the BossaNova representation. Next, we explore the key aspects of the parametric space of our representation (Section 5.2). We then perform a comparison with state-of-the-art methods (Section 5.3), including both experiments with methods we have reimplemented ourselves, and published results reported in the literature. In order to make that comparison fair, we carefully follow the experimental protocol of each dataset. In what concerns the methods we reimplemented, we compare BossaNova [Avila et al., 2013] to BOSSA [Avila et al., 2011], but also to one of the best methods currently available, the Fisher Vectors [Perronnin et al., 2010c]. To provide a control baseline, we also employ the classical BoW. Finally, we present our participation in the ImageCLEF 2012 Photo Annotation task, in which we were ranked at the 2nd place out of 13 participants, considering only visual-based approaches (Section 5.4).

All experiments were conducted on a 64-bit Debian Linux machine powered by Intel® Xeon® CPU X5677 @ 3.47 GHz with 16 cores and 144 GB RAM. Despite the large computational power available, we do not require that power to process our experimental results. Our source code is written in C, C++ and Java.

Experimental Setup

The low-level feature extraction has a big influence on the quality of the results. If not controlled, it can easily become a nuisance factor in the experiments. Therefore, to make the comparisons fair, we use the same low-level descriptors for all techniques evaluated. We have extracted SIFT descriptors [Vedaldi and Fulkerson, 2010] on a dense spatial grid, with the step-size corresponding to half of the patch-size, over 8 scales separated by a factor of 1.2, and the smallest patch-size set to 16 pixels. That feature extraction process is also employed by Krapac et al. [2011].

As a result, roughly 8,000 local descriptors are extracted from each image of MIRFLICKR, ImageCLEF 2011/2012 Photo Annotation and PASCAL VOC 2007 datasets, and close to 2,000 local descriptors are extracted from each image of 15-Scenes dataset. The dimensionality of the SIFT is reduced from 128 to 64 by using principal component analysis (PCA). That setup for local descriptor extraction proves to give very good performances in standard image datasets, as reported in [Chatfield et al., 2011].

In order to learn the codebooks, we apply the k-means clustering algorithm with Euclidean distance over one million randomly sampled descriptors. For Fisher Vectors [Perronnin et al., 2010c], the descriptor distribution is modeled using a Gaussian mixture model (GMM), whose parameters (w, μ, Σ) are also trained over one million randomly sampled descriptors, using an expectation maximization algorithm. For all mid-level representations, we incorporate spatial information using the standard spatial pyramidal matching (SPM) scheme [Lazebnik et al., 2006]. In total, 8 spatial cells are extracted for MIRFLICKR, ImageCLEF 2011/2012 Photo Annotation and PASCAL VOC 2007, and 21 spatial cells for 15-Scenes.

One-versus-all classification is performed by support vector machine (SVM) classifiers. We use a linear SVM for Fisher Vectors, since it is well known that nonlinear kernels do not improve performances for those representations, see [Perronnin et al., 2010c]. For BoW [Sivic and Zisserman, 2003], BOSSA [Avila et al., 2011] and BossaNova [Avila et al., 2013], we use a nonlinear Gauss-ℓ_2 kernel. Kernel matrices are computed as $exp(-\gamma d(x, x'))$ with d being the distance and γ being set to the inverse of the pairwise mean distances.

Statistical significance tests for the differences between the means were performed using a Student t-test [Jain, 1991], paired over the dataset classes. The test consists of determining a confidence interval for the differences and simply checking if the interval includes zero, *i.e.*, if the confidence interval does not include zero, the difference is significant at that confidence level (see [Jain, 1991, chap. 13] for more details). Also, for the analysis of the improvements brought by each enhancement of BossaNova over

BOSSA, we have employed a factorial analysis of variance (ANOVA) [Jain, 1991, chap. 20], *i.e.*, a statistical procedure to analyze the significance of various factors (weighting scheme, localized-soft coding strategy and normalization). Those statistical tests are explored in the following section.

5.1 BOSSA to BossaNova Improvements Analysis

In this section, in order to quantify the performance gains of BossaNova over BOSSA, we propose to evaluate the individual performance increase brought out by each of the three proposed improvements: (i) learning the weighting scheme to balance the word-count (BoW) and the distances-histogram parts of the vectors, (ii) using a localized-soft coding strategy, and (iii) applying a new normalization to the final vector.

The joint activation of the three steps leads to eight different configurations where the performance of the corresponding mid-level representation is evaluated (denoted as **Weight**, **Soft** and **Norm** in Tables 5.1 and 5.2). Then, we apply a statistical t-test [Jain, 1991] to attest the significance of the difference between two given configurations. We perform the test for paired samples, *i.e.*, we evaluate the performance of two configurations on N different folds of train/test images and compute the difference between the performance metrics on each fold. The confidence interval (CI) for the average difference is computed using a Student-t model, and the difference is considered significative if the interval does not include zero (marked with ✓). For the tests in this section, we ask for a confidence of 95%.

Table 5.1 shows the evaluation of the eight different configurations on the 15-Scenes database, for $N = 30$ folds. We can see that the performances, measured by accuracy, monotonically increase from configuration **1** (BOSSA) to **8** (BossaNova). When only one improvement is added to BOSSA (configurations **2**, **3** and **4**), the performance gain is always significant. That already proves the relevance of the three modifications. When two improvements are incorporated, the performances increase are significant when compared to BOSSA (**1**), but also when compared to configurations with only one improvement: configurations **5**, **6** and **7** are all significantly better than the best configuration with one improvement (**4**). Adding the three improvements, the difference is again significant: **8** is better than **7**.

Testing just for the difference between BOSSA (**1**) and BossaNova (**8**) allows us to set the confidence to the large value of 99.9% and still obtain a CI that does not include zero, showing therefore that the difference is significant.

We apply the same setup on the PASCAL VOC database [Everingham et al.,

Table 5.1: Impact of the proposed improvements to the BossaNova on 15-Scenes [Lazebnik et al., 2006]. We use $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$. **Weight**: the weighted factor s, No = no cross-validation, Yes = cross-validation. **Soft**: soft assignment coding, No = hard assignment, Yes = localized soft assignment. **Norm**: normalization, No = ℓ_1 block normalization, Yes = power normalization + ℓ_2-normalization. The table shows the means and standard deviations of the 30 accuracy measures.

	Weight	**Soft**	**Norm**	**Accuracy**	**CI (95%)**
1	No	No	No	82.9 ± 0.5	
2	Yes	No	No	83.2 ± 0.2	$2 \leftrightarrow 1$ ✓
3	No	Yes	No	83.4 ± 0.5	$3 \leftrightarrow 1$ ✓
4	No	No	Yes	83.6 ± 0.1	$4 \leftrightarrow 1$ ✓
5	Yes	No	Yes	83.9 ± 0.1	$5 \leftrightarrow 1$ ✓, $5 \leftrightarrow 4$ ✓
6	Yes	Yes	No	84.5 ± 0.4	$6 \leftrightarrow 1$ ✓, $6 \leftrightarrow 4$ ✓
7	No	Yes	Yes	84.5 ± 0.4	$7 \leftrightarrow 1$ ✓, $7 \leftrightarrow 4$ ✓
8	Yes	Yes	Yes	85.3 ± 0.4	$8 \leftrightarrow 1$ ✓, $8 \leftrightarrow 7$ ✓

Table 5.2: Impact of the proposed improvements to the BossaNova on PASCAL VOC 2007 [Everingham et al., 2007]. We use $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$. **Weight**: the weighted factor s, No = no cross-validation, Yes = cross-validation. **Soft**: soft assignment coding, No = hard assignment, Yes = localized soft assignment. **Norm**: normalization, No = ℓ_1 block normalization, Yes = power normalization + ℓ_2-normalization. The table shows the means and standard deviations of the 10 mAP measures.

	Weight	**Soft**	**Norm**	**mAP**	**CI (95%)**
1	No	No	No	54.9 ± 0.5	
2	Yes	No	No	55.2 ± 0.4	$2 \leftrightarrow 1$ ✓
3	No	Yes	No	55.8 ± 0.5	$3 \leftrightarrow 1$ ✓
4	No	No	Yes	55.6 ± 0.4	$4 \leftrightarrow 1$ ✓
5	Yes	No	Yes	55.9 ± 0.4	$5 \leftrightarrow 1$ ✓, $5 \leftrightarrow 4$ ✓
6	Yes	Yes	No	56.4 ± 0.4	$6 \leftrightarrow 1$ ✓, $6 \leftrightarrow 4$ ✓
7	No	Yes	Yes	58.1 ± 0.4	$7 \leftrightarrow 1$ ✓, $7 \leftrightarrow 4$ ✓
8	Yes	Yes	Yes	58.8 ± 0.4	$8 \leftrightarrow 1$ ✓, $8 \leftrightarrow 7$ ✓

2007]. Here, the performance metric is the mAP, computed over the 20 classes for $N = 10$ folds[1]. The same conclusions apply: each improved configuration significantly outperforms its predecessor, as illustrated in Table 5.2.

Again, the difference between BOSSA (**1**) and BossaNova (**8**) is significant with a large confidence. For 99.9% confidence, the CI does not include the zero.

For both datasets, we have also tested the influence of the proposed improve-

[1]Note that in the VOC 2007 database, the train/val/test folds are fixed for evaluating performances. Here, we use random folds to obtain the necessary number of runs for statistical analysis.

ments using a factorial analysis of variance (ANOVA) [Jain, 1991]. In both cases, the models obtained were highly significant (with confidence above 99.9%) for all three improvements, confirming the results above. In addition, the ANOVA allows to measure the relative impact of each proposed improvement. For the more challenging PASCAL VOC dataset, the soft coding explains almost 48% of the BossaNova performance, while the two-step normalization explains about 31%. The weighting scheme, in isolation, is responsible for only 3% of the variation, but there is a cross-effect between the weighting and the soft coding that accounts for another 9%. The impact of the coding is clearly the largest, but the importance of the normalization is quite surprising, especially considering the optimization of that step is often neglected in the literature.

5.2 BossaNova Parameter Evaluation

The key parameters in BossaNova representation are the number of codewords M, the number of bins B in each histogram z_m, and the range of distances $[\alpha_m^{min}, \alpha_m^{max}]$ – the minimum distance α_m^{min} and the maximum distance α_m^{max} in the $\mathbb{R}^D$ descriptor space that define the bounds of the histogram.

The codebook size M has a similar meaning as in standard BoW approaches. Histogram size B defines the granularity to which $\text{pdf}(\alpha_{\mathbf{m}})$ is estimated. The choices of M and B are co-dependent, and $M \cdot B$ determines the compromise between accuracy and robustness. The smaller $M \cdot B$ is, the less the representation is accurate, the larger $M \cdot B$ is, the less confidence we have on the estimate of each bin of the histogram representing the underlying distribution. In addition, too large $M \cdot B$ values may lead to excessively sparse vector representations.

The bounds α_m^{min} and α_m^{max} define the range of distances for the histogram computation. Local descriptors outside those bounds are ignored. For α_m^{max}, the idea is to consider only descriptors that are "close enough" to the center, and to discard the remaining ones. For α_m^{min}, the idea is to avoid the empty regions that appear around each codeword, in order to avoid wasting space in the final descriptor.

In BossaNova, α_m^{min} and α_m^{max} are set up differently for each codeword $\mathbf{c}_m$. Since our codebook is created using k-means, we take advantage of the knowledge about the "size" of the clusters, given by the standard deviations σ_m. We set up the bounds as $\alpha_m^{min} = \lambda_{min} \cdot \sigma_m$ and $\alpha_m^{max} = \lambda_{max} \cdot \sigma_m$, as shown in Figure 5.1.

To provide more comprehensive analysis of our representation, we evaluate its behavior as three key parameters change: the codebook size M (Section 5.2.1), the number of bins B (Section 5.2.2) and the minimum distance α_m^{min} (Section 5.2.3).

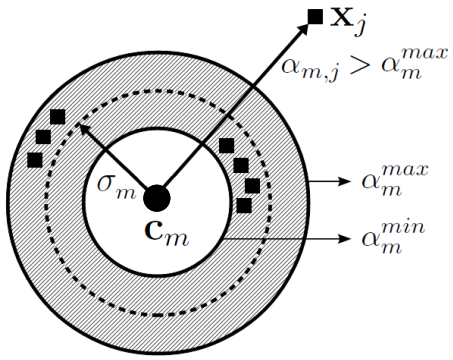


Figure 5.1: Illustration of the range of distances $[\alpha_m^{min}, \alpha_m^{max}]$ which defines the bounds of the histogram. The hatched area corresponds to the bounds. Local descriptors outside those bounds are ignored.

We report the results using the MIRFLICKR [Huiskes and Lew, 2008], but as our experiments already suggested, the conclusions can be generalized to the other datasets.

5.2.1 Codebook Size

The impact of codebook size M on BossaNova classification performance is shown on Table 5.3, which clearly shows that larger codebooks lead to higher accuracy. BoW performance, however, stops growing at 4096 visual words.

As stated in Section 5.3, the performances reported in Table 5.6 correspond to a BossaNova with good parameters, but not strongly fine-tuned. Therefore, our representation can reach an even higher score of 55.2% with a dictionary of size $M = 8192$. However, the last improvement from 4096 to 8192 is not that high, suggesting that the growth will soon stop. Meanwhile, the representation has doubled in size. Hence, we define as our standard setting $M = 4096$ in order to get a good tradeoff between effectiveness and efficiency.

Table 5.3: Codebook size impact on BossaNova (BN) and BoW performance (mAP (%)) on MIRFLICKR [Huiskes and Lew, 2008]. BN: $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$.

	Codebook size			
	1024	2048	4096	8192
BN [Avila et al., 2013]	51.8	52.9	54.4	55.2
BoW [Sivic and Zisserman, 2003]	50.3	51.3	51.5	51.1

Comparison with Hierarchical BoW

We contrast BossaNova to a Hierarchical BoW (H-BoW) since there are some similarities between our pooling approach and a 2-step descriptor space clustering. The pooling performed in BossaNova can indeed be regarded as a special form of clustering, where the second-level of clustering corresponds to regions that are equally spaced from the center. On the other hand, in a standard H-BoW, the second-level clusters are similar to the first-level ones (*e.g.*, hyper-sphere, if ℓ_2 norm is used for clustering).

We claim that the special shape of the second-level clustering, which is based on the idea of pooling descriptors depending on their similarity to the center, is better founded that a naive 2-level clustering (with Euclidean distance).

To achieve that comparison, we build a 2-level hierarchical codebook using BossaNova codebook size (M) at the first-level, and BossaNova histograms bin count plus one ($B + 1$) at the second-level. That makes the comparison fair, allocating the same size for both representations. For instance, BossaNova with a codebook of size $M = 4096$ and two bins per histogram ($B = 2$), will be compared with a H-BoW first-level of 4096 and second-level of 3 clusters (both representation are therefore of size $4096 \times 3 \times 8$, 8 being the spatial cells of the SPM scheme).

Table 5.4 compares BossaNova with H-BoW on the MIRFLICKR dataset. For each codebook size, we observe that BossaNova is superior to H-BoW, and that the difference tends to grow as the (first-level) codebook size grows. That confirms the relevance of the improved pooling scheme introduced in the dissertation.

Table 5.4: Comparison of BossaNova (BN) and Hierarchical BoW performance (mAP (%)) on MIRFLICKR [Huiskes and Lew, 2008]. BN: $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$.

	Codebook size		
	1024	2048	4096
BN [Avila et al., 2013]	51.8	52.9	54.4
Hierarchical BoW	50.6	51.3	51.4

5.2.2 Bin quantization

We next investigate how BossaNova classification performance is affected by the number of bins (B). Using $M = 4096$, the number of bins is varied among 2, 4 and 6. The results of our experiments are shown in Table 5.5.

Table 5.5: Bin quantization influence on BossaNova (BN) mAP (%) performances on MIRFLICKR [Huiskes and Lew, 2008]. BN: $M = 4096$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$.

	Number of Bins		
	$B = 2$	$B = 4$	$B = 6$
mAP	54.4	54.7	**54.9**

First, we observe that increasing the number of bins yields a slight amelioration in performance. However, the growth depends on the topic of MIRFLICKR dataset: for 30 out of 38 concepts the performance increases up to 1.9% and for 3 isolated concepts ($bird(r)$, $car(r)$, $sea(r)$) the performance decreases slightly, by 0.2%.

Once again, further investigations will certainly provide optimized parameters but with a higher complexity. We handled default parameters to 2 here in order to get compact representations.

5.2.3 Minimum Distance α_m^{min}

The fact that descriptors seldom, if ever, fall close to the codewords is a counter-intuitive consequence of the geometry of high-dimensional spaces. Figure 5.2 illustrates the phenomenon, displaying the average density of SIFT descriptors on the neighborhood of codewords, in MIRFLICKR dataset [Huiskes and Lew, 2008].

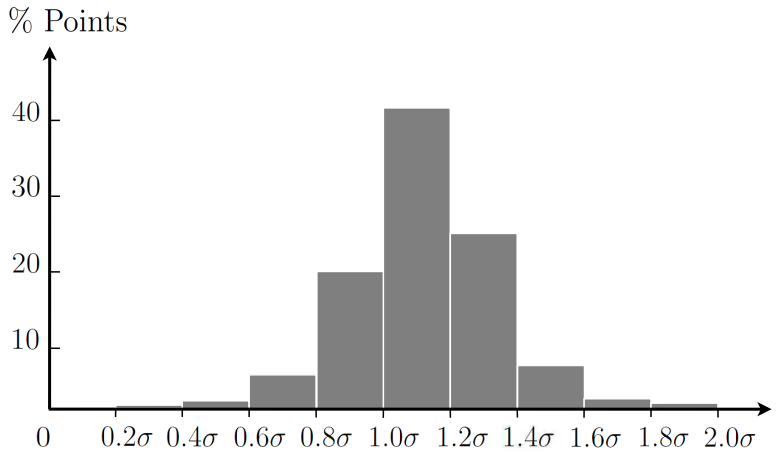


Figure 5.2: Average density of SIFT descriptors in the neighborhood of codewords in MIRFLICKR dataset [Huiskes and Lew, 2008], showing that descriptors seldom, if ever, are closer than a certain threshold to the codewords. That counter-intuitive phenomenon is a consequence of the "curse of dimensionality" [Bellman, 1961].

Note that the parameters may act jointly to the locality constraints defined in Section 4.4: a descriptor $\mathbf{x}_j$ that is the k-NN from a center $\mathbf{c}_m$ is not considered for generating the signature if $d_2(\mathbf{x}_j, \mathbf{c}_m) > \alpha_m^{max}$.

Therefore, we study the effects of the minimum distance α_m^{min} on BossaNova classification performance. Using the test values of BossaNova parameters (*i.e.*, $B = 2$, $M = 4096$, $\lambda_{max} = 2$), we set λ_{min} based on Figure 5.2.

For $\lambda_{min} = 0.4$ and $\lambda_{max} = 2$, corresponding to 95% of the total SIFT descriptors on the whole dataset, we obtain a mAP = 54.9% which is slightly better than the range of $\lambda_{min} = 0$ and $\lambda_{max} = 2$ (mAP = 54.4%, see Table 5.6). That is in accordance with our intuition.

Interestingly, we observe considerable improvements for the most of the concepts (up to 1%) and also a decrease for some ones (up to 0.5%). That suggests that setting a λ_{min} and even λ_{max} per codeword seems to be useful to exploit as future research.

5.3 Comparison of State-of-the-Art Methods

We compare BossaNova to other representations, perform our own re-implementation of those techniques. The methods chosen were:

- BossaNova (BN) [Avila et al., 2013], the method proposed in this dissertation.
- BOSSA [Avila et al., 2011], which can be regarded as a proof-of-concept of proposed pooling. Also, BOSSA is chosen to validate our BossaNova improvements.
- Fisher Vectors (FV) [Perronnin et al., 2010c], one of the best mid-level representations currently reported in the literature [Chatfield et al., 2011].
- The kernel combination BN + FV, chosen to evaluate the methods' complementarity[2].
- Bag-of-Words (BoW) [Sivic and Zisserman, 2003]. A classical histogram of codewords, obtained with hard quantization coding and average-pooling; it constitutes a control baseline for the other methods.

The "overall picture" from the comparison of state-of-the-art methods we have implemented ourselves can be summarized as follows. All recent methods improve the classification performance over the BoW baseline. Consequently, that illustrates the relevance of improving the pooling scheme introduced in this dissertation. Also,

[2]It is explicitly shown in the text when we apply the late fusion strategy.

we observe a considerable improvement of performance from BOSSA to BN, showing the benefits brought out by the weight factor, soft coding and new normalization. Furthermore, the combination of BN and Fisher Vector representations (obtained by a kernel fusion) outperforms both individual methods, which corresponds to a remarkable success of the complementariness of BN and FV representations. Besides, with at least 99% confidence, all differences are significant for those methods. Results published in the literature, unfortunately, do not include significance tests or confidence intervals.

We also report the best results available for each dataset. That allows us to evaluate other recent methods that build upon the standard baseline BoW, *e.g.*, methods using sparse coding and max pooling [Yang et al., 2009b; Boureau et al., 2010a].

It is important to note that, although we have chosen for BossaNova parameters we believed were good, in the interest of a fair comparison, we have not fine-tuned it for each dataset. Therefore, the numbers reported do not represent the limit of the performance achievable by the method (in a few cases higher results are achieved in this dissertation in Section 5.2, where we explore the parameters more thoroughly).

Moreover, two essential aspects should be kept in mind when interpreting the results of this section. The first is that many methods nowadays work by exploiting complex schemes, often involving dozens of different features and classifiers. Since our aim here is to isolate the performance of the mid-level representation component, we opt for a single-descriptor approach (using SIFT), and emphasize our comparison to baselines that only employ single-descriptor schemes.

The second one is that the impact of the low-level feature step (density, fine-tune parametrization) is nonneglectable, but currently very little understood. We have cooperated with the authors to bring the reported numbers to the best agreement possible, it was not our aim here to optimize the low-level extraction phase, neither for our method, nor for theirs. It is also important to notice that, although we have not optimized the low-level feature step, we consider a dense sampling strategy, which gives better results. Additionally, our results can be further improved by using high density sampling of local descriptors (typically, denser sampling yields higher performance [Chatfield et al., 2011]).

5.3.1 Results for MIRFLICKR

Table 5.6 shows the results over MIRFLICKR, and details the parameter settings for each method. We can notice that the BOSSA representation, our proof-of-concept, outperforms BoW with 1.2% absolute improvement (2.3% relative improvement). Comparing the BOSSA to the BN, our proposed representation, we observe an increase

Table 5.6: Image classification mAP (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on MIRFLICKR [Huiskes and Lew, 2008]. BOSSA: $M = 2048$, $B = 6$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, as in [Avila et al., 2011]; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013], BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c].

	mAP (%)
Our methods	
BOSSA [Avila et al., 2011]	52.7
BN [Avila et al., 2013]	54.4
BN + FV [Avila et al., 2013]	56.0
Implemented methods	
BoW [Sivic and Zisserman, 2003]	51.5
FV [Perronnin et al., 2010c]	54.3
Published results	
[Huiskes et al., 2010]	37.5
[Guillaumin et al., 2010]	53.0

from 52.7% to 54.4% (an absolute improvement of 1.7%).

Furthermore, BN is tied with FV, the current state-of-the-art method. Note that our representation (12,288 dimensions for each spatial cell) is about three times smaller than FV (32,768 dimensions for each spatial cell). Also, we observe that our method is better than FV for 22 out of 38 concepts. Additionally, unlike the overall picture, at 99% confidence the difference is not significant for BN and FV.

Finally, we can notice the considerable improvement obtained when combining BN and FV, reaching a mAP of 56.0%. The combination surpasses both individual methods for 31 out of 38 concepts while performing similarly for the seven remaining concepts. Table 5.7 shows the results of each concept over MIRFLICKR.

From the literature, we choose the baseline dataset result [Huiskes et al., 2010], and the best, as far as we know, result published [Guillaumin et al., 2010]. The baseline performances [Huiskes et al., 2010] are quite low, 14% below our re-implementation of the classical BoW. The main reason is the features employed there, global descriptors, which are much outperformed by highly discriminant local descriptors such as SIFT.

In comparison to Guillaumin et al.[3], BN performs better for 29 out of 38 concepts, and its mAP increases from 53.0% to 56.0%. It is notable BN employs only SIFT to build the mid-level representation, while Guillaumin et al. combines 15 different image representations, including SIFT.

[3]The authors also consider as features the image tags. Here, we show their results which use only the visual image content as features.

Table 5.7: Image classification AP and mAP (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on MIRFLICKR dataset [Huiskes and Lew, 2008]. BOSSA: $M = 2048$, $B = 6$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, as in [Avila et al., 2011]; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013], BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c]; MIR'10 [Huiskes et al., 2010]; CVPR'10 [Guillaumin et al., 2010].

	Our methods			**Impl. methods**		**Published results**	
	BOSSA	BN	BN + FV	BoW	FV	MIR'10	CVPR'10
1: animals	48.2	49.2	49.6	45.1	47.1	27.8	48.7
2: baby	14.9	16.5	16.4	14.0	14.1	8.4	17.0
3: baby(r)	19.6	21.4	22.6	18.7	20.5	8.8	21.4
4: bird	17.8	20.1	22.3	16.7	20.3	12.8	22.7
5: bird(r)	23.9	25.5	27.9	22.5	24.3	12.9	29.3
6: car	40.2	42.3	44.9	38.8	43.6	17.9	37.5
7: car(r)	55.6	57.7	62.6	52.3	60.8	22.7	52.2
8: clouds	83.7	85.6	86.0	82.2	84.2	65.1	82.5
9: clouds(r)	76.4	78.4	80.1	75.7	80.3	51.1	75.5
10: dog	32.2	33.2	36.2	32.0	31.5	15.5	32.3
11: dog(r)	35.8	36.8	40.5	35.6	35.7	15.6	36.7
12: female	60.2	61.8	65.5	60.0	62.5	46.1	57.5
13: female(r)	58.4	60.3	60.9	56.8	59.2	38.9	54.9
14: flower	46.3	47.5	50.8	44.3	49.2	46.9	53.6
15: flower(r)	59.4	61.8	66.3	58.1	66.1	51.9	54.9
16: food	44.8	45.1	46.6	44.4	44.1	29.3	50.1
17: indoor	73.9	75.7	75.9	71.6	74.2	60.5	74.5
18: lake	32.6	33.6	35.9	32.4	34.6	18.8	31.3
19: male	53.9	55.9	56.4	53.0	55.3	40.7	51.7
20: male(r)	46.2	49.9	50.2	45.9	47.0	29.4	45.0
21: night	62.9	64.2	64.6	61.1	63.7	55.4	64.9
22: night(r)	47.5	50.9	50.6	46.8	49.5	39.0	55.8
23: people	81.6	83.3	83.2	80.1	81.7	63.1	78.9
24: people(r)	78.8	81.1	81.1	77.0	79.1	55.8	75.1
25: plant life	76.2	77.8	78.1	76.1	78.3	68.7	78.5
26: portrait	72.3	74.8	75.7	70.4	74.1	49.3	68.1
27: portrait(r)	72.6	74.9	75.8	70.2	74.1	49.3	68.2
28: river	28.6	29.7	33.3	28.2	32.2	17.9	26.5
29: river(r)	7.9	8.0	11.8	7.4	10.3	10.2	8.1
30: sea	54.4	57.0	59.4	53.3	58.4	36.6	57.1
31: sea(r)	34.2	36.2	36.6	33.5	33.4	12.6	33.4
32: sky	87.3	88.7	88.9	86.7	88.1	77.5	86.6
33: structures	80.1	82.1	82.6	79.9	81.6	62.6	77.4
34: sunset	53.7	54.4	55.9	53.6	54.6	58.8	66.5
35: transport	48.4	50.3	51.3	46.8	49.2	29.8	46.4
36: tree	70.9	71.7	73.3	70.7	73.8	51.4	67.1
37: tree(r)	59.4	61.4	64.1	57.1	61.5	20.5	54.8
38: water	61.5	63.5	65.8	58.7	64.6	44.8	62.2
mAP	52.7	54.4	56.0	51.5	54.3	37.5	53.0

To the best of our knowledge, ours is the best result reported to date on MIR-FLICKR dataset, using only visual features.

5.3.2 Results for ImageCLEF 2011 Photo Annotation

Table 5.8 gives the results, both the ones implemented and tested by us, and the ones reported on literature. We note an absolute improvement of 1.7% from BoW to BOSSA, highlighting the relevance of our pooling scheme. We also observe a considerable improvement of performance from BOSSA to BN, from 32.9% to 35.3% (a 2.4% absolute improvement). Furthermore, the combination of BN and Fisher Vector representations outperforms the other methods by up to 3.1%.

We also compare our results with those of the five best systems reported in the literature. In the ImageCLEF 2011 Photo Annotation task, each group registered for the challenge is restricted to a maximum of five runs. Table 5.8 shows the best run for each group, with the restriction to results that employed only the visual information. We also show the results of each concept for the (re)implemented methods and the two best systems (see Table 5.9).

The best system during the competition ([Binder et al., 2011]) reported 38.8%

Table 5.8: Image classification mAP (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on ImageCLEF 2011 Photo Annotation task [Nowak et al., 2011]. BOSSA: $M = 2048$, $B = 6$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, as in [Avila et al., 2011]; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013], BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c].

	mAP (%)
Our methods	
BOSSA [Avila et al., 2011]	32.9
BN [Avila et al., 2013]	35.3
BN + FV [Avila et al., 2013]	38.4
Implemented methods	
BoW [Sivic and Zisserman, 2003]	31.2
FV [Perronnin et al., 2010c]	36.8
Published results	
[Mbanya et al., 2011]	33.5
[Le and Satoh, 2011]	33.7
[van de Sande and Snoek, 2011]	36.7
[Su and Jurie, 2011]	38.2
[Binder et al., 2011]	38.8

Table 5.9: Image classification AP and mAP (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on ImageCLEF 2011 Photo Annotation task [Nowak et al., 2011]. BOSSA: $M = 2048$, $B = 6$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, as in [Avila et al., 2011]; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013], BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c]; Top1 [Binder et al., 2011]; Top2 [Su and Jurie, 2011].

	Our methods			**Impl. methods**		**Published results**	
	BOSSA	BN	BN + FV	BoW	FV	Top2	Top1
1: party life	25.2	29.3	31.2	20.0	30.5	33.0	32.4
2: family friends	49.6	52.7	55.6	47.6	54.0	54.8	54.6
3: beach holidays	33.4	40.9	46.2	32.4	44.9	43.0	49.1
4: buildings sights	57.8	58.8	63.8	53.8	61.2	61.4	60.4
5: snow	13.8	16.0	19.2	13.4	16.8	24.5	25.3
6: city life	51.0	52.9	56.5	50.4	55.0	56.7	54.9
7: landscape nature	80.5	81.9	82.2	79.0	81.5	78.2	80.5
8: sports	12.3	13.7	16.4	9.8	15.8	14.7	17.1
9: desert	5.8	15.6	19.3	3.6	17.6	22.6	16.2
10: spring	10.8	13.1	15.4	7.8	14.5	19.1	21.9
11: summer	23.0	23.0	24.7	21.8	23.2	31.7	32.3
12: autumn	19.0	19.2	22.2	18.0	21.3	32.8	37.6
13: winter	15.9	21.3	27.9	15.5	26.0	25.6	30.0
14: indoor	56.4	58.9	62.0	55.9	59.1	61.1	62.9
15: outdoor	88.5	88.3	90.5	88.5	88.7	86.4	90.1
16: plants	71.3	73.4	75.3	70.1	74.8	77.2	79.8
17: flowers	33.2	41.7	45.3	32.2	43.2	51.1	52.8
18: trees	63.5	64.5	69.3	61.5	68.5	66.6	68.0
19: sky	84.9	85.8	89.6	83.7	87.5	85.6	89.2
20: clouds	83.2	83.5	89.2	82.2	83.7	80.8	84.7
21: water	61.4	65.5	67.7	59.5	66.0	63.1	67.6
22: lake	28.6	34.1	37.6	25.4	36.0	32.2	33.9
23: river	23.1	23.1	28.3	22.1	27.0	28.5	27.2
24: sea	48.2	52.7	56.2	45.8	55.4	51.4	52.5
25: mountains	47.7	53.9	57.2	44.1	56.3	54.6	56.4
26: day	84.9	84.7	85.9	84.6	85.3	84.7	87.5
27: night	49.6	55.3	49.6	47.5	56.0	58.6	59.2
28: sunny	38.1	38.6	42.3	36.1	40.0	50.5	51.8
29: sunset sunrise	59.2	60.9	64.1	56.8	62.9	74.7	80.2
30: still life	35.3	39.3	40.3	31.3	39.8	42.9	41.3
31: macro	45.5	48.6	50.5	44.5	49.8	52.8	51.2
32: portrait	63.5	67.9	69.9	61.7	68.3	68.3	67.7
33: overexposed	12.8	17.2	18.7	11.6	18.3	20.6	24.1
34: underexposed	24.2	28.3	30.2	23.2	30.1	34.5	32.9
35: neutral illumination	95.4	96.4	98.2	94.5	97.7	98.0	98.3
36: motion blur	24.6	28.9	30.3	22.5	29.4	29.7	25.7
37: out of focus	23.2	23.3	26.1	22.8	24.8	27.8	24.3
38: partly blurred	76.5	77.3	80.4	76.0	79.1	72.9	74.5
39: no blur	92.0	92.2	93.6	91.7	92.4	91.0	90.7
40: single person	52.4	56.2	58.4	50.3	55.7	58.8	57.8
41: small group	29.1	29.9	35.0	27.3	33.0	35.8	38.8
42: big group	40.2	44.9	49.2	34.7	48.5	45.0	45.7
43: no persons	89.0	89.8	92.0	88.9	90.3	89.8	91.9
44: animals	44.1	48.9	54.4	43.2	53.3	56.1	52.6
45: food	45.2	47.1	50.5	44.6	48.5	56.6	54.9

continued on next page

	Our methods			Impl. methods		Published results	
	BOSSA	BN	BN + FV	BoW	FV	Top2	Top1
46: vehicle	45.9	49.6	52.3	44.7	50.9	51.0	49.9
47: aesthetic impression	27.0	27.0	27.7	26.1	27.4	32.0	31.1
48: overall quality	20.0	22.6	26.0	20.0	20.2	22.9	28.8
49: fancy	15.9	17.3	19.9	15.3	18.0	22.7	24.8
50: architecture	30.5	33.2	36.5	29.5	34.0	34.0	35.4
51: street	33.0	34.9	37.2	31.8	36.4	37.7	39.0
52: church	18.2	19.9	22.4	20.3	24.9	14.2	18.0
53: bridge	8.3	9.3	10.5	9.6	12.3	10.9	13.0
54: park garden	32.0	37.0	41.5	40.2	43.0	47.8	45.9
55: rain	0.5	3.3	6.1	5.9	7.3	6.2	1.0
56: toy	17.8	18.9	23.5	20.9	25.5	27.7	28.5
57: musical instrument	3.2	5.4	8.0	7.7	9.4	8.8	7.0
58: shadow	9.4	10.4	11.4	10.4	16.4	14.9	19.5
59: body part	18.1	21.9	26.3	24.8	28.9	27.8	30.1
60: travel	11.1	12.3	19.1	17.4	21.3	14.4	20.8
61: work	3.7	4.3	7.9	6.6	8.3	13.2	5.6
62: birthday	0.8	0.9	1.2	1.0	1.4	1.0	0.9
63: visual arts	32.6	33.2	35.0	34.3	39.3	33.4	38.5
64: graffiti	4.8	5.0	5.4	5.2	7.1	8.8	3.0
65: painting	13.9	16.9	20.7	19.7	23.9	24.7	24.5
66: artificial	11.7	12.7	11.9	12.7	14.7	12.6	14.5
67: natural	68.0	69.2	71.9	70.3	75.3	72.6	73.8
68: technical	5.9	6.7	9.6	7.7	10.5	6.5	7.7
69: abstract	1.8	1.9	2.0	2.3	2.4	2.1	3.4
70: boring	7.5	7.9	8.1	8.3	9.9	9.2	9.5
71: cute	55.5	57.5	61.7	59.9	63.5	62.2	62.7
72: dog	29.0	32.1	38.6	36.9	40.4	41.7	38.4
73: cat	12.4	20.4	27.0	22.7	26.4	17.8	19.8
74: bird	19.0	24.2	30.6	29.3	33.9	27.9	27.5
75: horse	5.7	6.3	10.8	9.2	12.4	9.4	12.9
76: fish	1.3	2.3	2.8	2.7	3.6	2.4	4.1
77: insect	15.4	16.8	20.5	19.1	22.9	24.1	22.9
78: car	30.9	31.9	41.5	39.8	44.2	40.1	39.3
79: bicycle	16.0	17.5	31.7	28.4	33.5	32.4	32.0
80: ship	10.5	14.8	18.9	16.9	20.5	12.9	12.0
81: train	16.3	17.3	20.2	20.6	22.4	2.8	19.2
82: airplane	9.9	15.8	21.4	15.8	22.8	23.2	16.6
83: skateboard	0.1	0.1	0.2	0.1	0.3	0.2	0.6
84: female	42.0	43.0	51.0	50.2	54.6	51.5	49.8
85: male	18.4	20.2	26.4	24.1	27.4	22.1	21.7
86: baby	12.5	13.1	18.1	16.4	20.3	24.1	25.5
87: child	8.2	9.3	16.6	14.3	17.3	18.2	20.1
88: teenager	18.8	20.5	26.7	25.3	26.5	27.1	27.6
89: adult	47.9	48.7	54.3	52.9	56.9	57.3	56.3
90: old person	4.9	5.7	6.5	5.7	6.8	8.4	10.5
91: happy	35.2	38.1	43.3	42.7	46.2	44.3	43.3
92: funny	29.8	30.3	32.3	30.3	34.9	36.8	35.9
93: euphoric	4.7	5.9	7.1	6.2	7.9	8.8	13.6
94: active	25.8	27.0	32.9	29.7	33.4	35.2	36.0
95: scary	11.6	12.6	14.1	13.7	15.4	20.3	20.0
96: unpleasant	18.0	18.5	21.3	18.9	22.8	26.5	25.8
97: melancholic	23.1	23.8	24.8	22.9	29.8	34.3	35.7
98: inactive	44.8	45.6	49.7	45.3	53.3	54.1	55.8
99: calm	48.6	50.9	52.9	50.2	56.4	56.5	57.1
mAP	32.9	35.3	38.4	31.2	36.8	38.2	38.8

mAP, employing nonsparse multiple kernel learning and multi-task learning. They apply SIFT and color channel combinations to build different extensions of the BoW models with respect to sampling strategies and BoW mappings. The system of Su and Jurie [2011] uses many features, such as SIFT, HoG, Texton, Lab–1948, SSIM, and Canny, aggregating them by a BoW into a global histogram. Fisher Vectors and contextual information were used as enhancement of the BoW models. The method of van de Sande and Snoek [2011] employs several color SIFT features with Harris-Laplace and dense sampling, and apply the SVM classifier. The system of Le and Satoh [2011] also use numerous features. As global features, they use color moments, color histogram, edge orientation histogram and local binary patterns; and as local features, keypoint detectors such as Harris Laplace, Hessian Laplace, Harris Affine, and dense sampling are used to extract SIFT descriptors. Again, classification is performed with a SVM classifier. The approach of Mbanya et al. [2011] is based on the BoW model. They apply feature fusion of the opponent SIFT descriptor and the GIST descriptor. Moreover, a post-classification processing step is incorporated in order to refine classification results based on rules of inference and exclusion between concepts. As we can notice, all those top-performing systems employ complex combinations of several low-level features to achieve their good results.

In view of that, our results of 35.3% for BN, and 38.4% for BN + FV (the latter practically tied with the best reported results) are remarkably good, since we employ just SIFT descriptors. Moreover, the performance our method can be further improved by feature combination expansions [Picard et al., 2010, 2012].

5.3.3 Results for PASCAL VOC 2007

Table 5.10 shows the results, detailing the parameter settings for each method. Again, we observe that the BOSSA representation, our proof-of-concept, outperforms BoW with 1.2% absolute improvement. Also, we achieve a considerable improvement of performance from BOSSA to BN, from 54.4% to 58.5% (an absolute improvement of 4.1%). Furthermore, the combination BN + FV outperforms the previous methods. For some categories its improvement in mAP reached up to 10%, especially challenging ones (*e.g.*, *bottle*, *cow*). Additionally, our late fusion strategy reaches a performance of 62.4%. Table 5.11 shows the results of each visual object class for the (re)implemented methods and the available published methods.

Table 5.10 also shows the comparison with published results. The comparison with Krapac et al. [2011] is particularly relevant, because we employ the same low-level descriptor extraction as them, although our representation ends up being more

Table 5.10: Image classification mAP (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on PASCAL VOC 2007 dataset [Everingham et al., 2007]. BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013]; BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c].

	mAP (%)
Our methods	
BOSSA [Avila et al., 2011]	54.4
BN [Avila et al., 2013]	58.5
BN + FV [Avila et al., 2013]	61.6
Late Fusion (BN + FV)	62.4
Implemented methods	
BoW [Sivic and Zisserman, 2003]	53.2
FV [Perronnin et al., 2010c]	59.5
Published results	
[Krapac et al., 2011]	56.7
[Wang et al., 2010]	59.3
[Chatfield et al., 2011]	61.7
[Sánchez et al., 2012]	66.3

compact. The LLC method of Wang et al. [2010] is evaluated with HOG descriptors. LLC was also evaluated on extremely dense SIFT descriptors (sampling step of 3 pixels at four scales), roughly 70,000 per image, obtaining a mAP of 53.8% with a codebook of 4,000 words [Chatfield et al., 2011].

Zhou et al. [2010] published a score of 64.0% using Super-Vector (SV) coding, but Chatfield et al. showed that the best reproducible result for SV coding is 58.2%[4]. Moreover, Chatfield et al. achieved 61.7% for Fisher Vector. Those results are encouraging, since the SIFT descriptors employed on those experiments are extremely dense. By using SIFT features nearly 10 times less dense, our result of 62.4% surpasses the result reported by Chatfield et al. for FV.

The best published result, using only SIFT descriptors as low-level features, is 66.3% for a system based on a late fusion approach [Sánchez et al., 2012], which averages the outputs of the classifiers from a (i) FV system based on the combination of augmented low-level features and an objectness measure to estimate the location of objects in images and (ii) a spatial pyramids system. The authors also employed dense SIFT, whose dimensionality are reduced to 80 by PCA. Furthermore, FV are extracted

[4]The difference results from nontrivial optimizations not described in their paper, making it extremely hard to reproduce.

Table 5.11: Image classification AP and mAP (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on PASCAL VOC 2007 dataset [Everingham et al., 2007]. BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013]; BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c]; BMVC'11 [Chatfield et al., 2011]; PRL' 12 [Sánchez et al., 2012].

	Our methods				**Impl. methods**		**Published results**	
	BOSSA	BN	BN + FV	LF	BoW	FV	BMVC'11	PRL'12
1: aeroplane	75.9	79.5	82.1	82.8	74.5	80.5	79.0	83.8
2: bicycle	59.3	64.5	67.0	69.2	57.3	64.9	67.4	72.0
3: bird	43.7	49.8	53.9	55.3	42.6	50.5	51.9	59.7
4: boat	69.4	72.2	75.1	75.0	68.1	73.0	70.9	74.6
5: bottle	19.3	21.5	31.5	30.2	18.3	27.2	30.8	37.8
6: bus	61.2	64.6	67.7	71.0	60.6	65.0	72.2	72.9
7: car	76.1	79.4	82.1	82.5	74.7	80.6	79.9	82.9
8: cat	58.9	59.5	60.8	61.9	56.2	59.3	61.4	67.7
9: chair	50.8	53.2	54.8	55.7	50.1	54.2	56.0	57.8
10: cow	36.7	41.0	44.1	45.7	36.5	42.5	49.6	55.1
11: dining table	39.9	57.0	62.4	59.4	38.8	59.5	58.4	66.7
12: dog	39.5	43.9	45.9	47.8	38.9	44.8	44.8	54.9
13: horse	77.0	77.2	83.0	81.3	76.7	79.5	78.8	81.6
14: motorbike	62.1	65.1	68.2	67.4	61.7	65.4	70.8	71.2
15: person	83.6	86.0	87.0	87.4	82.8	86.0	85.0	87.0
16: potted plant	23.5	27.6	29.9	31.2	22.5	27.6	31.7	37.6
17: sheep	37.9	42.4	44.6	47.6	36.0	42.9	51.0	53.5
18: sofa	46.9	52.8	55.2	57.6	45.0	53.6	56.4	63.0
19: train	75.4	79.4	82.1	82.3	74.3	80.9	80.2	84.0
20: tv/monitor	49.9	52.8	55.3	57.2	47.6	52.9	57.5	60.7
mAP	54.4	58.5	61.6	62.4	53.2	59.5	61.7	66.3

by using a model with 1024 Gaussians.

Finally, Znaidia et al. [2012] reported 68.3% on the PASCAL VOC 2007 but also considering as features the image tags. Without access to such information, their BoW baseline dropped to 52.1%.

5.3.4 Results for 15-Scenes

Results, both the ones implemented and tested by us, and the ones reported on the literature are shown in Table 5.12. Once again, we achieve an absolute improvement of 1.8% from BoW to BOSSA, validating the relevance of our pooling scheme. We also observe that the BN method surpasses BOSSA by 2.4%, which confirms the proposed improvements of BossaNova over BOSSA. In comparison to FV, BN classification performance is peculiarly inferior.We must note for one single class (*industrial*) our result is much lower than expected, weighting down the averages. When combining BN and

Table 5.12: Image classification accuracy (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on 15-Scenes dataset [Lazebnik et al., 2006]. BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013]; BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c]. The table shows the means and standard deviations of the 30 accuracy measures.

	Accuracy (%)
Our methods	
BOSSA [Avila et al., 2011]	82.9 ± 0.5
BN [Avila et al., 2013]	85.3 ± 0.4
BN + FV [Avila et al., 2013]	88.9 ± 0.3
Implemented methods	
BoW [Sivic and Zisserman, 2003]	81.1 ± 0.6
FV [Perronnin et al., 2010c]	88.1 ± 0.2
Published results	
[Yang et al., 2009b]	80.3 ± 0.9
[Lazebnik et al., 2006]	81.4 ± 0.5
[Boureau et al., 2010a]	85.6 ± 0.2
[Krapac et al., 2011]	88.2 ± 0.6

Table 5.13: Image classification accuracy (%) results of BossaNova, BOSSA, standard implemented state-of-the-art representations and published methods on 15-Scenes dataset [Lazebnik et al., 2006]. BOSSA: $M = 4096$, $B = 2$, $\lambda_{min} = 0$, $\lambda_{max} = 2$; BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$, as in [Avila et al., 2013]; BoW: $M = 4096$; FV: 256 Gaussians, as in [Perronnin et al., 2010c]; CVPR'06: [Lazebnik et al., 2006]; ICCV'11: [Krapac et al., 2011].

	Our methods			**Impl. methods**		**Published results**	
	BOSSA	BN	BN + FV	BoW	FV	CVPR'06	ICCV'11
1: bedroom	71.8	72.3	75.5	67.6	74.8	68.3	73
2: coast	87.5	88.1	89.8	86.3	89.4	82.4	90
3: forest	95.9	96.1	96.2	94.7	96.2	94.7	96
4: highway	85.3	88.1	92.2	83.6	90.3	86.6	91
5: industrial	65.4	70.7	80.0	61.5	78.2	65.4	79
6: inside city	82.3	84.0	88.4	78.2	87.3	80.5	91
7: kitchen	74.3	77.1	82.3	72.5	81.8	68.5	82
8: living room	59.1	68.4	74.3	58.6	73.5	60.4	71
9: mountain	90.1	90.2	95.1	89.3	94.3	88.8	93
10: office	98.5	98.9	99.8	96.2	99.6	92.7	96
11: open country	75.7	78.0	80.0	74.7	79.4	70.5	83
12: store	75.9	82.0	89.5	74.9	88.1	76.2	84
13: street	89.6	92.7	94.2	88.8	93.1	90.2	94
14: suburb	100.0	100.0	100.0	98.2	100.0	99.4	100
15: tall building	91.5	93.6	96.9	90.7	95.6	91.1	96
Accuracy	82.9	85.3	88.9	81.1	88.1	81.4	88.2

FV methods, that issue is solved, and the combination is better than FV in isolation. The combination BN + FV surpasses both individual methods for 13 out of 15 natural scene categories (see Table 5.13).

We also compare our results with those of the best systems reported in the literature. BN outperforms considerably the methods reported by Yang et al. [2009b] and Lazebnik et al. [2006], using improved BoW with sparse coding and max pooling. If we now take our best result (88.9%), we observe that it is slightly better than the result of Krapac et al. [2011], obtained with spatial FV. Again, that comparison is relevant since both of us employ similar low-level local descriptor extractions.

Figure 5.3 illustrates the confusion matrix for our best classification performance. Not surprisingly, confusion occurs between indoor classes (*e.g.*, *bedroom*, *living room*, *kitchen*), urban architecture classes (*e.g.*, *inside city*, *street*, *tall building*) and also between natural classes (*e.g.*, *coast*, *open country*). Our result reaches near state-of-the-art performance for that dataset.

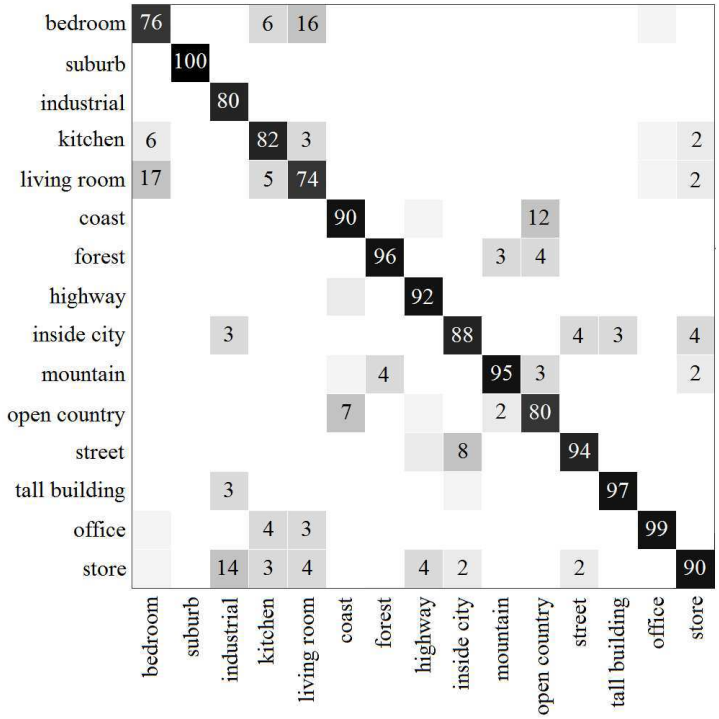


Figure 5.3: Confusion matrix for the 15-Scenes dataset [Lazebnik et al., 2006]. The average classification rates for individual classes are listed along the diagonal, and the columns are the true classes.

5.4 BossaNova in the ImageCLEF 2012 Challenge

In this section, we report our results in the ImageCLEF 2012 Photo Annotation task. The findings according to the official evaluations confirms that: the proposed image representation in this dissertation has the potential to become a new standard representation in image classification tasks.

In total, 13 teams submitted 28 runs exclusively used visual features, where the maximum number of runs per team was limited to five. In our participation, we submitted four runs. Our best result (mAP = 34.4%), which applies the combination of BossaNova and Fisher Vector representations, achieved the second rank among the 28 purely visual submissions, while our BossaNova representation achieved the third rank (mAP = 33.6%), see Table 5.14.

Table 5.14 shows the best run of the five best teams in the ImageCLEF 2012 Flickr Photo Annotation task[5], and details the parameter settings for our method. We also show the results of each concept for the five best teams (see Table 5.15). Among those teams, all differences are significant with at least 99% confidence, except for our team (Top2) and the first team (Top1), whose difference is not significant.

Table 5.14: Image classification mAP (%) results for the best visual run per team on ImageCLEF 2012 Flickr Photo Annotation task [Thomee and Popescu, 2012]. BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$; FV: 384 Gaussians.

	Rank	mAP (%)
[Liu et al., 2012a]	1	34.8
BN + FV [Avila et al., 2012]	2	34.4
BN [Avila et al., 2012]	3	33.6
Paper not available	6	33.2
[Ushiku et al., 2012]	10	32.4
[Xioufis et al., 2012]	11	31.8

The best system [Liu et al., 2012a] reported 34.8% mAP, applying a combination of the top 5 features among the 24 visual features (including color, texture, shape, high level, and SIFT) for each concept based on the Selective Weighted Late Fusion scheme [Liu et al., 2012b]. Also, they applied BoW models with 4000 codewords and soft assignment. The method of Ushiku et al. [2012] uses numerous descriptors (SIFT, C-SIFT, RGB-SIFT, OpponentSIFT and LBP). Fisher Vectors are used with 256 Gaussians. A linear classifier for each label is obtained with an online multilabel learning called Passive-Aggressive with Averaged Pairwise Loss. The approach of Xioufis et al.

[5]All results are available at http://www.imageclef.org/2012/photo-flickr/annotation.

Table 5.15: Image classification AP and mAP (%) results for the best visual run per team on ImageCLEF 2012 Flickr Photo Annotation task [Thomee and Popescu, 2012]. Ours (BN + FV) BN: $M = 4096$, $B = 2$, $\lambda_{min} = 0.4$, $\lambda_{max} = 2$, $s = 10^{-3}$; FV: 384 Gaussians; Top1 [Liu et al., 2012a]; Top6 (paper not available); Top10 [Ushiku et al., 2012]; Top11 [Xioufis et al., 2012]

	Top1	Ours	Top6	Top10	Top11
1: timeofday_day	52.4	53.8	54.2	51.8	54.7
2: timeofday_night	34.9	34.6	35.1	31.9	32.4
3: timeofday_sunrisesunset	44.0	39.0	39.7	38.4	43.8
4: celestial_sun	50.4	47.0	45.3	44.4	50.5
5: celestial_moon	38.9	36.1	31.6	36.1	26.8
6: celestial_stars	63.3	66.3	65.0	64.0	54.0
7: weather_clearsky	60.0	53.1	56.7	49.9	56.3
8: weather_overcastsky	35.1	32.6	29.5	29.5	29.7
9: weather_cloudysky	68.3	65.7	64.7	65.0	64.5
10: weather_rainbow	51.5	37.9	43.0	33.3	45.4
11: weather_lightning	24.7	16.9	18.2	14.3	18.6
12: weather_fogmist	33.3	33.2	30.4	28.2	32.5
13: weather_snowice	23.4	20.6	23.6	18.4	21.6
14: combustion_flames	19.6	14.5	19.1	8.4	21.5
15: combustion_smoke	18.4	19.3	16.9	14.8	16.9
16: combustion_fireworks	67.3	72.2	61.9	68.4	64.6
17: lighting_shadow	28.5	25.7	23.7	23.1	25.4
18: lighting_reflection	43.0	45.3	41.1	41.3	35.0
19: lighting_silhouette	57.4	57.4	54.9	55.4	55.6
20: lighting_lenseffect	43.3	41.4	40.1	40.1	42.7
21: scape_mountainhill	26.9	35.5	31.8	32.8	32.3
22: scape_desert	17.8	7.5	7.6	6.3	9.4
23: scape_forestpark	53.5	50.0	52.2	46.8	50.2
24: scape_coast	60.4	60.0	59.9	60.8	58.3
25: scape_rural	33.9	32.2	29.9	30.0	31.0
26: scape_city	61.9	65.0	61.3	61.8	56.3
27: scape_graffiti	34.9	31.6	33.3	31.9	29.6
28: water_underwater	25.7	18.9	28.0	10.2	29.8
29: water_seaocean	30.2	30.5	29.0	31.0	28.9
30: water_lake	17.1	25.7	21.0	25.1	18.4
31: water_riverstream	22.4	22.3	19.4	20.0	22.6
32: water_other	8.9	18.6	16.7	11.0	13.5
33: flora_tree	71.4	73.0	69.6	68.4	69.1
34: flora_plant	21.1	13.9	14.6	15.4	20.3
35: flora_flower	64.3	58.9	53.3	56.8	61.4
36: flora_grass	58.8	48.3	51.3	47.0	57.1
37: fauna_cat	25.0	29.9	24.0	27.0	16.2
38: fauna_dog	42.9	44.7	41.4	44.9	37.1
39: fauna_horse	26.5	26.1	22.4	23.5	21.4
40: fauna_fish	14.2	14.7	20.3	11.1	16.7
41: fauna_bird	31.6	38.5	29.3	30.4	28.6
42: fauna_insect	28.8	26.0	24.5	24.1	21.6
43: fauna_spider	11.4	19.5	1.7	3.3	9.1
44: fauna_amphibianreptile	1.1	1.3	1.5	1.7	3.7

continued on next page

	Top1	Ours	Top6	Top10	Top11
45: fauna_rodent	16.8	22.2	16.2	20.3	7.5
46: quantity_none	92.3	92.7	92.2	92.7	91.2
47: quantity_one	60.3	60.9	61.2	59.8	56.5
48: quantity_two	14.3	20.3	16.3	14.0	11.7
49: quantity_three	6.6	13.3	6.7	6.8	6.2
50: quantity_smallgroup	25.5	22.6	26.8	25.1	21.9
51: quantity_biggroup	44.8	45.2	40.0	38.4	39.8
52: age_baby	30.9	32.2	24.2	25.9	23.6
53: age_child	21.1	20.4	14.2	20.0	13.0
54: age_teenager	15.5	10.0	16.0	10.0	16.3
55: age_adult	61.9	61.2	62.2	61.9	60.2
56: age_elderly	14.3	8.6	9.3	8.0	6.9
57: gender_male	52.4	53.1	51.6	51.7	49.2
58: gender_female	58.4	61.6	58.1	60.0	57.3
59: relation_familyfriends	34.9	36.4	33.1	33.8	29.2
60: relation_coworkers	20.5	21.4	19.2	21.4	22.0
61: relation_strangers	22.3	17.8	23.2	20.5	17.6
62: quality_noblur	87.9	88.7	87.5	88.8	87.5
63: quality_partialblur	76.1	79.0	76.0	77.3	72.5
64: quality_completeblurv	18.8	24.1	20.3	25.3	19.4
65: quality_motionblur	35.8	39.5	31.6	35.3	29.2
66: quality_artifacts	15.4	15.6	19.5	18.6	17.8
67: style_pictureinpicture	12.6	22.3	18.1	17.3	6.4
68: style_circularwarp	32.4	32.7	32.3	29.7	23.7
69: style_graycolor	29.0	4.0	15.9	8.6	21.7
70: style_overlay	25.2	31.8	23.6	27.5	18.6
71: view_portrait	37.4	39.0	39.7	39.9	36.9
72: view_closeupmacro	34.9	33.9	35.2	31.4	36.7
73: view_indoor	40.8	39.4	41.4	37.1	38.0
74: view_outdoor	59.1	58.1	58.4	55.8	58.3
75: setting_citylife	59.9	60.4	58.0	58.7	55.7
76: setting_partylife	32.1	30.9	31.7	30.0	27.5
77: setting_homelife	39.4	40.0	38.6	36.2	35.5
78: setting_sportsrecreation	20.3	20.2	22.9	18.7	21.3
79: setting_fooddrink	55.5	52.0	49.7	51.4	49.7
80: sentiment_happy	26.2	20.2	22.5	19.8	25.1
81: sentiment_calm	40.0	38.7	38.4	37.9	37.8
82: sentiment_inactive	23.3	16.2	23.7	24.0	15.6
83: sentiment_melancholic	21.2	13.2	21.1	14.9	15.4
84: sentiment_unpleasant	7.1	9.0	8.8	8.7	9.2
85: sentiment_scary	10.1	7.3	15.6	10.3	8.9
86: sentiment_active	16.1	17.1	15.7	14.8	18.1
87: sentiment_euphoric	3.4	6.4	4.6	3.6	5.2
88: sentiment_funny	14.3	19.6	14.3	14.3	19.5
89: transport_cycle	38.6	38.9	33.6	35.2	27.8
90: transport_car	47.1	49.9	46.3	46.9	37.8
91: transport_truckbus	8.6	12.2	7.3	13.0	2.1
92: transport_rail	29.2	27.8	23.6	31.0	22.5
93: transport_water	16.5	25.7	16.7	22.9	13.7
94: transport_air	16.8	12.7	17.1	15.4	13.7
mAP	34.8	34.4	33.2	32.4	31.8

[2012] also employs several descriptors (SURF, SIFT and color SIFT) which are used by different visual representations (BoW, VLAD and VLAT). For each combination of descriptor, a multi-label model is built using the Binary Relevance approach coupled with Random Forests as the base classifier. Moreover, a late fusion scheme averages the output of the different multi-label models.

In short, we can notice (again) that all those top-performing systems employ complex combinations of several low-level features to achieve their good results. Our team achieved the second and the third rank using a single low-level feature (SIFT descriptors) and SVM classifiers. On account of that, our results of 34.4% for BN + FV, and 33.6% for BN are notably good.

5.5 Conclusion

In this chapter, we have presented our experimental results, which were organized in three groups. First, we have proposed to evaluate the impact of each improvement of BossaNova over BOSSA in statistically sound experiments. We have validated through a Student t-test the relevance of the three modifications. Also, we have analyzed the significance of each improvement (and combinations) using the ANOVA test. We have observed that the semi-soft assignment explains almost 48% of the improvements, while the normalization explains about 31%. The weighting scheme, however, is responsible for only 3% of the variation.

The second round of experiments has explored the behavior of the key parameters in BossaNova representation: the number of codewords M, the number of bins B in each local histogram z_m, and the range of distances $[\alpha_m^{min}, \alpha_m^{max}]$.

Finally, the third group of experiments are a comparison with state-of-the-art methods, which have allowed us: (i) to confirm the relevance of the improved pooling scheme introduced in this dissertation, (ii) to show the benefits brought out by the three proposed improvements of BossaNova over BOSSA; (iii) to observe that BossaNova is tied with Fisher Vector, the current state-of-the-art method; and iv) to validate the complementariness of BossaNova and Fisher Vector representations.

Additionally, we have reported our results in the ImageCLEF 2012 Photo Annotation task, in which we have achieved the 2$^{\text{nd}}$ rank among 28 visual submissions and 13 teams.

Chapter 6

Application: Pornography Detection

Pornography consumption has increased in recent years, which is due in large part to the availability and anonymity provided by the Internet [Short et al., 2012]. Pornographic material, however, is often unwelcome is certain environments (*e.g.*, schools, workplaces), channels (*e.g.*, general-purpose social networks), or for certain publics (*e.g.*, children). That raises the need to detect and filter such content.

Pornography is less straightforward to define than it may seem at first, since it is a high-level semantic category, not easily translatable in terms of simple visual characteristics. Though it certainly relates to nudity, pornography is a different concept: many activities which involve a high degree of body exposure (swimming, boxing, sunbathing, etc.) have nothing to do with it. That is why systems based on skin detection [Jones and Rehg, 2002; Zheng and Daoudi, 2004; Rowley et al., 2006; Lee et al., 2009b; Zuo et al., 2010; Bouirouga et al., 2012] often accuse false positives in contexts like beach shots or sports.

A commonly used definition is that pornography is "any sexually *explicit* material with the *aim* of sexual arousal or fantasy" [Short et al., 2012]. That raises several challenges. First and foremost, what threshold of explicitness must be crossed for the work to be considered pornographic? Some authors deal with that issue by further dividing the classes [Deselaers et al., 2008] but that not only falls short of providing a clear cut definition, but also complicates the classification task. The matter of purpose is still more problematic, because it is not an objective property of the document. Here, we have opted to keep the evaluation conceptually simple, by assigning only two classes (pornographic and nonpornographic). On the other hand, we took great care to make them representative.

In this chapter, we explore our approach in the real-world application of pornography detection, which because of its high-level conceptual nature, involves large

intra-class variability. In Section 6.1, we explore some related work, both in terms of images and videos pornography detection. In Section 6.2, we introduce our own pornography dataset. In Section 6.3, we present our scheme for pornography detection. In Section 6.4, we discuss our experimental results. Finally, in Section 6.5, we relate our concluding remarks.

6.1 Related Work

Most work regarding the detection of pornographic material has been done for the image domain [Ries and Lienhart, 2012]. The vast majority of those works is based on the detection of human skin. For example, in [Fleck et al., 1996; Forsyth and Fleck, 1996, 1997, 1999], the authors proposed to detect skin regions in an image and match them with human bodies by applying geometric grouping rules. Jones and Rehg [2002] focused on the detection of human skin by constructing RGB color histograms from a large dataset of skin and non-skin pixels, which allows to estimate the "skin probability" of a pixel based on its color. Rowley et al. [2006] used Jones and Rehg' skin color histograms in a system installed in Google's Safe Search. Lee et al. [2007] developed a learning-based chromatic distribution matching scheme to determine the image's skin chroma distribution. Zuo et al. [2010] introduced a patch-based skin color detection that verifies whether all the pixels in a small patch correspond to human skin tone. Hu et al. [2011] also proposed to model skin patches rather than skin pixels.

Few methods have explored other possibilities. Bag-of-Words models (see Section 2.2) have been employed for many complex visual classification tasks, including pornography detection in images and videos. Deselaers et al. [2008] first proposed a BoW model to filter pornographic images, which greatly improved the efficiency of the identification of pornographic images. Lopes et al. developed a BoW-based approach, which used the HueSIFT color descriptor, to classify images [Lopes et al., 2009b] and videos [Lopes et al., 2009a] of pornography. Ulges and Stahl [2011] introduced a color-enhanced visual word features in YUV color space to classify child pornography. Steel [2012] proposed a pornographic images recognition method based on visual words, by using mask-SIFT in a cascading classification system.

Those previous works have explored only bags of static features. Very few works have been applied spatiotemporal features or other motion information (such as optical flow, feature trajectories) for detection of pornography. Tong et al. [2005] proposed a method to estimate the period of a signal to classify periodic motion patterns. Endeshaw et al. [2008] developed a fast method for detection of indecent video content

using repetitive motion analysis. Jansohn et al. [2009] introduced a framework that combines keyframe-based methods with a statistical analysis of MPEG-4 motion vectors. Valle et al. [2012] compared the use of several features, including spatiotemporal local descriptors for video (such as STIP descriptor [Laptev, 2005]), in a BoW-based approach for pornography detection.

Also, other approaches have been employed audio analysis as an additional feature for the identification of pornographic videos. Rea et al. [2006] combined skin color estimation with the detection of periodic patterns in a video's audio signal. Liu et al. [2011b] demonstrated improvements by fusion visual features (color moments and edge histograms) with "audio words". In a similar fashion, Ulges et al. [2012] proposed an approach of late fusing motion histograms with "audio words".

The importance of pornography detection is attested by the large literature on the subject. Web filtering is essential to avoid adult or pornographic material where it is not welcome. There are commercial softwares that block Web sites with this kind of content (*e.g.*, CyberPatrol, NetNanny, K9 Web Protection). Also, there is software which scan a computer for pornographic content (*e.g.*, SurfRecon, Porn Detection Stick, PornSeer Pro). The latter pornography-detection software, the PornSeer Pro, is readily available for evaluation purposes.

6.2 The Pornography Dataset

There are no standardized datasets for pornography detection, primarily due to copyright issues and the potential legal limitations on distributions of large quantities of pornographic material. As such, a representative dataset of internet videos, both pornographic and nonpornographic, was created for this experiment.

The Pornography dataset contains nearly 80 hours of 400 pornographic and 400 nonpornographic videos. For the pornography class, we have browsed websites which only host that kind of material[1] (solving, in a way, the matter of purpose). The dataset consists of several genres of pornography and depicts actors of many ethnicities, including multi-ethnic ones (see Table 6.1).

For the nonpornography class, we have browsed general-public purpose video network (*e.g.*, YouTube) and selected two samples: 200 videos chosen at random (we called "easy") and 200 videos selected from textual search queries like "beach", "wrestling", "swimming", which we knew would be particularly challenging for the detector ("difficult") – the exposure of skin imposes a challenge to the system.

[1]For example, www.{RedTube, XTube, PornTube, Xvideos}.com

Table 6.1: Ethnic diversity on the pornographic videos.

Ethnicity	**% of Videos**
Asians	16%
Blacks	14%
Whites	46%
Multi-ethnic	24%

In order to download the videos, we benefited from batch downloader softwares, for example: we use the YouTube Robot[2] to download the "easy" nonpornographic videos; for "difficult" nonpornographic videos we employ the VDownloader[3], which allows us to manually select the videos; and to download the pornographic videos we make use of RedTube Grabber[4].

Figure 6.1 shows selected frames from a small sample of the dataset, illustrating the diversity of the pornographic videos and the challenges of the "difficult" nonpornographic ones. The Pornography dataset is not generally available to the community at large, due to copyright problems, but access to it can be granted after a case-by-case analysis, and the acceptance of an agreement available at http://www.npdi.dcc.ufmg.br/pornography.

Figure 6.1: Illustration of the diversity of the pornographic videos (top row) and the challenges of the "difficult" nonpornographic ones (middle row). The easy cases are shown at bottom row. The huge diversity of cases in both pornographic and nonpornographic videos makes that task very challenging.

[2]http://www.youtuberobot.com/
[3]http://vdownloader.com/
[4]http://www.redtube-grabber.com/

6.3 Our Scheme

The scheme we propose works by extracting elements from the video, extracting low-level features from those elements, generating mid-level representations and training the classifier. In the classification phase, the classifier opinion is asked for each individual video element, and the final decision is reached by majority voting. The whole scheme is illustrated on Figure 6.2 and explained in the following.

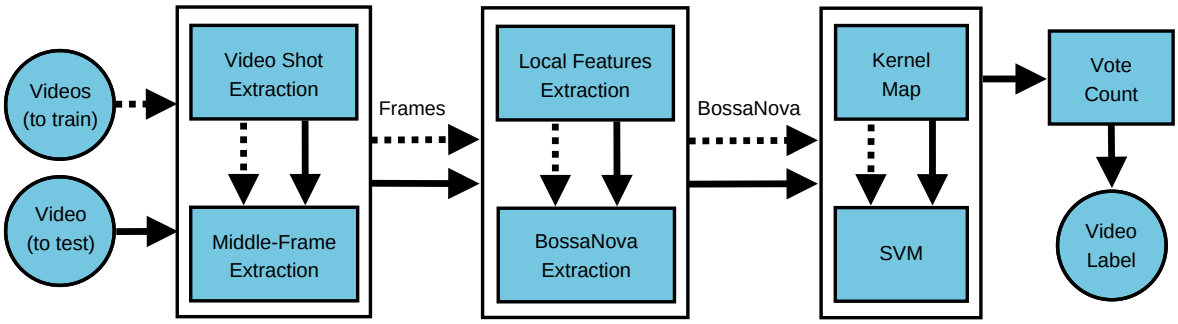


Figure 6.2: Our scheme for pornography video classification. The data flow for training is represented by the dashed lines, while the data flow for classification is shown on solid lines.

Preprocessing: We preprocess the dataset by segmenting videos into shots. An industry-standard segmentation software[5] has been used. On average there are 20 shots per video.

As it is often done in video analysis, a keyframe is selected to summarize the content of the shot into a static image. Although there are sophisticated ways to choose the keyframe, in this proof-of-concept application, we opted to simply selected the middle-frame of each video shot. In total, there are 16,727 shots. Table 6.2 summarizes the Pornography dataset.

Feature extraction: In the low-level feature extraction, we have extracted local descriptors for each frame, in particular, HueSIFT descriptors [van de Sande et al., 2010]. In the mid-level feature extraction, we apply our proposed BossaNova representation (see Chapter 4).

Training: The training step is performed by the SVM classifier. Here, care is taken to balance the classes (porn and nonporn) so each is given roughly the same

[5]http://www.stoik.com/products/svc/

number of training videos at this step. We apply a classical 5-fold cross-validation, generating nearly 640 videos for training and 160 for testing on each fold (see Table 6.3).

Classification: The SVM classifier casts a vote over each frame: positive (porn) or negative (nonporn). The majority label is given to the video.

Table 6.2: Summary of the Pornography dataset.

Class	**Videos**	**Hours**	**Shots per video**
Porn	400	57	15.6
Nonporn ("Easy")	200	11.5	33.8
Nonporn ("Difficult")	200	8.5	17.5
All videos	**800**	**77**	**20.6**

Table 6.3: Number of frames (shots) for each training and testing sets in the Pornography dataset. In total, each run contains nearly 640 videos for training and 160 for testing.

Runs	**#train**		**#test**	
	nonporn	porn	nonporn	porn
run1	8,194	4,909	2,146	1,478
run2	8,488	4,933	1,852	1,454
run3	8,470	5,144	1,870	1,243
run4	8,351	5,262	1,989	1,125
run5	7,857	5,300	2,483	1,087

6.4 Experiments

In the experiments, we investigated the power of the BossaNova representation for pornography detection. Our main goal is to compare the performance of BossaNova [Avila et al., 2013] with BOSSA [Avila et al., 2011].

Also, obtaining a baseline to compare with our method was a major challenge since, in general, the numbers reported on the literature are not comparable from one work to another. Often, the datasets are given only very cursory description, making next to impossible to make a fair assessment of the actual experimental conditions.

Therefore, we have opted to compare ourselves to PornSeer Pro[6], an industry standard video pornography detection system. It is based on the detection of specific features (like breast, genitals or the act of intercourse) on individual frames. It examines each individual frame of the video.

In this section, we first describe our experimental setup and we then show and discuss our results.

6.4.1 Experimental Setup

As a low-level local descriptor, we employ the 165-dimensional HueSIFT descriptor [van de Sande et al., 2010], a SIFT variant including color information, which is particularly relevant for our dataset. The HueSIFT descriptors are extracted densely every 6 pixels and a sampling scale of 1.2. As a result, 3,500 local descriptors, on average, are extracted from each image of Pornography dataset.

In order to learn the codebooks, we apply the k-means clustering algorithm with Euclidean distance over one million randomly sampled descriptors. For classification, we use a nonlinear Gauss-ℓ_2 kernel. Kernel matrices are computed as $exp(-\gamma d(x, x'))$ with d being the distance and γ being set to the inverse of the pairwise mean distances.

We report the image classification performance by using the mean Average Precision (mAP), and the video classification by accuracy rate, where the final video label is obtained by majority voting over the images. We also use a confusion table to illustrate the results.

6.4.2 Results

Table 6.4 shows the results of our experiments over Pornography dataset, and details the parameter settings for each method.

Once again, as we observed in the Chapter 5, BossaNova outperforms both BoW and BOSSA representations. Comparing BOSSA with BoW, we already notice a considerable improvement of 3.2% and 4.1% for image and video classification, respectively. If we now compare BossaNova with BOSSA, we also observe a considerable increase of 1.8% and 2.4% for image and video classification, respectively. That confirms the advantages introduced by BossaNova representation.

We also compare our results with the PornSeer Pro, a pornography-detection software, which uses two parameters "threshold" and "decision" to tune the relation between hit rate and false alarm rate. Usually, a small "threshold" is used together

[6]http://www.yangsky.com/products/dshowseer/porndetection/PornSeePro.htm

Table 6.4: Comparison of the BossaNova, BOSSA and BoW representations on the Pornography dataset. mAP (%) is computed at image classification level, and Accuracy rate is reported for video classification. For each method, we use their tested configuration parameters, namely BN: $M = 256$, $B = 10$, $\lambda_{min} = 0$, $\lambda_{max} = 3$, $s = 10^{-3}$; BOSSA: $M = 256$, $B = 10$, $\lambda_{min} = 0$, $\lambda_{max} = 3$; BoW: $M = 256$.

	mAP (frames)	Acc. rate (videos)
Our methods		
BN [Avila et al., 2013]	96.4 ± 1	89.5 ± 1
BOSSA [Avila et al., 2011]	94.6 ± 1	87.1 ± 2
Implemented methods		
BoW [Sivic and Zisserman, 2003]	91.4 ± 1	83.0 ± 3

Table 6.5: The average confusion matrix for BossaNova.

		Video was labeled as	
		porn	nonporn
Video was	porn	88.2%	11.8%
	nonporn	9.2%	90.8%

Table 6.6: The average confusion matrix for PornSeer Pro.

		Video was labeled as	
		porn	nonporn
Video was	porn	65.1%	34.9%
	nonporn	12.5%	87.5%

with a small "decision" to keep the same false alarm rate and the hit rate. We employed the PornSeer Pro default values. Tables 6.5 and 6.6 show the confusion matrices for our BossaNova and the PornSeer Pro.

6.4.3 Discussion

Our scheme is able to correctly identify 9 out of 10 of the pornographic videos, with few false positives. This is very important, since the cost of false alarms is high on the social network context, for it tends to overwhelm the human operators. The false positive rate attained may appear high at first, but it must be taken in the context of a very challenging dataset. Considering that half of the nonpornographic test videos were difficult cases, the rates are, actually, low.

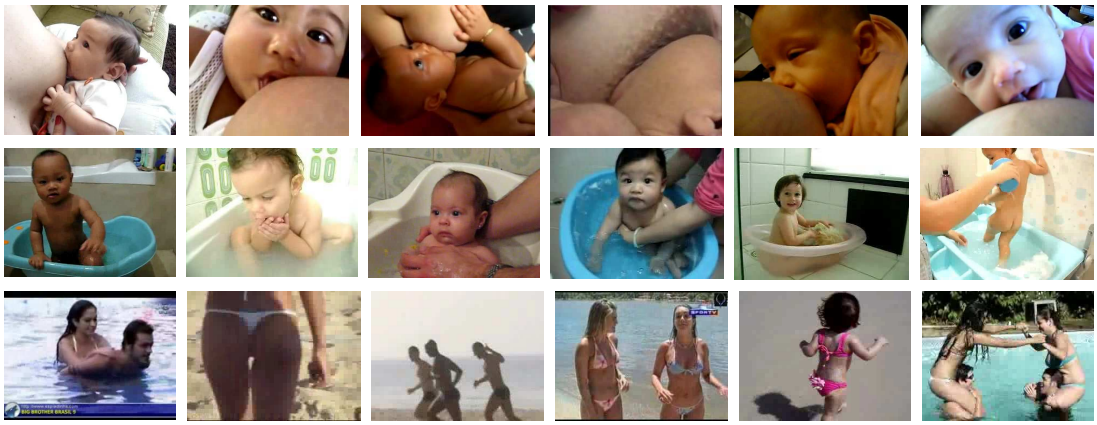


Figure 6.3: Frames examples corresponding to very challenging nonpornographic videos: breastfeeding frames (top row), frames of children being bathed (middle row), and beach frames (bottom row).

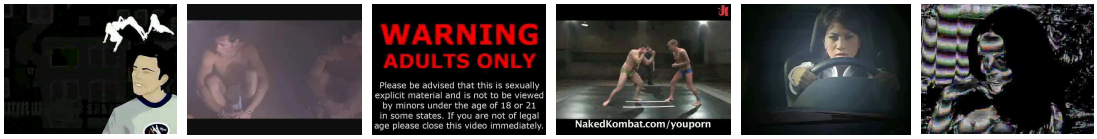


Figure 6.4: Frames examples corresponding to very challenging pornographic videos: frames with very poor quality and with few explicit elements.

It is instructive to study the cases where our method fails. The stubborn false positives correspond to very challenging nonpornographic videos: breastfeeding sequences, sequences of children being bathed, and beach scenes (see Figure 6.3). The method succeeds for many videos with those subjects, but those particular ones have the additional difficulty of having very few shots (typically 1 or 2), giving no allowance for classification errors. PornSeer Pro gave a wrong classification for all those clips.

The analysis of the most hard false negatives revealed that the method has difficulty when the videos are of very poor quality (typical of amateur porn, often uploaded from webcams) or when the clip is only borderline pornographic, with few explicit elements (see Figure 6.4). PornSeer also had difficulty with those clips, misclassifying many of them.

6.5 Conclusion

Internet pornography use has increased over the past 10 years [Short et al., 2012]. The example given by Xvideos, the biggest porn site on the web, is illustrative, with

4.4 billion page views and 350 million unique visitors per month[7]. Also, a report by New York-based technology site, the ExtremeTech[8], suggests that a staggering 30% of all internet traffic is pornography. The increasing prevalence of pornographic content poses a challenge, because such content is not welcome in some environments or for some kinds of public (*e.g.*, children), generating the need to detect and filter it.

In this chapter, we have explored our BossaNova approach in the challenging real-world application of pornography detection. Our scheme has as its advantage the fact that it does not depend on any skin detector or shape models to classify pornography; besides, it shows good results.

Additionally, our results can be improved even further by considering recent local descriptors. For example, Wang et al. [2011] introduced a novel local descriptor, based on motion boundary histograms, to encode the trajectory information. In the context of action classification, their descriptor with a BoW-based approach consistently outperforms other state-of-the-art descriptors. Ullah and Laptev [2012] proposed a supervised approach to learn local motion descriptors from a large pool of annotated video data. The authors have shown in their experiments that the proposed representation is discriminative as well as complementary to BoW representation.

[7]http://digitaljournal.com/article/322668#ixzz2E2OCBk80
[8]http://www.extremetech.com/computing/123929-just-how-big-are-porn-sites

Chapter 7

Conclusion

In this chapter, we provide a summary of the major contributions and findings of this dissertation. In addition, we discuss some interesting issues that we could not address due to our limitation of time and scope, and that we left as future work directions.

7.1 Contributions

The main objective of this dissertation was studying and advancing the state-of-the-art in mid-level image representations for tasks of classification. The BossaNova image representation [Avila et al., 2012, 2013] is the corollary of those efforts. Among the contributions of our work, we emphasize:

- Definition and implementation of a novel image representation for classification tasks. After analyzing the BoW model, we have pointed out the weakness of the standard pooling operation and, therefore, we have proposed the BossaNova image representation [Avila et al., 2013], which offers a more information-preserving pooling operation based on a distance-to-codeword distribution. Our scheme has the advantage of being conceptually simple, and easily adaptable. A preliminary version of the BossaNova, called BOSSA [Avila et al., 2011], has allowed us to gain several insights into the benefits of the density-based choice of the representation and to explore the compromises between the opposite goals of discrimination versus generalization, representativeness versus compactness.

- Statistical evaluation of the impact of the three proposed improvements (the semi-soft coding scheme, the normalization strategy, and the weighting scheme) of BossaNova over BOSSA. By analyzing the isolated and joint impact of each enhancement on the BossaNova representation, we have validated through a t-test

the relevance of the three modifications. Also, we have analyzed the significance of each improvement (and combinations) using the statistical test ANOVA. We have observed that the semi-soft assignment explains almost 48% of the improvements, while the normalization explains about 31%. Those results confirm the importance of the normalization step, which was in the past often neglected.

- Experimental evaluation of state-of-the-art representations based on the BoW model for classification tasks, including representations we reimplemented ourselves, and also results reported in literature from standard datasets/protocols. We have observed the importance of controlling carefully all conditions when comparing different representations. The empirical comparisons on challenge benchmarks (MIRFLICKR, ImageCLEF 2011/2012 Photo Annotation, VOC 2007 and 15-Scenes) have shown the advantage of BossaNova when compared to traditional techniques. Moreover, our participation at ImageCLEF 2012 Photo Annotation challenge achieved the 2nd rank among all submissions using only visual information [Avila et al., 2012], with the absolute difference between the first and our result was 0.4%. Hence, the BossaNova representation has the potential to advance the state of the art in image representations for concept detection.

- Proposal and evaluation of a novel image representation based on complementarity of BossaNova and Fisher Vector representation. The latter representation models the distribution of local descriptors in each codeword with a single Gaussian. However, when that Gaussian assumption does not hold, the pooled representation may be unrepresentative of the local descriptor statistics. In contrast to that, BossaNova representation uses additional locality constraints during the pooling. We have confirmed empirically on many benchmarks that combining BossaNova with Fisher Vector indeed boosts the classification performances.

- Empirical evaluation of BossaNova representation in the challenging real-world application of pornography detection, and the development of a novel dataset to support this challenge. Our pornography dataset is not freely available, due to copyright issues and the potential legal limitations on distributions of large quantities of pornographic material. However, the data is available to researchers, on provision that a user agreement is signed http://www.npdi.dcc.ufmg.br/pornography.

- Publication of the BossaNova source code, which can be downloaded from http://www.npdi.dcc.ufmg.br/bossanova/. We hope that it would provide common ground for future comparisons.

7.2 Future Work

In addition to the contributions presented in this dissertation, a number of open questions were raised that suggest further investigation.

BossaNova Parameters Study

In Chapter 5, we have experimentally evaluated the BossaNova parameters. Regarding the bin quantization (Section 5.2.2), we have observed that increasing the number of bins yields a slight amelioration in average performance. However, the growth depends on the visual concept (in this case, a concept of MIRFLICKR dataset). On that account, we aim to validate the behavior of bin quantization in other datasets. Hence, we believe that we can improve our results by setting the number of bins per concept.

Considering now the effects of the minimum distance on BossaNova classification performance (Section 5.2.3), we have noticed that setting the minimum distance according to Figure 5.2 (*i.e.*, $\lambda_{min} = 0.4$ and $\lambda_{max} = 2$, which corresponds to 95% of the total SIFT descriptors on the whole dataset), leads to considerable improvements for the most of the concepts and also a decrease for some ones. Therefore, we propose setting a λ_{min} and even a λ_{max} per codeword, aiming at enhancing our representation. By fixing a minimum distance per codeword, we avoid the empty regions that appear around each codeword, and consequently wasting space in the final descriptor.

Multiple Combinations of Descriptors and Classifiers

A trend in the top-performing BoW systems is to have multiple combinations of patch detectors, descriptors and spatial pyramids, to train one classifier per channel and then to combine the output of the classifiers [Binder et al., 2011; Liu et al., 2012a]. Systems following this paradigm have consistently performed among the best in the successive ImageCLEF evaluations [Nowak et al., 2011; Thomee and Popescu, 2012], for example.

By contrast, in Chapter 5, we have shown our empirical results using only SIFT descriptors and nonlinear SVM classifiers. Considering that very simple experimental setup, our BossaNova scheme obtained remarkable results on several benchmarks.

Now equipped with that representation, we want to evaluate the use of multiple descriptors (both local and global features) and a multiple kernel learning (MKL) algorithm [Vedaldi et al., 2009], in order to learn a combination of different kernel functions, obtaining a similarity measure that better matches the underlying problem. In this way, we hope to significantly increase the performance.

We also hope to improve our results by exploiting further late fusion strategies to combine BossaNova and Fisher Vector representations. Late fusions schemes have shown notable results on various benchmarks, *e.g.*, ImageCLEF 2012 Photo Annotation [Liu et al., 2012a] and PASCAL VOC 2007 [Sánchez et al., 2012].

Large-Scale Experiments

In a classification context, the proposed BossaNova representation is used in conjunction to Gauss-ℓ_2 nonlinear kernels, because linear SVMs have been repeatedly reported to be inferior to nonlinear SVMs on BoW-based representation [Perronnin et al., 2010b; Vedaldi and Zisserman, 2012].

The learning of nonlinear SVMs scales somewhere between $O(N^2)$ and $O(N^3)$ (where N is the number of training images) and becomes impractical for large-scale problems, *i.e.* databases with more than one million images, such as ImageNet Large Scale Visual Recognition 2012 dataset[1] (1000 categories and 1,2 million training images). This is in contrast with linear SVMs whose training cost is in $O(N)$ [Joachims, 2006] and which can therefore be efficiently learned with large quantities of images. Note that for the datasets we evaluated (MIRFLICKR, ImageCLEF 2011, ImageCLEF 2012, PASCAL VOC 2007, 15-Scenes, Oxford Flowers), nonlinear SVMs are suited for training and testing.

Recent works have focused on approximating nonlinear kernels by linear ones, by providing approximated features maps [Vedaldi and Zisserman, 2012; Williams and Seeger, 2001]. In most cases, the approximated representations reach performances comparable to those of the exact kernels. Therefore, we aim to apply those strategies upon BossaNova to handle large-scale visual recognition tasks.

7.3 Publications

Journal

- **S. Avila**, N. Thome, M. Cord, E. Valle and A. Araújo. Pooling in image representation: the visual codeword point of view. Computer Vision and Image Understanding (CVIU): Special Issue on Visual Concept Detection, 117(5), pages 453–465, 2013.

[1] http://www.image-net.org/challenges/LSVRC/2012/

International Conferences

- T. Durand, N. Thome, M. Cord, and **S. Avila**. Image classification using object detectors. In: International Conference on Image Processing (ICIP), Melbourne, Australia, September 2013.

- **S. Avila**, N. Thome, M. Cord, E. Valle and A. de A. Araújo. BossaNova at ImageCLEF 2012 Flickr Photo Annotation task. In: Working Notes of the Conference and Labs of the Evaluation Forum (CLEF), Rome, Italy, 2012.

- **S. Avila**, N. Thome, M. Cord, E. Valle and A. de A. Araújo. BOSSA: extended BoW formalism for image classification. In International Conference on Image Processing (ICIP), pages 2966–2969, Brussels, Belgium, September 2011.

- A. Lopes, **S. Avila**, A. Peixoto, R. Oliveira, A. de A. Araújo. A bag-of-features approach based on hue-SIFT descriptor for nude detection. In: 17th European Signal Processing Conference (EUSIPCO), pages 1552–1556, Glasgow, 2009.

Brazilian Conferences

- **S. Avila**, N. Thome, M. Cord, E. Valle and A. de A. Araújo. Extended bag-of-words formalism for image classification. In: 26th Conference on Graphics, Patterns, and Images (SIBGRAPI) – Workshop of Theses and Dissertations (WTD), Arequipa, Peru, 2013.

- A. Lopes, **S. Avila**, A. Peixoto, R. Oliveira, M. Coelho, A. de A. Araújo. Nude detection in video using bag-of-visual-features. In: 22th Conference on Graphics, Patterns, and Images (SIBGRAPI), Rio de Janeiro, Brazil, 2009.

- E. Valle, **S. Avila**, F. Souza, M. Coelho, and A. de A. Araújo. Content-based filtering for video sharing social networks. In Brazilian Symposium on Information and Computer System Security (SBSeg), Curitiba, Brazil, 2012.

Others

- EMC^2 Summer School on Big Data. Rio de Janeiro, RJ, Brazil, 04–07 February 2013.

- Workshop for Women in Machine Learning (WiML): Theory, Applications, Experiences. Granada, Spain, December 2011. Poster presentation – BOSSA: extended BoW formalism for image classification.

- ENS/INRIA Visual Recognition and Machine Learning Summer School. Paris, France, 25–29 July 2011. Poster presentation – BOSSA: extended BoW formalism for image classification.

Appendix A

BossaNova Fisher Derivation

In this appendix, we detail the computation of the gradient g introduced in Section 4.8 as a Fisher score. Therefore, recall that the gradient g is given by:

$$g(\theta, X) = \left(\frac{\partial \frac{1}{T} log \mathcal{L}_\theta(X)}{\partial \theta_i} \right)_{i=1}^{N}, \tag{A.1}$$

where θ represents here all the parameters α_i and $\beta_{(q,i)}$, $X = \{x_t, t = 1 \dots T\}$ the image, and $\mathcal{L}$ the likelihood. We parameterize the multinomial laws by using:

$$w_i = \frac{exp(\alpha_i)}{\sum_j exp(\alpha_j)}, \tag{A.2}$$

$$w_{(q,i)} = \frac{exp(\beta_{(q,i)})}{\sum_j exp(\beta_{(j,i)})}. \tag{A.3}$$

As Krapac et al. [2011], we consider the average log-likelihood of the T local features in an image. When reporting the likelihood from $\mathcal{L}_\theta(X) = \prod_{t=1}^{T} p_\theta(x_t)$, we obtain:

$$\begin{aligned} \frac{\partial \frac{1}{T} log \mathcal{L}_\theta(X)}{\partial \theta_i} &= \frac{\partial}{\partial \theta_i} \frac{1}{T} \sum_{t=1}^{T} log(p_\theta(x_t)), \\ &= \frac{1}{T} \sum_{t=1}^{T} \frac{\partial}{\partial \theta_i} log(p_\theta(x_t)), \\ &= \frac{1}{T} \sum_{t=1}^{T} \frac{1}{p_\theta(x_t)} \frac{\partial}{\partial \theta_i} (p_\theta(x_t)). \end{aligned} \tag{A.4}$$

As $p(x|\theta) = p_\theta(x) = \sum_{k=1}^{K} w_k p_k(x)$, we have:

$$\frac{\partial}{\partial \alpha_i}(p_\theta(x_t)) = \sum_{k=1}^{K} p_k(x_t) \frac{\partial}{\partial \alpha_i} w_k. \tag{A.5}$$

Knowing that, it is easy to show:

$$\frac{\partial}{\partial \alpha_i} w_k = \frac{\partial}{\partial \alpha_i} \left(\frac{exp(\alpha_k)}{\sum_j exp(\alpha_j)} \right) = w_k(\mathbb{1}_{k=i} - w_i). \tag{A.6}$$

Then, we have:

$$\begin{aligned} \frac{\partial \frac{1}{T} log \mathcal{L}_\theta(X)}{\partial \alpha_i} &= \frac{1}{T} \sum_{t=1}^{T} \frac{1}{p_\theta(x_t)} \sum_{k=1}^{K} p_k(x_t) w_k (\mathbb{1}_{k=i} - w_i), \\ &= \frac{1}{T} \sum_{t=1}^{T} \frac{1}{p_\theta(x_t)} \left(p_i(x_t) w_i - \sum_{k=1}^{K} p_k(x_t) w_k w_i \right), \\ &= \left(\frac{1}{T} \sum_{t=1}^{T} \frac{p_i(x_t) w_i}{p_\theta(x_t)} \right) - w_i. \end{aligned} \tag{A.7}$$

Using $\gamma_i(x_t) = p_i(x_t) w_i / p_\theta(x_t)$ the probability for observation x_t to have been generated by th i-th mixture term introduced in Section 4.8, we get the final result:

$$\frac{\partial \frac{1}{T} log \mathcal{L}_\theta(X)}{\partial \alpha_i} = \frac{1}{T} \sum_{t=1}^{T} \gamma_i(x_t) - w_i. \tag{A.8}$$

For the $\beta_{(q,i)}$ parameters, we apply the same derivation scheme on the second mixture expression, considering each sub-mixture independently:

$$\begin{aligned} \frac{\partial}{\partial \beta_{(q,i)}}(p_\theta(x_t)) &= \frac{\partial}{\partial \beta_{(q,i)}} \left(\sum_{k=1}^{K} w_k p_k(x_t) \right), \\ &= \frac{\partial}{\partial \beta_{(q,i)}} \left(\sum_{k=1}^{K} w_k \left[\sum_{b=1}^{B} w_{(b,k)} p_b(x_t|k) \right] \right). \end{aligned} \tag{A.9}$$

Since the derivative is null, except for $k = i$, we get:

$$\frac{\partial}{\partial \beta_{(q,i)}}(p_\theta(x_t)) = w_i \sum_{b=1}^{B} \frac{\partial}{\partial \beta_{(q,i)}} \left(w_{(b,i)} \right) p_b(x_t|i). \tag{A.10}$$

As for α_i, we have:

$$\frac{\partial}{\partial \beta_{(q,i)}}(w_{(b,i)}) = w_{(b,i)}(\mathbb{1}_{b=q} - w_{(q,i)}). \tag{A.11}$$

Substituting Equation A.11 into Equation A.10, we get:

$$\begin{aligned}
\frac{\partial}{\partial \beta_{(q,i)}}(p_\theta(x_t)) &= w_i \sum_{b=1}^{B} w_{(b,i)}(\mathbb{1}_{b=q} - w_{(q,i)}) p_b(x_t|i), \\
&= w_i \left(w_{(q,i)} p_q(x_t|i) - w_{(q,i)} \sum_{b=1}^{B} w_{(b,i)} p_b(x_t|i) \right), \\
&= w_i \left(w_{(q,i)} p_q(x_t|i) - w_{(q,i)} p_i(x_t) \right). \qquad \text{(A.12)}
\end{aligned}$$

Consequently, we have:

$$\begin{aligned}
\frac{\partial \frac{1}{T} log \mathcal{L}_\theta(X)}{\partial \beta_{(q,i)}} &= \frac{1}{T} \sum_{t=1}^{T} \frac{w_i}{p_\theta(x_t)} \times w_{(q,i)} \left(p_q(x_t|i) - p_i(x_t) \right), \\
&= \frac{1}{T} \sum_{t=1}^{T} \left(\frac{w_i}{p_\theta(x_t)} p_i(x_t) \gamma_{(q,i)}(x_t) - \frac{w_{(q,i)}}{p_\theta(x_t)} w_i p_i(x_t) \right), \\
&= \frac{1}{T} \sum_{t=1}^{T} \left(\gamma_{(q,i)}(x_t) \gamma_i(x_t) - w_{(q,i)} \gamma_i(x_t) \right), \\
&= \frac{1}{T} \sum_{t=1}^{T} \left(\gamma_{(q,i)}(x_t) - w_{(q,i)} \right) \gamma_i(x_t). \qquad \text{(A.13)}
\end{aligned}$$

Finally, we obtain a vector with the scores A.8 and A.13 for all the parameters as the image representation.

Bibliography

Abdullah, A., Veltkamp, R. C., and Wiering, M. A. (2009). Spatial pyramids and two-layer stacking SVM classifiers for image categorization: a comparative study. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1130–1137. 36

Ahmad, M. B. and Choi, T.-S. (1999). Local threshold and boolean function based edge detection. *IEEE Transactions on Consumer Electronics*, 45(1):674–679. 14

Aizerman, M. A., Braverman, E. A., and Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control*, pages 821–837. 32

Alpaydin, E. (2010). *Introduction to Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2 edition. 34, 35

Ankerst, M., Kastenmüller, G., Kriegel, H.-P., and Seidl, T. (1999). 3d shape histograms for similarity search and classification in spatial databases. In *International Symposium on Advances in Spatial Databases*, pages 207–226. 14

Avila, S., Thome, N., Cord, M., Valle, E., and de A. Araújo, A. (2011). BOSSA: extended BoW formalism for image classification. In *International Conference on Image Processing (ICIP)*, pages 2909–2912. 8, 27, 30, 38, 66, 80, 83, 84, 91, 93, 94, 95, 96, 99, 101, 112, 114, 117

Avila, S., Thome, N., Cord, M., Valle, E., and de A. Araújo, A. (2012). BossaNova at ImageCLEF 2012 Flickr Photo Annotation Task. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF)*. 8, 26, 38, 80, 103, 117, 118

Avila, S., Thome, N., Cord, M., Valle, E., and de A. Araújo, A. (2013). Pooling in image representation: the visual codeword point of view. *Computer Vision and Image Understanding (CVIU), Special Issue on Visual Concept Detection*, 117(5):453–465. 8, 26, 29, 33, 38, 80, 83, 84, 88, 89, 91, 93, 94, 95, 96, 99, 100, 101, 112, 114, 117

Azizpour, H. and Laptev, I. (2012). Object detection using strongly-supervised deformable part models. In *European conference on Computer Vision (ECCV)*, pages 836–849. 42, 43

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. Addison Wesley, 1st edition. 5, 19

Basu, M. (2002). Gaussian-based edge-detection method - a survey. *IEEE International Conference on Systems, Man, and Cybernetics*, 32(3):252–260. 14

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359. 6, 17, 18

Bay, H., Tuytelaars, T., and Gool, L. V. (2006). SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 404–417. 18

Beaudet, P. R. (1978). Rotationally invariant image operators. In *International Joint Conference on Pattern Recognition*, pages 579–583. 16

Behmo, R., Marcombes, P., Dalalyan, A., and Prinet, V. (2010). Towards optimal naive bayes nearest neighbor. In *European Conference on Computer Vision (ECCV)*, pages 171–184. 37

Bellman, R. E. (1961). *Adaptive control processes - A guided tour*. Princeton University Press. 28, 67, 90

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(4):509–522. 14

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127. 41

Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy Layer-Wise Training of Deep Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 153–160. 41

Benois-Pineau, J., Precioso, F., and Cord, M. (2012). *Visual indexing and retrieval*. Springer Verlag. 11

Biederman, I. (1987). Recognition-by-Components: A Theory of Human Image Understanding. *Psychological Review*, 94(2):115–147. 3

Biederman, I. (1995). *An Invitation to Cognitive Science: Visual Cognition*, volume 2, chapter Visual Object Recognition, pages 121–165. MIT Press. 3

Binder, A., Samek, W., Kloft, M., Müller, C., Müller, K.-R., and Kawanabe, M. (2011). The joint submission of the tu berlin and fraunhofer first (tubfi) to the ImageCLEF 2011 photo annotation task. In *Cross-Language Evaluation Forum (CLEF Notebook Papers/Labs/Workshop)*. 53, 95, 96, 119

Boiman, O., Shechtman, E., and Irani, M. (2008). In Defense of Nearest-Neighbor Based Image Classification. In *Computer Vision and Pattern Recognition (CVPR)*. 37

Bordes, A. (2010). *New Algorithms for Large-Scale Support Vector Machines*. PhD thesis, Université Pierre et Marie Curie. 33

Bordes, A., Bottou, L., and Gallinari, P. (2009). SGD-QN: Careful quasi-newton stochastic gradient descent. *Journal of Machine Learning Research*, 10:1737–1754. 33

Bosch, A., Zisserman, A., and Muñoz, X. (2007). Image classification using random forests and ferns. In *International Conference on Computer Vision (ICCV)*, pages 1–8. 15, 36

Bosch, A., Zisserman, A., and noz, X. M. (2006). Scene classification via plsa. In *European Conference on Computer Vision (ECCV)*, pages 517–530. 18, 38

Bottou, L. (2007). Stochastic gradient descent examples on toy problems. 33

Bottou, L. and Bousquet, O. (2008). The tradeoffs of large scale learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 161–168. 33

Bouirouga, H., Fkihi, S. E., Jilbab, A., and Aboutajdine, D. (2012). Skin detection in pornographic videos using threshold technique. *Journal of Theoretical and Applied Information Technology*, 35(1):7–19. 107

Boureau, Y., Bach, F., LeCun, Y., and Ponce, J. (2010a). Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2559–2566. 6, 19, 20, 24, 25, 39, 63, 64, 72, 92, 101

Boureau, Y., Le Roux, N., Bach, F., Ponce, J., and LeCun, Y. (2011). Ask the locals: multi-way local pooling for image recognition. In *International Conference on Computer Vision (ICCV)*, pages 2651–2658. 26, 30, 72, 74

Boureau, Y.-L., Ponce, J., and LeCun, Y. (2010b). A theoretical analysis of feature pooling in visual recognition. In *International Conference on Machine Learning (ICML)*, pages 111–118. 7, 25

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24(2):123–140. 34, 35

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32. 35

Brown, G., Wyatt, J. L., Harris, R., and Yao, X. (2005). Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20. 34

Burghouts, G. J. and Geusebroek, J.-M. (2009). Performance evaluation of local colour invariants. *Computer Vision and Image Understanding (CVIU)*, 113(1):48–62. 18

Burl, M. C., Weber, M., and Perona, P. (1998). A probabilistic approach to object recognition using local photometry and global geometry. In *European Conference on Computer Vision (ECCV)*, pages 628–641. 42, 43

Carneiro, G. and Lowe, D. (2006). Sparse flexible models of local features. In *European Conference on Computer Vision (ECCV)*, pages 29–43. 43

Carson, C., Belongie, S., Greenspan, H., and Malik, J. (2002). Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 24(8):102–1038. 15

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27. 29, 33

Chatfield, K., Lempitsky, V., Vedaldi, A., and Zisserman, A. (2011). The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference (BMVC)*. 15, 26, 30, 57, 70, 84, 91, 92, 99, 100

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: deep neural networks with multitask learning. In *International Conference on Machine Learning (ICML)*, pages 160–167. 41

Cord, M. and Cunningham, P. (2008). *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*. Cognitive Technologies, Springer. 63

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3):273--297. 32

Costa, L. d. F. d. and Cesar Jr., R. M. (2000). *Shape Analysis and Classification: Theory and Practice*. CRC Press, Inc., 1 edition. 13

Cristianini, N. and Shawe-Taylor, J. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press. 32

Csurka, G., Bray, C., Dance, C., and Fan, L. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22. 20, 38

Cunningham, P. (2008). *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, chapter Dimension Reduction. Cognitive Technologies, Springer. 28

Cunningham, P., Cord, M., and Delany, S. J. (2008). *Machine Learning Techniques for Multimedia: Case Studies on Organization and Retrieval*, chapter Supervised Learning. Cognitive Technologies, Springer. 37

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 886–893. 14, 17, 74

Datta, R., Joshi, D., Li, J., and Wang, J. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40. 11

Del Bimbo, A. (1999). *Visual information retrieval*. Morgan Kaufmann Publishers Inc. 13

Deselaers, T., Pimenidis, L., and Ney, H. (2008). Bag-of-visual-words models for adult image classification and filtering. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4. 107, 108

Douze, M., Jégou, H., Sandhawalia, H., Amsaleg, L., and Schmid, C. (2009). Evaluation of gist descriptors for web-scale image search. In *ACM International Conference on Image and Video Retrieval (CIVR)*, pages 1–8. 15

Duda, R. O., Hart, P. E., and Stork, D. G. (2001). *Pattern Classification*. John Wiley and Sons, 2 edition. 20, 37, 38

Endeshaw, T., Garcia, J., and Jakobsson, A. (2008). Fast classification of indecent video by low complexity repetitive motion detection. In *IEEE Applied Imagery Pattern Recognition Workshop*, pages 1–7. 108

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2007). The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html. 56, 57, 58, 83, 85, 86, 99, 100

Faloutsos, C., Barber, R., Flickner, M., Hafner, J., Niblack, W., Petkovic, D., and Equitz, W. (1994). Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3-4):231–262. 13

Farquhar, J., Szedmak, S., Meng, H., , and Shawe-Taylor, J. (2005). Improving bag-of-keypoints image categorisation. Technical report, University of Southampton. 39

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative Model Based Vision)*. 58

Fei-Fei, L., Fergus, R., and Perona, P. (2007). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding (CVIU)*, 106(1):59–70. 29, 42, 43

Fei-Fei, L. and Perona, P. (2005). A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition (CVPR)*, pages 524–531. 38, 43

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1627–1645. 42, 43

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79. 43

Felzenszwalb, P. F., McAllester, D. A., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition (CVPR)*. 42

Feng, J., Ni, B., Tian, Q., and Yan, S. (2011). Geometric ℓ_p-norm feature pooling for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2609–2704. 25

Fergus, R. (2005). *Visual Object Category Recognition*. PhD thesis, University of Oxford. 42

Fergus, R., Perona, P., and Zisserman, A. (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Computer Vision and Pattern Recognition (CVPR)*, pages 264–271. 42, 43

Fergus, R., Perona, P., and Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 380–397. 43

Ferrari, V., Marín-Jiménez, M., and Zisserman, A. (2008). Progressive search space reduction for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*. 43

Fischler, M. A. and Elschlager, R. A. (1973). The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92. 42, 43

Fleck, M., Forsyth, D. A., and Bregler, C. (1996). Finding naked people. In *European Conference on Computer Vision (ECCV)*, pages 593–602. 108

Forsyth, D. A. and Fleck, M. M. (1996). Identifying nude pictures. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 103–108. 108

Forsyth, D. A. and Fleck, M. M. (1997). Body plans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 678–683. 108

Forsyth, D. A. and Fleck, M. M. (1999). Automatic detection of human nudes. *International Journal on Computer Vision (IJCV)*, 32(1):63–77. 108

Fournier, J., Cord, M., and Philipp-Foliguet, S. (2001). RETIN: A content-based image indexing and retrieval system. *Pattern Analysis and Applications Journal, Special issue on image indexation*, 4(2/3):153–173. 20

Freund, Y. and Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37. 35

Friedman, J., Hastie, T., , and Tibshirani, R. (2000). Additive logistic regression: a statistical view of boosting. In *Ann. Statist.* 35

Frome, A., Huber, D., Kolluri, R., and Bülow, T. (2004). Recognizing objects in range data using regional point descriptors. In *European Conference on Computer Vision (ECCV)*, pages 224–237. 14

Fukushima, K. and Miyake, S. (1982). Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognition*, 15(6):455–469. 40

Fulkerson, B., Vedaldi, A., and Soatto, S. (2008). Localizing objects with smart dictionaries. In *European Conference on Computer Vision (ECCV)*, pages 179–192. 38

Gao, S., Tsang, I. W.-H., Chia, L.-T., and Zhao, P. (2010). Local features are not lonely - laplacian sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3555–3561. 59

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *International Conference on Computer Vision (ICCV)*, pages 221–228. 35

Goh, H., Thome, N., Cord, M., and Lim, J.-H. (2012). Unsupervised and supervised visual codes with restricted boltzmann machines. In *European conference on Computer Vision (ECCV)*, pages 298–311. 39

Golub, G. and Van Loan, C. (1996). *Matrix Computations*. Johns Hopkins University Press, 3 edition. 29

Gonzalez, R. C. and Woods, R. E. (2006). *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc. 14

Gosselin, P., Cord, M., and Philipp-Foliguet, S. (2008). Combining visual dictionary, kernel-based similarity and learning strategy for image category retrieval. *Computer Vision and Image Understanding (CVIU)*, 3:403–417. 11

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision (ICCV)*, pages 1458–1465. 25

Grauman, K. and Leibe, B. (2011). *Visual Object Recognition. Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers. 44

Guillaumin, M., Verbeek, J., and Schmid, C. (2010). Multimodal semi-supervised learning for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 902–909. 33, 49, 93, 94

Guyon, I., Boser, B. E., and Vapnik, V. (1993). Automatic capacity tuning of very large vc-dimension classifiers. In *Advances in Neural Information Processing Systems (NIPS)*, pages 147–155. 32

Han, S. and Vasconcelos, N. (2010). Biologically plausible saliency mechanisms improve feedforward object recognition. *Vision Research*, 50(22):2295–2307. 40

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6):610–621. 13

Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151. 16

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computing*, 14(8):1771–1800. 41

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507. 41

Ho, T. K. (1995). Random decision forest. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 278–282. 35

Holub, A. and Perona, P. (2005). A discriminative framework for modelling object classes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 664–671. 42

Holub, A. D., Welling, M., and Perona, P. (2005). Combining generative models and fisher kernels for object recognition. In *International Conference on Computer Vision (ICCV)*, pages 136–143. 78

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441. 29

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundararajan, S. (2008). A dual coordinate descent method for large-scale linear svm. In *International Conference on Machine learning (ICML)*, pages 408–415. 33

Hu, M.-K. (1962). Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187. 14

Hu, W., Zuo, H., Wu, O., Chen, Y., Zhang, Z., and Suter, D. (2011). Recognition of adult images, videos, and web page bags. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 7S(1):28:1--28:24. 108

Huang, F.-J. and LeCun, Y. (2006). Large-scale learning with svm and convolutional nets for generic object categorization. In *Computer Vision and Pattern Recognition Conference (CVPR)*. 40

Huang, J., Ravi Kumar, S., Mitra, M., Zhu, W.-J., and Zabih, R. (1999). Spatial color indexing and applications. *International Journal of Computer Vision (IJCV)*, 35(3):245–268. 13

Hubel, D. and Wiesel, T. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3):574–591. 40

Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1):215–243. 40

Huiskes, M. and Lew, M. (2008). The MIR Flickr Retrieval Evaluation. In *ACM International Conference on Multimedia Information Retrieval (MIR)*, pages 39–43. 47, 48, 58, 83, 88, 89, 90, 93, 94

Huiskes, M. J., Thomee, B., and Lew, M. S. (2010). New trends and ideas in visual concept detection: The MIR Flickr Retrieval Evaluation Initiative. In *ACM International Conference on Multimedia Information Retrieval (MIR)*, pages 527–536. 49, 50, 93, 94

Jaakkola, T. and Haussler, D. (1998). Exploiting generative models in discriminative classifiers. In *In Advances in Neural Information Processing Systems*, pages 487–493. 26, 78

Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley & Sons, Inc. 84, 85, 87

Jansohn, C., Ulges, A., and Breuel, T. M. (2009). Detecting pornographic video content by combining image features with motion information. In *ACM International Conference on Multimedia*, pages 601–604. 109

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). What is the best multi-stage architecture for object recognition? In *International Conference on Computer Vision (ICCV)*, pages 2146–2153. 41

Jégou, H. and Chum, O. (2012). Negative evidences and co-occurrences in image retrieval: the benefit of PCA and whitening. In *European Conference on Computer Vision (ECCV)*. 29, 31

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010). Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3304–3311. 26, 29, 30, 38, 65, 75

Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., and Schmid, C. (2012). Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(9):1704–1716. 31, 70

Jia, Y., Huang, C., and Darrell, T. (2012). Beyond spatial pyramids: Receptive field learning for pooled image features. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3370–3377. 25

Jiang, Z., Lin, Z., and Davis, L. S. (2011). Learning a discriminative dictionary for sparse coding via label consistent K-SVD. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1697–1704. 39

Joachims, T. (2006). Training linear SVMs in linear time. In *International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 217–226. 33, 120

Jolliffe, I. (2002). *Principal Component Analysis*. Springer Verlag, 2 edition. 28

Jones, M. J. and Rehg, J. M. (2002). Statistical color models with application to skin detection. *International Journal of Computer Vision (IJCV)*, 46(1):81–96. 107, 108

Jurie, F. and Triggs, B. (2005). Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision (ICCV)*, pages 604–610. 15, 38

Kavukcuoglu, K., Ranzato, M., , and LeCun, Y. (2008). Fast inference in sparse coding algorithms with applications to object recognition. Technical report, Computational and Biological Learning Lab, Courant Institute, NYU. 41

Kavukcuoglu, K., Sermanet, P., Boureau, Y., Gregor, K., Mathieu, M., and LeCun, Y. (2010). Learning convolutional feature hierachies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS)*. 40

Ke, Y. and Sukthankar, R. (2004). Pca-sift: a more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition (CVPR)*, pages 506–513. 18

Kim, J. and Grauman, K. (2011). Boundary preserving dense local regions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1553–1560. 15

Kohonen, T. (1988). Self-organized formation of topologically correct feature maps. In *Neurocomputing: foundations of research*, pages 509–521. MIT Press. 20

Koniusz, P. and Mikolajczyk, K. (2011). Spatial coordinate coding to reduce histogram representations, dominant angle and colour pyramid match. In *International Conference on Image Processing (ICIP)*, pages 661–664. 25, 60

Koniusz, P., Yan, F., and Mikolajczyk, K. (2013). Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer Vision and Image Understanding (CVIU), Special Issue on Visual Concept Detection*, 117(5):479–492. 22

Krapac, J. (2011). *Image Representations for Ranking and Classification*. PhD thesis, Caen University. 77

Krapac, J., Verbeeky, J., and Jurie, F. (2011). Modeling spatial layout with fisher vectors for image categorization. In *International Conference on Computer Vision (ICCV)*, pages 1487–1494. 8, 26, 27, 29, 33, 38, 58, 65, 80, 84, 98, 99, 101, 102, 123

Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1106–1114. 40, 41

Kumar, M. P., Zisserman, A., and Torr, P. H. S. (2009). Efficient discriminative learning of parts-based models. In *International Conference on Computer Vision (ICCV)*, pages 552–559. 42

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207. 34

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision (IJCV)*, 64(2-3):107–123. 109

Larlus, D. and Jurie, F. (2009). Latent mixture vocabularies for object categorization and segmentation. *Image and Vision Computing*, 27(5):523–534. 39

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *International Conference on Machine Learning (ICML)*, pages 473–480. 41

Lazebnik, S. and Raginsky, M. (2009). Supervised learning of quantizer codebooks by information loss minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(7):1294–1309. 39

Lazebnik, S., Schmid, C., and Ponce, J. (2005). A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(8):1265–1278. 18

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2169–2178. 25, 26, 33, 38, 58, 66, 83, 84, 86, 101, 102

Le, D.-D. and Satoh, S. (2011). Nii, japan at ImageCLEF 2011 photo annotation task. In *Cross-Language Evaluation Forum (CLEF Notebook Papers/Labs/Workshop)*. 95, 98

Lechervy, A., Gosselin, P.-H., and Precioso, F. (2012). Boosting kernel combination for multi-class image categorization. In *International Conference on Image Processing (ICIP)*. 35, 60

LeCun, Y., Boser, B., Denker, J. S., Howard, R. E., Habbard, W., Jackel, L. D., and Henderson, D. (1990). Advances in neural information processing systems (nips). In *Handwritten digit recognition with a back-propagation network*, pages 396–404. 40

LeCun, Y., Huang, F. J., and Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Computer Vision and Pattern Recognition (CVPR)*, pages 97–104. 40

LeCun, Y., Kavukvuoglu, K., and Farabet, C. (2010). Convolutional networks and applications in vision. In *International Symposium on Circuits and Systems (ISCAS)*. 40

Lee, H., Chaitanya, E., and Ng, A. (2008). Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems (NIPS)*, pages 873–880. 41

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009a). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning (ICML)*, pages 609–616. 41

Lee, J.-S., Kuo, Y.-M., Chung, P.-C., and Chen, E.-L. (2007). Naked image detection based on adaptive and extensible skin color model. *Pattern Recognition*, 40(8):2261–2270. 108

Lee, S., Shim, W., and Kim, S. (2009b). Hierarchical system for objectionable video detection. *IEEE Transactions on Consumer Electronics*, 55(2):677–684. 107

Leibe, B., Leonardis, A., and Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in Computer Vision*. 44

Leibe, B., Leonardis, A., and Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision (IJCV)*, 77(1-3):259–289. 42, 44

Leibe, B. and Schiele, B. (2003). Interleaved object categorization and segmentation. In *British Machine Vision Conference*, pages 759–768. 15, 38

Leistner, C., Saffari, A., Santner, J., and Bischof, H. (2009). Semi-supervised random forests. In *International Conference on Computer Vision (ICCV)*, pages 506–513. 36

Lepetit, V. and Fua, P. (2006). Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1465–1479. 36

Lew, M., Sebe, N., Djeraba, C., and Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2. 11

Li, B., Xiao, R., Li, Z., Cai, R., Lu, B.-L., and 0001, L. Z. (2011). Rank-sift: Learning to rank repeatable local interest points. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1737–1744. 18

Li, J. and Allinson, N. M. (2008). A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10-12):1771–1787. 17

Linde, Y., Buzo, A., and Gray, R. M. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28:84–95. 20

Lindeberg, T. (1998). Feature detection with automatic scale selection. *International Journal of Computer Vision (IJCV)*, 30(2):79–116. 17

Liu, J. and Shah, M. (2007). Scene Modeling Using Co-Clustering. In *International Conference on Computer Vision (ICCV)*, pages 1–7. 38

Liu, J., Yang, Y., and Shah, M. (2009). Learning semantic visual vocabularies using diffusion distance. In *Computer Vision and Pattern Recognition (CVPR)*, pages 461–468. 39

Liu, L., Wang, L., and Liu, X. (2011a). In defense of soft-assignment coding. In *International Conference on Computer Vision (ICCV)*, pages 2486–2493. 7, 24, 25, 30, 45, 70, 72

Liu, N., Dellandrea, E., Chen, L., Trus, A., Zhu, C., Zhang, Y., Bichot, C.-E., Bres, S., and Tellez, B. (2012a). LIRIS-Imagine at ImageCLEF 2012 Photo Annotation task. In *Working Notes of the 2012 Conference and Labs of the Evaluation Forum*. 54, 103, 104, 119, 120

Liu, N., Dellandrea, E., Zhu, C., Bichot, C.-E., and Chen, L. (2012b). A selective weighted late fusion for visual concept recognition. In *ECCV 2012 Workshop on Information fusion in Computer Vision for Concept Recognition*. 103

Liu, Y., Wang, X., Zhang, Y., and Tang, S. (2011b). Fusing audio-words with visual features for pornographic video detection. In *International Conference on Trust, Security and Privacy in Computing and Communications (TRUSTCOM)*, pages 1488–1493. 109

Lopes, A., Avila, S., Peixoto, A., Oliveira, R., Coelho, M., and de A. Araújo, A. (2009a). Nude detection in video using bag-of-visual-features. In *Brazilian Symposium on Computer Graphics and Image (SIBGRAPI)*, pages 224–231. 108

Lopes, A., Avila, S., Peixoto, A., Oliveira, R., and de A. Araújo, A. (2009b). A bag-of-features approach based on hue-sift descriptor for nude detection. In *European Signal Processing Conference (EUSIPCO)*, pages 1552–1556. 108

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60:91–110. 6, 17, 18, 19, 63, 70, 74

Lowe, D. (2012). Local naive bayes nearest neighbor for image classification. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3650–3656. 37

Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, pages 1150–1157. 18

Lu, L., Toyama, K., and Hager, G. D. (2005). A two level approach for scene recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 688–695. 34

Ma, W. Y. and Manjunath, B. S. (1999). NeTra: A toolbox for navigating large image databases. *ACM Multimedia Systems*, 7(3):184–198. 20

Mairal, J., Bach, F., Ponce, J., and Sapiro, G. (2010). Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60. 24

Mairal, J., Bach, F., Ponce, J., Sapiro, G., and Zisserman, A. (2008). Supervised dictionary learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1033–1040. 39

Manjunath, B. S. and Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 18(8):837–842. 13

Mbanya, E., Gerke, S., Hentschel, C., and Ndjiki-Nya, P. (2011). Sample selection, category specific features and reasoning. In *Cross-Language Evaluation Forum (CLEF Notebook Papers/Labs/Workshop)*. 95, 98

Mikolajczy, K. and Schmid, C. (2004). Scale and affine interest point detectors. *International Journal of Computer Vision (IJCV)*, 60(1):63–86. 17

Mikolajczyk, K. and Schmid, C. (2002). An affine invariant interest point detector. In *European Conference Computer Vision (ECCV)*. 17

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630. 17, 18

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Gool, L. V. (2005). A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1/2):43–72. 17

Müller, H., Clough, P., Deselaers, T., and Caputo, B. (2010). *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*. Springer Publishing Company, Incorporated, 1st edition. 50

Mokhtarian, F. (1995). Silhouette-based isolated object recognition through curvature scale space. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(5):539–544. 14

Moosmann, F., Nowak, E., and Jurie, F. (2008). Randomized clustering forests for image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(9):1632–1646. 39

Moosmann, F., Triggs, B., and Jurie, F. (2006). Fast discriminative visual codebooks using randomized clustering forests. In *Advances in Neural Information Processing Systems (NIPS)*, pages 985–992. 36

Mutch, J. and Lowe, D. G. (2008). Object class recognition and localization using sparse features with limited receptive fields. *International Journal of Computer Vision (IJCV)*, 80(1):45–57. 40

Nadernejad, E., Sharifzadeh, S., and Hassanpour, H. (2008). Edge detection techniques: Evaluations and comparisons. *Applied Mathematical Sciences*, 2(31):1507–1520. 14

Nebauer, C. (1998). Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks*, 9(4):685–695. 40

Negrel, R., Picard, D., P.H, and Gosselin (2012). Using spatial pyramids with compacted vlat for image categorization. In *International Conference on Pattern Recognition (ICPR)*. 26, 30

Nilsback, M.-E. and Zisserman, A. (2006). A Visual Vocabulary for Flower Classification. In *Computer Vision and Pattern Recognition (CVPR)*. 59, 60, 66, 68

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2161–2168. 30, 38

Nowak, E., Jurie, F., and Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision (ECCV)*. 38

Nowak, S., Nagel, K., and Liebetrau, J. (2011). The CLEF 2011 Photo Annotation and Concept-based Retrieval Tasks. In *Cross-Language Evaluation Forum (CLEF Notebook Papers/Labs/Workshop)*. 3, 50, 51, 58, 83, 95, 96, 119

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision (IJCV)*, 42(3):145–175. 14, 58

Oliveira, G. L., Nascimento, E. R., Vieira, A. W., and Campos, M. F. M. (2012). Sparse spatial coding: A novel approach for efficient and accurate object recognition. In *International Conference on Robotics and Automation (ICRA)*, pages 2592–2598. 24

Opelt, A., Fussenegger, M., Pinz, A., and Auer, P. (2004). Weak hypotheses and boosting for generic object detection and recognition. In *European Conference on Computer Vision (ECCV)*, pages 71–84. 35

Ott, P. and Everingham, M. (2011). Shared parts for deformable part-based models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1513–1520. 42, 43

Parikh, D., Zitnick, C. L., and Chen, T. (2009). Unsupervised learning of hierarchical spatial structures in images. In *Computer Vision and Pattern Recognition (CVPR)*. 38

Parizi, S., Oberlin, J., and Felzenszwalb, P. (2012). Reconfigurable models for scene recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2775–2782. 44

Pass, G. and Zabih, R. (1996). Histogram refinement for content-based image retrieval. In *IEEE Workshop on Applications of Computer Vision (WACV)*. 13

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2:559–572. 29

Penatti, O. A. B., Valle, E., and da S. Torres, R. (2011). Encoding spatial arrangement of visual words. In *Conference on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (CIARP)*, pages 240–247. 26, 38

Perronnin, F. and Dance, C. (2007). Fisher kernels on visual vocabularies for image categorization. In *Computer Vision and Pattern Recognition (CVPR)*. 27, 38, 78, 80

Perronnin, F., Dance, C., Csurka, G., and Bressan, M. (2006). Adapted vocabularies for generic visual categorization. In *European Conference on Computer Vision (ECCV)*, pages 464–475. 39

Perronnin, F., Liu, Y., Sánchez, J., and Poirier, H. (2010a). Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3384–3391. 29

Perronnin, F., Sánchez, J., and Liu, Y. (2010b). Large-scale image categorization with explicit data embedding. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2297–2304. 70, 120

Perronnin, F., Sánchez, J., and Mensink, T. (2010c). Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision (ECCV)*, pages 143–156. 8, 26, 27, 28, 29, 30, 31, 33, 38, 45, 65, 69, 70, 72, 75, 83, 84, 91, 93, 94, 95, 96, 99, 100, 101

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2008). Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition (CVPR)*. 22

Picard, D. and Gosselin, P. (2011). Improving image similarity with vectors of locally aggregated tensors. In *International Conference on Image Processing (ICIP)*, pages 669–672. 26, 30, 33, 38, 45, 65

Picard, D., Thome, N., and Cord, M. (2010). An efficient system for combining complementary kernels in complex visual categorization tasks. In *International Conference on Image Processing (ICIP)*, pages 3877–3880. 98

Picard, D., Thome, N., and Cord, M. (2012). Learning geometric combinations of gaussian kernels with alternating quasi-newton algorithm. In *European Symposium on Artificial Neural Networks (ESANN)*. 98

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C. J. C., and Smola, A. J., editors, *Advances in kernel methods*, pages 185–208. MIT Press. 33

Prewitt, J. (1970). *Picture processing and Psychopictorics*, chapter Object Enhancement and Extraction. Academic Press, Inc. 14

Quelhas, P., Monay, F., Odobez, J.-M., Gatica-Perez, D., and Tuytelaars, T. (2007). A thousand words in a scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(9):1575–1589. 38

Ramanan, D., Forsyth, D. A., and Zisserman, A. (2007). Tracking people by learning their appearance. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(1):65–81. 43

Ramanan, D. and Sminchisescu, C. (2006). Training deformable models for localization. In *Computer Vision and Pattern Recognition (CVPR)*, pages 206–213. 42

Ranzato, M., Boureau, Y.-L., and LeCun, Y. (2007a). Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems (NIPS)*. 41

Ranzato, M., Huang, F., Boureau, Y., and LeCun, Y. (2007b). Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *Computer Vision and Pattern Recognition Conference (CVPR)*. 40

Rea, N., Lacey, G., Lambe, C., and Dahyot, R. (2006). Multimodal periodicity analysis for illicit content detection in videos. In *European Conference on Visual Media Production (CVMP)*, pages 106–114. 109

Ries, C. and Lienhart, R. (2012). A survey on visual adult image recognition. *Multimedia Tools and Applications*, pages 1–28. 108

Riesenhuber, M. and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2:1019–1025. 40

Rocha, A., Hauagge, D. C., Wainer, J., and Goldenstein, S. (2008). Automatic produce classification from images using color, texture and appearance cues. In *Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pages 3--10. 35

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39. 33

Rowley, H. A., Jing, Y., and Baluja, S. (2006). Large scale image-based adult-content filtering. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 290–296. 107, 108

Russakovsky, O., Lin, Y., Yu, K., and Fei-Fei, L. (2012). Object-centric spatial pooling for image classification. In *European Conference on Computer Vision (ECCV)*, pages 1–15. 26

Safadi, B. (2012). *Indexation sémantique des images et des vidéos par apprentissage actif*. PhD thesis, Grenoble University. 31

Saffari, A., Grabner, H., and Bischof, H. (2008). SERBoost: Semi-supervised boosting with expectation regularization. In *European Conference on Computer Vision*, pages 588–601. 35

Sánchez, J., Perronnin, F., and deCampos, T. E. (2012). Modeling the spatial layout of images beyond spatial pyramids. *Pattern Recognition Letters*. 26, 29, 33, 38, 99, 100, 120

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning*, 5(2):197–227. 35

Schapire, R. E., Freund, Y., Bartlett, P., and Lee, W. S. (1998). Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. *The Annals of Statistics*, 26(5):1651–1686. 35

Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *International Journal of Computer Vision (IJCV)*, 37(2):151–172. 17

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319. 29

Scholkopf, B. and Smola, A. J. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press. 32

Sebe, N., Cohen, I., Garg, A., and Huang, T. (2005). *Machine Learning in Computer Vision*. Springer Verlag. 63

Sermanet, P., Chintala, S., and LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. In *International Conference on Pattern Recognition (ICPR)*. 40

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007). Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(3):411–426. 40

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *International Conference on Machine Learning (ICML)*, pages 807–814. 33

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press. 29, 79

Short, M. B., Black, L., Smith, A. H., Wetterneck., C. T., and Wells, D. E. (2012). A review of internet pornography use research: methodology and content from the past 10 years. *Cyberpsychology, Behavior, and Social Networking*, 15(1):13–23. 107, 115

Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *Computer Vision and Pattern Recognition (CVPR)*. 36

Sivic, J. and Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision (ICCV)*, volume 2. 5, 18, 19, 20, 29, 38, 63, 64, 66, 84, 88, 91, 93, 95, 99, 101, 114

Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 22(12):1349–1380. 2, 11

Smolensky, P. (1986). Information processing in dynamical systems: foundations of harmony theory. In Rumelhart, D. E., McClelland, J. L., and PDP Research Group, C., editors, *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*, pages 194–281. MIT Press Cambridge. 41

Sobel, I. E. (1970). *Camera models and machine perception*. PhD thesis, Stanford University. 14

Steel, C. (2012). The mask-sift cascading classifier for pornography detection. In *World Congress on Internet Security (WorldCIS)*, pages 139–142. 108

Stehling, R. O., Nascimento, M. A., and Falcão, A. X. (2002). A compact and efficient image retrieval approach based on border/interior pixel classification. In *International Conference on Information and Knowledge Management (CIKM)*, pages 102–109. 13

Stricker, M. and Orengo, M. (1995). Similarity of color images. In *Storage and Retrieval for Image and Video Databases*, pages 381–392. 13

Su, Y. and Jurie, F. (2011). Semantic contexts and fisher vectors for the ImageCLEF 2011 photo annotation task. In *Cross-Language Evaluation Forum (CLEF Notebook Papers/Labs/Workshop)*. 53, 95, 96, 98

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision (IJCV)*, 7(1):11–32. 13, 14

Tamura, H., Mori, S., and Yamawaki, T. (1978). Textural Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man and Cybernetics*, 6(8):460–473. 13

Taylor, G., Fergus, R., LeCun, Y., and Bregler, C. (2010). Convolutional learning of spatio-temporal features. In *European Conference on Computer Vision (ECCV)*. 40

Thériault, C., Thome, N., and Cord, M. (2011). HMAX-S: Deep scale representation for biologically inspired image categorization. In *International Conference on Image Processing (ICIP)*, pages 1261–1264. 40

Thériault, C., Thome, N., and Cord, M. (2012). Extended coding and pooling in the HMAX model. *IEEE Transactions on Image Processing*. 40

Thomee, B. and Popescu, A. (2012). Overview of the ImageCLEF 2012 Flickr Photo Annotation and Retrieval Task. In Forner, P., Karlgren, J., and Womser-Hacker, C., editors, *CLEF (Online Working Notes/Labs/Workshop)*. 53, 54, 58, 83, 103, 104, 119

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature*, 381:520–522. 3

Tola, E., Lepetit, V., and Fua, P. (2010). DAISY: An Efficient Dense Descriptor Applied to Wide Baseline Stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(5):815–830. 17

Tong, X., Duan, L., Xu, C., Tian, Q., Hanqing, L., Wang, J., , and Jin, J. (2005). Periodicity detection of local motion. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 650–653. 108

Torralba, A., Fergus, A., and Weiss, Y. (2008). Small codes and large databases for recognition. In *Computer Vision and Pattern Recognition (CVPR)*. 14

Torralba, A., Murphy, K. P., Freeman, W. T., and Rubin, M. A. (2003). Context-based vision system for place and object recognition. In *IEEE International Conference on Computer Vision (ICCV)*. 14

Trémeau, A., Tominaga, S., and Plataniotis, K. N. (2008). Color in image and video processing: most recent trends and future research directions. *Journal on Image and Video Processing*, 2008(7):1–26. 13

Tsai, C.-F. (2005). Training support vector machines based on stacked generalization for image classification. *Journal Neurocomputing*, 64:497–503. 36

Tsuda, K., Kin, T., and Asai, K. (2002). Marginalized kernels for biological sequences. In *International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 268–275. 78

Tuceryan, M. and Jain, A. K. (2000). *Handbook of Pattern Recognition and Computer Vision*, chapter Texture Analysis. World Scientific Publishing Co., Inc., 2 edition. 13

Tuytelaars, T. (2010). Dense interest points. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2281–2288. 15, 16

Tuytelaars, T., Fritz, M., Saenko, K., and Darrell, T. (2011). The nbnn kernel. In *International Conference on Computer Vision (ICCV)*. 38

Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280. 14, 15, 16, 17

Ulges, A., Schulze, C., Borth, D., and Stahl, A. (2012). Pornography detection in video benefits (a lot) from a multi-modal approach. In *International Workshop on Audio and Multimedia Methods for Large-Scale Video Analysis*, pages 21–26. 109

Ulges, A. and Stahl, A. (2011). Automatic detection of child pornography using color visual words. In *International Conference on Multimedia Retrieval (ICME)*, pages 1–6. 108

Ullah, M. M. and Laptev, I. (2012). Actlets: A novel local representation for human action recognition in video. In *International Conference on Image Processing (ICIP)*, pages 777–780. 116

Ushiku, Y., Muraoka, H., Inaba, S., Fujisawa, T., Yasumoto, K., Gunji, N., Higuchi, T., Hara, Y., Harada, T., and Kuniyoshi, Y. (2012). ISI at ImageCLEF 2012: Scalable System for Image Annotation. In *Working Notes of the 2012 Conference and Labs of the Evaluation Forum*. 54, 103, 104

Valle, E., Avila, S., da Luz Jr., A., de Souza, F., Coelho, M., and de A. Araújo, A. (2012). Content-based filtering for video sharing social networks. In *Brazilian Symposium on Information and Computer System Security*. 109

van de Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(9):1582–1596. 18, 66, 111, 113

van de Sande, K. E. A. and Snoek, C. G. M. (2011). The university of amsterdam's concept detection system at ImageCLEF 2011. In *Cross-Language Evaluation Forum (CLEF Notebook Papers/Labs/Workshop)*. 53, 95, 98

van de Weijer, J. and Schmid, C. (2006). Coloring local feature extraction. In *European Conference on Computer Vision (ECCV)*, pages 334–348. 18

van Gemert, J., Veenman, C., Smeulders, A., and Geusebroek, J.-M. (2010). Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32:1271–1283. 21, 22, 33, 45, 57, 72

van Gemert, J. C., Geusebroek, J.-M., Veenman, C. J., and Smeulders, A. W. (2008). Kernel codebooks for scene categorization. In *European Conference on Computer Vision (ECCV)*, pages 696–709. 22

van Gemert, J. C., Geusebroek, J.-M., Veenman, C. J., Snoek, C. G. M., and Smeulders, A. W. M. (2006). Robust Scene Categorization by Learning Image Statistics in Context. In *CVPR Workshop on Semantic Learning Applications in Multimedia*. 39

Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons. 32

Vapnik, V. N. and Lerner, A. (1963). Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780. 32

Vedaldi, A. and Fulkerson, B. (2010). VLFeat - An open and portable library of computer vision algorithms. In *ACM International Conference on Multimedia*. 84

Vedaldi, A., Gulshan, V., Varma, M., and Zisserman, A. (2009). Multiple kernels for object detection. In *International Conference on Computer Vision (ICCV)*, pages 606–613. 15, 119

Vedaldi, A. and Zisserman, A. (2012). Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(3). 30, 33, 70, 120

Viitaniemi, V. and Laaksonen, J. (2008). Experiments on selection of codebooks for local image feature histograms. In *International Conference on Visual Information Systems: Web-Based Visual Information Search and Management*, pages 126–137. 38

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. (2008). Extracting and composing robust features with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, pages 1096–1103. 41

Vogel, J. and Schiele, B. (2007). Semantic modeling of natural scenes for content-based image retrieval. *International Journal of Computer Vision (IJCV)*, 72(2):133–157. 39

Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176. 116

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367. 24, 72, 99

Weber, M., Welling, M., , and Perona, P. (2000). Towards Automatic Discovery of Object Categories. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2101–2108. 42, 43

Weston, J., Ratle, F., and Collobert, R. (2008). Deep learning via semi-supervised embedding. In *International Conference on Machine Learning (ICML)*, pages 1168–1175. 41

Willamowski, J., Arregui, D., Csurka, G., Dance, C. R., and Fan, L. (2004). Categorizing nine visual classes using local appearance descriptors. In *International Conference on Pattern Recognition (ICPR)*. 38

Williams, C. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems (NIPS)*, pages 682–688. 33, 120

Winn, J. and Criminisi, A. (2006). Object class recognition at a glance. In *Computer Vision and Pattern Recognition (CVPR)*. 36

Winn, J., Criminisi, A., and Minka, T. (2005). Object categorization by learned universal visual dictionary. In *International Conference on Computer Vision (ICCV)*, pages 1800–1807. 39

Wolf, L. and Martin, I. (2005). Robust boosting for learning from few examples. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 359–364. 35

Wolpert, D. H. (1992). Stacked generalization. *Neural Networks*, 5(5). 36

Xioufis, E. S., Tsoumakas, G., and Vlahavas, I. (2012). MLKD's Participation at Imageof the 2012 Conference and Labs of the Evaluation Forum Photo Annotation and Concept-based Retrieval Tasks. In *Working Notes of the 2012 Conference and Labs of the Evaluation Forum*. 54, 103, 104

Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *ACM International Conference on Multimedia Information Retrieval (MIR)*, pages 197–206. 30

Yang, J., Li, Y., Tian, Y., Duan, L., and Gao, W. (2009a). Group sensitive multiple kernel learning for object categorization. In *International Conference on Computer Vision (ICCV)*. 57

Yang, J., Yu, K., Gong, Y., and Huang, T. (2009b). Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801. 24, 25, 33, 72, 74, 92, 101, 102

Yang, M., Kpalma, K., and Ronsin, J. (2008). A Survey of Shape Feature Extraction Techniques. *Pattern Recognition*, pages 43–90. 14

Yang, Y. and Ramanan, D. (2011). Articulated pose estimation with flexible mixtures-of-parts. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1385–1392. 43

Yao, B., Khosla, A., and Fei-Fei, L. (2011). Combining randomization and discrimination for fine-grained image categorization. In *Computer Vision and Pattern Recognition (CVPR)*. 36

Yu, K., Zhang, T., and Gong, Y. (2009). Nonlinear learning using local coordinate coding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2223–2231. 24

Zhang, D. and Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19. 13

Zhang, H., Berg, A. C., Maire, M., and Malik, J. (2006). Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2126–2136. 37

Zhang, W. and Dietterich, T. G. (2008). Learning visual dictionaries and decision lists for object recognition. In *International Conference on Pattern Recognition (ICPR)*, pages 1–4. 35

Zheng, H. and Daoudi, M. (2004). Blocking adult images based on statistical skin detection. *Electronic Letters on Computer Vision and Image Analysis*, 4(2). 107

Zhou, X., Yu, K., Zhang, T., and Huang, T. (2010). Image classification using super-vector coding of local image descriptors. In *European Conference on Computer Vision (ECCV)*, pages 141–154. 8, 26, 29, 30, 33, 45, 57, 65, 69, 75, 99

Zhu, L., Chen, Y., Yuille, A. L., and Freeman, W. T. (2010). Latent hierarchical structural learning for object detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1062–1069. 42

Ziou, D. and Tabbone, S. (1998). Edge Detection Techniques - An Overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559. 14

Znaidia, A., Shabou, A., Popescu, A., le Borgne, H., and Hudelot, C. (2012). Multimodal feature generation framework for semantic image classification. In *International Conference on Multimedia Retrieval (ICMR)*, pages 1–8. 36, 57, 100

Zuo, H., Hu, W., and Wu, O. (2010). Patch-based skin color detection and its application to pornography image filtering. In *International Conference on World Wide Web (WWW)*, pages 1227–1228. 107, 108