REPRESENTAÇÕES DE CARACTERÍSTICAS VISUAIS DE BAIXO CUSTO PARA RECUPERAÇÃO DE IMAGENS

RAMON FIGUEIREDO PESSOA

REPRESENTAÇÕES DE CARACTERÍSTICAS VISUAIS DE BAIXO CUSTO PARA RECUPERAÇÃO DE IMAGENS

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Jefersson Alex dos Santos
Coorientador: William Robson Schwartz

Belo Horizonte
Dezembro de 2015

RAMON FIGUEIREDO PESSOA

LOW-COST VISUAL FEATURE REPRESENTATIONS FOR IMAGE RETRIEVAL

Dissertation presented to the Graduate Program in Computer Science of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Master in Computer Science.

Advisor: Jefersson Alex dos Santos
Co-Advisor: William Robson Schwartz

Belo Horizonte
December 2015

© 2015, Ramon Figueiredo Pessoa.
Todos os direitos reservados.

Ficha catalografica elaborada pela Biblioteca do ICEx – UFMG

Pessoa, Ramon Figueiredo

P475l Low-Cost Visual Feature Representations for Image Retrieval / Ramon Figueiredo Pessoa. — Belo Horizonte, 2015
xxxiii, 144 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais - Departamento de Ciência da Computação

Orientador: Jefersson Alex dos Santos
Coorientador: William Robson Schwartz

1. Computação — Teses, 1. Visão Computacional — Teses, 3. Reconhecimentos de Padrões — Teses, 4. Compressão de Imagens — Teses, 5. Processamento Digital de Imagens — Tecnicas Digitais — Teses, 6. Sistemas de Recuperação da Informação — Teses, 7. Images — Interpretação — Teses. I. Orientador. II. Coorientador. III. Título

CDU 519.6*84(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Low-cost visual feature representations for image retrieval

RAMON FIGUEIREDO PESSOA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. JEFERSSON ALEX DOS SANTOS - Orientador
Departamento de Ciência da Computação - UFMG

PROF. WILLIAM ROBSON SCHWARTZ - Coorientador
Departamento de Ciência da Computação - UFMG

PROF. ALEXEI MANSO CORREA MACHADO
Departamento de Ciência da Computação - PUCMG

PROF. DANIEL CARLOS GUIMARÃES PEDRONETTE
Departamento de Estatística, Matemática - UNESP

PROF. ERICKSON RANGEL DO NASCIMENTO
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 18 de dezembro de 2015.

To my dear parents, Selma Pereira Figueiredo and Jorge Sergio Fernandes Pessoa (in memory).

Acknowledgments

First of all, I wish to acknowledge God who gave me strength to get where I am. "Because Thou hast been my help, therefore in the shadow of thy wings I will rejoice." (Psalm 63:7).

I would like to express my very great appreciation to my advisor Professor Jefersson Alex dos Santos and my co-advisor William Robson Schwartz. They made possible to carry out this work with discussions, valuable and constructive suggestions and fundamental ideas. By patience, learning and scientific maturity. In particular Professor Jefersson by his example advisor, the opportunities offered through the master degree, friendship, advice. He became a source of encouragement.

I would also like to thank Professor Arnaldo de Albuquerque Araujo for the opportunity given to be part of Núcleo de Processamento Digital de Imagens/UFMG and initial guidance in my master's degree.

Many thanks go to my girlfriend Cecília for her continuous support and love, companionship, prayers and efforts on behalf of my studies. My mother, Selma, for her love, affection and prayers. My family for the attention and trust.

My grateful thanks to colleagues and friends of the research groups PATREO/UFMG (Pattern Recognition and Earth Observation), SSIG/UFMG (Smart Surveillance Interest Group), NPDI/UFMG (Núcleo de Processamento Digital de Imagens): Waner de Oliveira Miranda, Carlos Antônio Caetano Júnior, Cássio Elias dos Santos Júnior, Edemir Ferreira de A. Junior, Keiller Nogueira, Artur Jordão Lima Correia, Érico Marco Dias Alves Pereira, Antônio Carlos Nazaré Júnior, Raphael Felipe de Carvalho Prates, Ricardo Barbosa Kloss, Rensso Victor Hugo Mora Colque, Victor Hugo Cunha de Melo, Gabriel Resende Gonçalves, Samira Santos da Silva, Jéssica Sena de Souza, Henrique Dias e César Augusto Moura Ferreira. Thank you for friendship, joy, relaxation and scientific experience. In these groups, I met great people, friends and researchers.

I could not fail to thank Carlos Caetano, Waner Miranda, Cássio Elias, Érico Marco, Ricardo Kloss, Sandra Avila, Otávio Penatti, Keiller Nogueira, Antônio Nazaré

and Raphael Prates by exchanging experiences and for assisting me in different parts, but fundamental in the development of this work.

This work was partially financed by Brazilian National Research Council (CNPq), Minas Gerais Research Foundation (FAPEMIG) and Coordination for the Improvement of Higher Education Personnel (CAPES).

Finally, I would like to thank everyone who contributed directly or indirectly in carrying out this work. Thank you very much!

"O mundo está nas mãos daqueles que tem coragem de sonhar, e correr o risco de viver seus sonhos. É justamente a possibilidade de realizar um sonho que torna a vida interessante."
(Paulo Coelho)

Resumo

Busca por Conteúdo Visual em dispositivos Móveis (BCVM) é uma nova área de pesquisa em Recuperação de Imagem por Conteúdo (RIC), que oferece os serviços de busca e recuperação de informação visual especificamente para dispositivos móveis. Os principais desafios em Busca por Conteúdo Visual em dispositivos Móveis (BCVM) incluem variações nas condições de captura de imagem, como iluminação diferente, mudanças de escala e ângulo de visão, limitações da bateria e alto custo de rede incorridos pela transmissão de dados.

O objetivo principal deste trabalho é a comparação de técnicas eficientes e eficazes para extração de características (*features*) em dispositivos móveis a fim de recuperar imagens principalmente em *smartphones*. Alcançamos nosso objetivo comparando e propondo técnicas para compressão de vetor de característica de imagens e representação de nível médio (*bag of words*). Algumas abordagens reduzem o consumo de energia em dispositivos móveis porque elas enviam vetores de características mais compactos a serem processadas no lado do servidor. Uma série de experimentos também foram realizados para avaliar aspectos de eficácia, eficiência e compacidade de características extraídas de imagens com o objetivo de realizar recuperação de imagens por conteúdo em dispositivos móveis. Neste caso, o usuário decide a melhor configuração considerando o triplo *trade-off* sobre eficácia, eficiência, e compacidade de características visuais.

Desse modo, abordamos duas questões de pesquisa, a fim de investigar e propor soluções efetivas para a recuperação de imagens em dispositivos móveis: 1) representação de baixo custo para a recuperação de imagens por conteúdo visual em dispositivos móveis e 2) extração de características visuais com informação espacial.

Em primeiro lugar, analisamos o uso de descritores binários usando representação de nível médio e descritores globais (cor, textura e forma) no contexto de recuperação de imagem em dispositivos móveis, bem como o uso de técnicas de compressão de características de imagem. Nós testamos vinte representações de nível médio de descritores de binários ("cinco descritores binários" × "quatro estratégias de bag of words": os

descritores BinBoost, BRIEF, BRISK, FREAK, ORB com *bag of words* usando *hard assignment* com *average pooling* ou *bag of words* usando *hard assignment* com *maximum pooling* ou *bag of words* usando *soft assignment* com *average pooling* ou *bag of words* usando *soft assignment* com *maximum pooling*), dez descritores de cor, cinco descritores de textura e dois descritores de forma. Nós também analisamos o impacto de usar amostragem densa e amostragem esparsa (*keypoints*) para calcular descritores usando *bag of words* (a amostragem densa é a melhor opção).

A segunda questão de pesquisa refere-se a investigação do problema de extrair informações espaciais de imagens para melhorar a qualidade da representação de imagem em dispositivos móveis, que podem ser cruciais para distinguir tipos de objetos e cenas. Os métodos tradicionais de agrupamento (*bag of words*) geralmente descartam a configuração espacial na imagem. Nós propomos duas abordagens de *spatial bag of visual words* chamadas BOBGrid (spatial Bag Of BIC Grid) e BOBSlic (spatial Bag Of Slic) e comparamos elas com o nosso baseline de *spatial bag of visual words* chamado WSA (visual Word Spatial Arrangement) e com uma melhoria do *bag of visual words* tradicional chamada BOSSANova (Bag Of Statistical Sampling Analysis).

Os experimentos realizados indicam que os descritores BIC (Border/Interior Pixel Classification – um descritor de cor) and DEOBSM (bag of words usando amostragem densa, descritor ORB, *Soft assignment* e *Maximum pooling*) são as melhores opções considerando o triplo *trade-off* sobre eficácia, eficiência, e compacidade de características visuais. Análises estatísticas mostram que BOBGrid e BOBSlic são melhores do que nosso *baseline* WSA no conjunto de dados WANG. BOBGrid e BOBSlic também mostraram precisão maior em comparação ao BOSSANova no conjunto de dados WANG.

Palavras-chave: Recuperação de Imagens em Dispositivos Móveis, Descritores Globais, Estratégias de Amostragem, Descritores Binários, *Bag of Visuais Words*, *Bag of Visual Words* usando Informação Espacial, Compressão de Características Visuais.

Abstract

Mobile Visual Search (MVS) is a new research area in Content-Based Image retrieval (CBIR) which provides the services of search and retrieval of visual information specifically for mobile devices. The main challenges on mobile visual search include variations in image capturing conditions like different illumination, changes of scale and view angle, limitations of battery and high network cost incurred by data transmission.

The main purpose of this work is the comparison of efficient and effective techniques for feature extraction on mobile devices in order to retrieve images especially on smartphones. We achieve our goal by comparing and proposing techniques to feature vector compression and mid-level representation (bag of words). Some approaches reduce energy consumption in mobile devices because they send more compact feature vector to be processed on the server side. A series of experiments were also conducted to evaluate aspects of effectiveness, efficiency and compactness of extracted features of images in order to perform content-based image retrieval on mobile devices. In this case, the user decides the best triple trade-off configuration regarding effectiveness, efficiency, and compactness of visual features.

Therefore, we addressed two research issues in order to investigate and to propose effective solutions for image retrieval on mobile devices: 1) low-cost representation for mobile image search and 2) spatial visual feature extraction.

First, we analyze the use of binary descriptors using mid-level representation and global descriptors (color, texture, and shape) in image retrieval context on mobile devices, as well as, image features compression techniques. We have tested twenty mid-level representations of binary descriptors ("five binary descriptors" $\times$ "four bag of words strategies": BinBoost, BRIEF, BRISK, FREAK, ORB descriptors with bag of words using hard assignment with average pooling or bag of words using hard assignment with maximum pooling or bag of words using soft assignment with average pooling or bag of words using soft assignment with maximum pooling), ten color descriptors, five texture descriptors and two shape descriptors. We also analyze the impact of dense sampling and sparse sampling to compute descriptors using bags of words strategies

(dense sampling is the best option).

The second research issue refers to the problem of extracting spatial information on images to improve the quality of image representation on mobile devices, which could be crucial to distinguish types of objects and scenes. The traditional pooling methods usually discard the spatial configuration for visual words in the image. We propose two approaches of spatial bags of visual words called BOBGrid (spatial Bag Of BIC Grid) and BOBSlic (spatial Bag Of Slic) and compare them with our baseline called WSA (visual Word Spatial Arrangement) and with an improvement of the traditional bag of visual words called BOSSANova (Bag Of Statistical Sampling Analysis).

The experiments indicate that the descriptors BIC (Border/Interior Pixel Classification – a color descriptor) and DEOBSM (bag of words using DEnse sampling, ORB descriptor, Soft assignment and Maximum pooling) are the best options considering the trade-off configuration regarding effectiveness, efficiency, and compactness of visual features. In statistical analyzes, BOBGrid and BOBSlic are better than our baseline WSA in the WANG dataset. BOBGrid and BOBSlic also show higher precision compared to the BOSSANova in the WANG dataset.

Palavras-chave: Mobile Image Search, Global Descriptors, Sampling Strategies, Binary Descriptors, Bag of Visual Words, Spatial Bag of Visual Words, Visual Features Compression.

List of Figures

List of Tables

List of Acronyms

ACC *Auto-Correlogram Color*

ANOVA *Analysis of Variance*

Assignment *Step of associating the feature vector of a point detected in the image with the visual words in the dictionary; Also known as coding.*

BB *BinBoost descriptor*

BF *BRIEF descriptor*

BIC *Border/Interior Pixel Classification*

BinBoost *Boosting Binary Keypoint Descriptors*

BK *BRISK descriptor*

BN *BossaNova*

BoF *Bag-of-Features*

BOSSA *Bag Of Statistical Sampling Analysis*

BoVW *Bag of Visual Words*

BoF *Bag of Features*

BoW *Bag of Words*

BRIEF *Binary Robust Independent Elementary Features*

BRISK *Binary Robust Invariant Scalable Keypoints*

colorbitmap *Color Bitmap = Color features and image bitmap*

CBIR Content-Based Image Retrieval

CGCH *Cumulative Global Color Histogram*

CHOG *Compressed Histograms of Gradient*

CI *Confidence Interval*

CSD *Color Structure Descriptor*

CWHSV *Color Wavelet HSV*

CWLUV *Color Wavelet LUV*

DE *Dense sampling scheme where regions in an image are obtained by using a dense grid*

EOAC *Edge Orientation Auto-Correlogram*

FAST $^{(1)}$ *Fast Accelerated Segment Test*

FAST $^{(2)}$ *Fast Keypoint Recognition using Random Ferns*

FK *FREAK descriptor*

FREAK *Fast Retina Keypoint*

FT *FAST Keypoint Detector*

GCH *Global Color Histogram*

GT *GFTT (GoodFeaturesToTrackDetector) Keypoint Detector*

HA *Hard assignment using Average pooling*

Hard *Assignment scheme where a feature vector is assigned to only one visual work in the dictionary*

HM *Hard assignment using Maximum pooling*

HOG *Histograms of Oriented Gradients*

HS *Harris Keypoint Detector*

JAC *Joint Auto-Correlogram*

LBP *Local Binary Pattern*

k-NN *k Nearest Neighbors*

LAS	*Local Activity Spectrum*
LDA	*Linear Least Squares*
LBP	*Local Binary Pattern*
LCH	*Local Color Histogram*
MAP	*Mean Average Precision*
MIR	*Mobile Information Retrieval*
MR	*MSER Keypoint Detector*
MSER	*Maximally stable extremal regions*
MVS	*Mobile Visual Search*
OB	*ORB descriptor or ORB keypoint Detector*
ORB	*Oriented Fast and Rotated BRIEF*
P@N	*Precision measure for the top N retrieved images*
PCA	*Principal Component Analysis*
PLS	*Partial Least Squares*
Pooling	*Strategy for summarizing/selecting the assignment values from the coding/assignment step, generating the image feature vector*
QCCH	*Quantized Compound Change Histogram*
SA	*Soft assignment using Average pooling*
SASI	*Statistical Analysis of Structural Information*
SF	*SURF Keypoint Detector*
SIFT	*Scale Invariant Feature Transform*
SLIC	*Simple Linear Iterative Clustering*
SM	*Soft assignment using Maximum pooling*
Soft	*Assignment scheme where a feature vector can be assigned to more than one visual work in the dictionary*

Sparse *Sparse sampling scheme where regions in the image are obtained by using interest-point detector*

SPM *Spatial Pyramid Matching*

SPYTEC *Spherical Pyramid-Technique*

SURF *Speeded Up Robust Feature*

UNSER *Michael UNSER (paper author name) – Paper: Sum and difference histograms for texture classification*

VLAD *Vector of Locally Aggregated Descriptors*

VOC *Visual Object Classes*

WSA *visual Word Spatial Arrangement*

Contents

Chapter 1

Introduction

In 2014, the number of smartphone users worldwide reached around 1.75 billion [Cisco, 2015]. Recent forecasts indicate that the growth of smart mobile devices usage may increase even more the next years [Cisco, 2015]. Some important characteristics of recently developed devices are the high processing power and functions to capture images with high resolution. In addition to the new available wireless technologies, the current scenario has created a large demand for multimedia content processing applications. Many challenges and opportunities have emerged concerning image/video processing tasks, such as annotation, categorization, detection and retrieval.

Besides the traditional challenges, such as, translation, rotation and changes in scale and illumination, image processing in mobile devices is limited by many other constraints. For instance, the memory and computing resources may be very limited [Girod et al., 2011]. Significant system latency occurs on mobile based visual search systems, which is negligible in the case of computer-based visual search systems. The latency can be caused by processing delay on the mobile device, transmission delay on the network, and processing delay on the server. Due to the limited battery life of mobile devices, energy usage is also a critical issue [Girod et al., 2011]. Regarding feature extraction from images, those constraints configure a *trade-off* among effectiveness, efficiency and compactness. Therefore, it is important to perform a thorough evaluation regarding such aspects, as has been done in the literature in other domains (web image retrieval [Penatti and da Silva Torres, 2008], remote sensing image classification [dos Santos et al., 2010], mobile visual recognition and machine learning [Chatzilari et al., 2013], mobile augmented reality [Chen et al., 2013]).

1.1 Motivation

Due to the exponential growing of internet accessibility in the last decades, large multimedia collections have been created and managed in several applications. Social networks, digital libraries, and biodiversity information systems are just some examples. Given the large size of these collections, it is essential to provide efficient and effective mechanisms to retrieve images, which is the objective of the content-based image retrieval (CBIR) systems. In these systems, the searching process consists of finding the most similar image/videos stored in databases. The process relies on the use of descriptors, which can be characterized by two functions: feature extraction and similarity computation [Dos Santos et al., 2008]. The feature vectors encode image properties, such as color, texture and shape. The similarity between two images or videos is computed as a function of the distance between their feature vectors.

A novel class of applications that use images or video records to retrieve related information has been implemented on mobile devices. It is known as Mobile Visual Search (MVS) [Girod et al., 2011]. MVS technologies overcome the inherent limitations of text-based information retrieval systems, such as semantic fuzziness and abstract expression of language. With the upcoming Big Data era [Girod et al., 2011], visual based information extraction, analysis and retrieval will have more advantages over other information processing methods. In practice, many interesting applications have been developed, including Google Goggles[1] and Vuforia[2]. Mobile Visual Search (MVS) has become an active research field in the past few years [Girod et al., 2011]. It involves methods from several research areas, including image processing, computer vision, information theory and machine learning.

1.2 Related Work

In Kumar and Lu [2010] is possible to find an analysis that suggests which cloud computing can potentially save energy for mobile users. They showed that not all applications are energy-efficient when migrated to the cloud. Mobile cloud computing services would be significantly different from cloud services for desktops because they must offer energy savings.

In Tsai et al. [2010], a mobile product recognition system is presented. By snapping a picture of a product with a camera-phone, the user can retrieve online informa-

[1] Acessed on November 2015: Google Goggles - `https://play.google.com/store/apps/details?id=com.google.android.apps.unveil&hl=pt_BR`

[2] Acessed on November 2015: Vuforia - `https://developer.vuforia.com/`

tion of the product. They extract low bit-rate local descriptors (CHoG) from the query image [Chandrasekhar et al., 2009], compress their locations using location histogram coding [Tsai et al., 2009] and send the significantly smaller compressed features instead of the query image, reducing the time spent in communication. The server processes the query data and recognizes the item among a large database of products. Tsai et al. [2010] also compress the inverted index in RAM to reduce the memory size and prevent memory swapping operations [Chen et al., 2010] and use fast geometric re-ranking to reduce the time for geometric verification [Chen et al., 2010].

Girod et al. [2011] present the Stanford Product Search system, a low latency interactive visual search system. The Stanford Product Search system can perform feature extraction and compression on the client, to reduce system latency. The authors choose the Hessian-blob detector speed up with integral images [Bay et al., 2008] which provides a good trade-off of repeatability and complexity. After interest point detection, they compute a "visual word" descriptor on the normalized patch. In their previous work they designed the Compressed Histogram of Gradients (CHoG) descriptor [Chandrasekhar et al., 2009, 2010b] witch is also use in this work. CHoG builds upon the principles of HoG descriptors with the goal of being highly discriminative at low bitrates Girod et al. [2011]. They use a data structure that can quickly return a shortlist of the database candidates most likely to match the query image and use location information of query and database features to confirm that the feature matches are consistent with a change in viewpoint between the two images. The geometric transformation between query and database image is estimated using robust regression techniques like RANSAC [Fischler and Bolles, 1981].

In Valenzuela et al. [2013], the authors propose to apply linear dimensionality-reduction kernels to reduce the dimensions of SIFT [Lowe, 2004] and SURF [Bay et al., 2008] feature vectors and employ bag-of-features (BoF) [Csurka et al., 2004; Lazebnik et al., 2006] to create global features to maintain or even enhance the accuracy of the retrieval, while spending less storage and computational time requirements. In the training stage the Corel Dataset[3] were selected. Then, a dimensionality reduction technique (such as PCA, PLS or LDA) is applied to the 250 images that are used to learn the projection kernel. Once the projection kernel is computed, the feature vectors corresponding to the remaining 500 images are projected onto the kernel to obtain the reduced feature vectors. The experiments compare the results achieved by the reduced feature vectors to the results obtained by the original features, demonstrating to gain in accuracy while reducing computational time and storage.

[3]Acessed on November 2015: Corel Dataset (2013) - `http://people.csail.mit.edu/torralba/research/bias/`.

Ascenso and Pereira [2013] present the lossless compression of binary image features which is proposed to further lower the energy and bandwidth requirements. The coding solution exploits the redundancy between descriptors of an image by sorting the descriptors and applying Differential Pulse Coded Modulation (DPCM) and arithmetic coding. The proposed coding method targets the ORB [Rublee et al., 2011], BRISK [Leutenegger et al., 2011] and FREAK [Ortiz, 2012] binary descriptors. Experiments have suggested that the coding efficiency of intra coding schemes is low mainly because the intensity tests were designed to have a rather unique descriptor. However, coding schemes that prioritize the importance of each intensity tests and consider rate constraints could lead to compression improvements [Redondi et al., 2013]. The spatial correlation between neighboring descriptors or the correlation between descriptors that describe a repetitive pattern of the image, i.e. descriptors of patches with similar appearance, can also be exploited with an efficient inter coding scheme.

In Zhuang et al. [2014], the authors propose a novel scheme to quantify the spatial context information and convert it into binary strings. They address the problem of constructing effective and efficient Content-Based Image Retrieval system by using binary features. To improve the discriminative power and robustness of binary descriptors, largely enhance the retrieval accuracy of the image search system where they propose a novel binary descriptor extraction algorithm. They adopt the ORB, BRIEF, and FREAK methods to extract the binary features, whose codes are published by the authors in OpenCV[4] [Bradski and Kaehler, 2013]. The experimental results have been performed on common public datasets (UK-Benchmark [Nister and Stewenius, 2006], Oxford5K [Philbin et al., 2008], Paris [Philbin et al., 2007]) and in a dataset used in Mikolajczyk and Schmid [2005]). The results show that their approach outperforms the original binary features with the similar memory usage and computational time, even the original binary feature schemes also use the spatial information.

Monteiro and Ascenso [2014] proposed a decision algorithm for coding visual features extracted from images. The authors use an algorithm that selects between intra-and-inter-prediction modes, exploiting the correlation between descriptor elements and descriptors. They use the state-of-the-art binary descriptor extraction algorithms (BRIEF, BRISK or FREAK) and a large set of binary descriptors are extracted from a dataset of images. Initial prediction order is done, by computing the residual error. Then, the intensity test order previously obtained is refined. The evaluation criterion is not based on the residual error but rather on the entropy (an estimate of the rate) of all binary descriptors, thus leading to a set of correlated descriptor elements

[4]Acessed on November 2015: OpenCV - `http://opencv.org/`

(DEs) that can be coded more efficiency. Regarding the Inter mode, each descriptor is differentially encoded (XOR based) online using other previously coded descriptors as reference since the descriptors can be reordered in any arbitrary way. To perform image retrieval, pair-wise matching between the decoded descriptors from the query images and the database descriptors is first performed. Then, wrong matches between the query and the database descriptors are removed using RANdom SAmple Consensus (RANSAC). This procedure returns a ranked list of database images given a certain query image.

1.3 Research Challenges

Due to the complexity of visual information compared with text or voice signals, the information description methods and retrieval algorithms in MVS are different from those of the traditional information search engines. In addition, because of large-scale databases used and the different features generated from each image, mobile visual search is required to deal with vast amount of data flow under real-time constraints.

There are two important challenges in CBIR: 1) finding a relevant visual content representation associated with a similarity measure that allows effective comparison with a query and 2) making retrieval scalable to large image datasets by building an appropriate index structure able to better exploit the content representation and similarity measure.

Mobile image retrieval applications pose a set of challenges. For instance, what part of the processing should be performed on mobile client, and what part is better carried out at the server [Girod et al., 2011]? When the database is small, it can be stored in the phone and image retrieval algorithms can be run locally [Takacs et al., 2008; Hong et al., 2009]. When the database is large, it has to be placed on a remote server and the retrieval algorithms are run remotely [Girod et al., 2011; Chen et al., 2013].

Mobile Visual Search (MVS) systems seek similar visual information as query, starting from a mobile terminal. Although much attention has been focused on mobile visual search recent years, a set of challenges still remain to be addressed. These challenges include:

1. variations in image capturing conditions such as different illumination (CBIR challenges);

2. changes in scale, rotation, translation and view angle (CBIR challenges);

3. limitations of battery and memory usage (MVS challenges);

4. high network cost incurred by data transmission (MVS challenges).

To overcome the aforementioned challenges, a MVS system with efficient and robust performance requiring less resource is necessary. 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 fruits and flowers.

Sometimes, color histograms, texture descriptors and simple shape descriptors are not so robust to retrieve image. Recently, more elaborated image representations, known as mid-level representations, have been proposed to deal with the complexity of the task, by aggregating several low-level local descriptions into a single feature vector.

Several works on the literature commonly use state-of-art image descriptors such as SIFT [Lowe, 2004] and SURF [Bay et al., 2008] together with mid-level representations. However, these descriptors require a high computational processing time, generating high vector dimensionality composed of real values. Thus, as an alternative, low complexity, binary descriptors can be applied to generate similar results when compared to non-binary descriptors state-of-the-art. Moreover, global low-level feature representation can be an alternative since they are less expensive.

1.4 Objectives and Contributions

The objective of this work is to make a study of low-cost representations for image feature extraction and to propose effective solutions to improve visual spatial representation for image retrieval on mobile devices.

In this thesis, we investigate feature representation strategies for allowing real-time content-based image retrieval using mobile devices as query interfaces. In this model, the feature extraction step is performed on the device and the search step is processed on the server, as illustrated in Figure 1.1. So, only the query feature vectors are required to be transferred. In image retrieval on mobile devices, we need compact and fast, but also accurate image representations. We achieve our objective by evaluating and developing the following methods: (1) robust and efficient feature extraction algorithms that fit mobile device constraints and (2) fast and effective feature vector compression.

The main contributions of this thesis are the following.

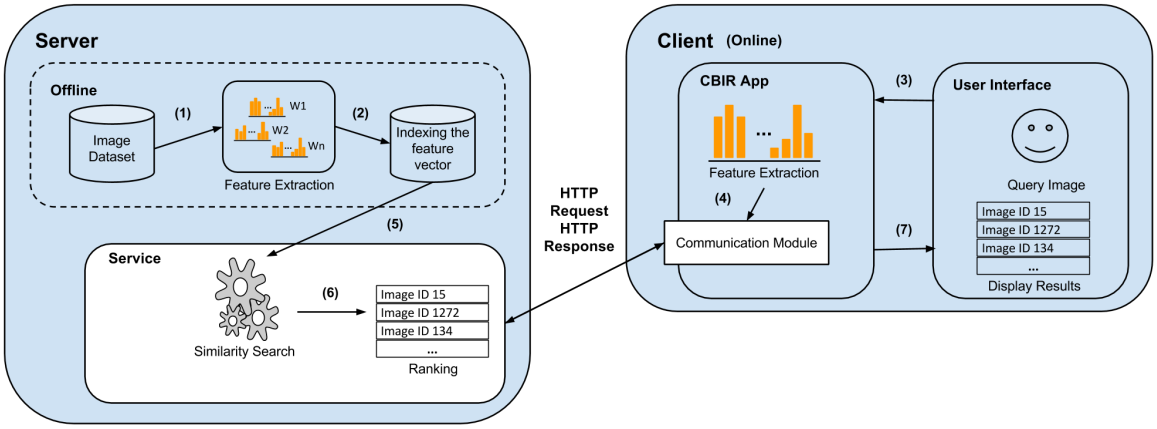


Figure 1.1: Mobile visual search architecture used in this work. Low-cost image feature extraction is performed on mobile devices and the search step is processed on the server the server side. Only the query feature vectors are required to be transferred. In image retrieval on mobile devices, we need compact and fast, but also accurate image representations.

1. A comparative study of binary descriptors using mid-level representation and global descriptors (color, texture, and shape) in an approach of image retrieval on mobile devices, as well as, image features compression techniques. We also analyze the impact of dense sampling and sparse sampling to compute descriptors using bags of words strategies;

2. We propose two new bag of visual word representations that include spatial information to improve the quality of image representation on mobile devices, which could be crucial to distinguish types of objects and scenes. We compare them, using statistical analyzes, with recent spatial bag of visual words;

Some results obtained in this work were published in XX Iberoamerican Congress on Pattern Recognition (CIARP 2015) and in XXVIII Conference on Graphics, Patterns and Images (SIBGRAPI Work in Progress 2015). In [Pessoa et al., 2015a], we performed an extensive study on low-cost representations for image feature extraction on mobile devices and in [Pessoa et al., 2015b], we performed an experimental comparison of feature extraction and distance metrics for image retrieval.

1.5 Organization of the Text

We organized the remainder of this work in five chapters.

Chapter 2: Background We present the background concepts necessary for the understanding of this work: (i) feature extraction (local and global features), (ii) mid-level representation, (iii) image feature compression, (iv) image segmentation and (vi) evaluation metrics.

Chapter 3: Benchmark Datasets We introduce the variety of benchmark image datasets used in this thesis. We detail each dataset and discuss how they differ from one another.

Chapter 4: Low-Cost Representation for Mobile Image Search We show the details of the proposed approaches of image retrieval on mobile devices using binary descriptors with bag-of-visual words representation and also using global descriptors (color, texture and shape). We present several analysis and point out the best configuration of low-cost image representation in terms of effectiveness, efficiency and compactness. We analyzed the impact of use dense sampling or sparse sampling for select points for descriptors used on mid-level representation (in particular, bag of words representations).

Chapter 5: Spatial Feature Representation for Mobile Image Search We used and analyzed four bags of visual words strategies: WSA (Visual Word Spatial Arrangement), BOSSANova (Bag Of Statistical Sampling Analysis) and two proposed approaches called BOBGrid (Spatial Bag of BIC Grid) and BOBSlic (Spatial Bag of Slic BIC).

Chapter 6: Conclusion and Future Work We summary the research activities and provide the concluding remarks and future directions of research in mobile image retrieval.

Chapter 2

Background

In this chapter, we present the background concepts necessary to understand the approaches we have analyzed and proposed in this thesis. Section 2.1 presents local and global descriptors used along this thesis. Section 2.2 presents mid-level representation and sampling strategies. Section 2.3 describes some compression techniques. Section 2.4 introduces image segmentation and presents the SLIC superpixel algorithm. Finally, section 2.5 presents the metrics used for evaluation of content-based image retrieval systems in terms of effectiveness, compacteness, and efficiency.

2.1 Feature Extraction

The majority of computer vision tasks employs low-level image features, including local and global features, to extract the information from images directly and represent the image content in a normalized form. In CBIR, local features are tipically employed with mid-level representation in order to encode the global visual content of the images [Penatti et al., 2011a]. Global features can represent an image with only one vector leading to reduced computation cost. Mid-level features are also good alternatives since they provide suitable representation for the amount of local features extraction.

2.1.1 Local Binary Features

One of the most famous non-binary descriptors used in the literature is known as Scale Invariant Feature Transform (SIFT) [Lowe, 2004]. This descriptor performs a scale-space analysis leading to a great performance in relation to the scale invariance. Other non-binary descriptors often used is the Speeded Up Robust Feature (SURF) [Bay et al., 2008]. SURF performs faster than SIFT by using a integral image and also

use different filter sizes to simulate the change of view scale to achieve invariance to rotation. Haar wavelets are employed to determine the main orientation of the gradient around each keypoint.

As an alternative, low-complexity binary descriptors have recently emerged. There are three main advantages of this kind of descriptor: (1) the time required for extracting, (2) the small size of extracted feature vector and (3) low-cost matching. Therefore, in this work we use binary descriptors instead of non-binary descriptors. In this subsection, we briefly describe some binary descriptors and their properties.

Binary Robust Independent Elementary Features [Calonder et al., 2010] The Binary Robust Independent Elementary Features (BRIEF) descriptor describes features using simple binary tests among pixels from a smoothed image. By itself, it is neither scale nor rotation invariant. Nevertheless, its performance is similar to a more complex local descriptor, the SURF, when compared to its robustness to illumination, blur, and perspective distortion. The BRIEF descriptor is represented by a binary string in which each bit represents a simple comparison between two elements inside a patch. The keypoint is the center of this patch. According Calonder et al. [2010], the most common strategy for choosing these points is based on a randomly way according to a Gaussian distribution with respect to the keypoint of the patch. Due to their simplicity, BRIEF descriptor are very fast to compute.

Oriented FAST and Rotated BRIEF [Rublee et al., 2011] The ORB descriptor (Oriented FAST and Rotated BRIEF) can be considered as an alternative for SIFT and SURF being two times faster than SIFT and one time faster than SURF. The ORB descriptor is partially invariant to scale and is robust to noise and invariant to rotation, solving the invariance problem of BRIEF. The sampling pattern is steered estimating the orientation, and usual binary tests are used for computing the descriptor. For selecting a couple of points, a k-nearest neighborhood strategy based on error-prone is done. ORB calculates the orientation as the angle of the vector from the keypoint center to the intensity centroid. ORB also assumes that intensity test must be uncorrelated to obtain good matching performance. ORB's descriptor size is fixed to a 256 bitstring.

Binary Robust Invariant Scalable Keypoints [Leutenegger et al., 2011] The Binary Robust Invariant Scalable Keypoints (BRISK) descriptor provides scale and rotation invariance, however it is very sensitive to light intensity variations. The BRISK descriptor computes a weighted Gaussian average over a selected pattern of points that are close to the keypoint. To achieve rotation invariance, the direction of each

keypoint is estimated by averaging the local gradients obtained with several long-distance sampling points. For comparing the points, Gaussian windows are used to set the bit to '1' or '0'. Due to its size, the BRISK descriptor is represented by a 512 bits, which is more greater that BRIEF and ORB, and consequently, more computation and storage are required.

Fast Retina Keypoint [Ortiz, 2012] The Fast Retina Keypoint (FREAK) descriptor also provides scale and rotation invariance, however its pattern is based on Gaussians and it is biologically-inspired on the retinal pattern of the human eye. In the creation of the FREAK descriptor, the same approach as for ORB was employed, notably using a learning algorithm to obtain the best intensity tests. In summary, FREAK improves upon the sampling pattern and method of pair selection that BRISK uses. Moreover, the patterns are much more concentrated near the keypoint that leads to a more accurate description of the keypoint. To speed up the matching process, the actual FREAK algorithm also uses a cascade for comparing these pairs, and puts the 64 most important bits in the beginning of the descriptor.

Boosting binary keypoint descriptors (BinBoost) [Trzcinski et al., 2013] BinBoost is a robust descriptor to lighting and scale changes. Different from the aforementioned binary descriptors which calculate the final descriptor based on simple binary tests comparing the intensity of pixels, each bit generated by BinBoost is calculated using a binary hash function the same as the classifier AdaBoost [Freund and Schapire, 1997]. This function is based on weak learners that take into account intensity gradients of guidance on the patch to be described. The hash function is optimized iteratively, i.e., at each iteration, incorrect samples will be assigned to a greater weight, while the weight of correct samples will be diminished. In this way, the next bit to be calculated will tend to correct the error of his predecessors different descriptors.

2.1.2 Global Feature Descriptors

Global features, which describe an image as a whole, can represent image content efficiently. They provide an overall spatial organization of scale and orientation information of the image. If by one side the literature indicates that, in general, global features are less effective than local ones, by the other side, they are more efficient since they are not dependent on mid-level representation. In this work, we use seventeen global descriptors (ten color descriptors, five texture descriptors and two shape descriptors), as shown in Table 2.1.

Color	Vector Size	Texture	Vector Size	Shape	Vector Size
ACC	256	**LAS**	256	**EOAC**	288
BIC	128	**LBP**	10	**SPYTEC**	16
CGCH	64	**QCCH**	40		
CBIT	306	**SASI**	64		
CSD	64	**UNSER**	32		
CWHSV	64				
CWLUV	127				
GCH	64				
JAC	256				
LCH	64				

Table 2.1: Global descriptors and their vector size. CBIT = Color Bitmap, JAC = Joint Auto-Correlogram.

Auto-Correlogram Color (ACC) [Huang et al., 1997] The role of this descriptor is to map the spatial information of colors by pixel correlations at different distances. It computes the probability of finding in the image two pixels with color C at distance d from each other. For each distance d, m probabilities are computed, where m represents the number of colors in the quantized space. The implemented version quantized the color space into 64 bins and considered 4 distance values (1, 3, 5, and 7).

Border/Interior Pixel Classification (BIC) [Stehling et al., 2002] In this descriptor, the first step of the feature vector extraction process relies on the classification of image pixels into border or interior ones. When a pixel has the same spectral value in the quantized space as its four neighbors (the ones which are above, below, on the right, and on the left), it is classified as interior. Otherwise, the pixel is classified as border. Two histograms are computed after the classification: one for the interior pixels and another for the border ones. Both histograms are merged to compose the feature vector. The implemented version quantized the color space into 64 bins for border and 64 bins for interior. The final vector lenght has 128 bins. We used the dlog function distance in our experiments, as well as the $L1$ (Manhattan) distance.

Cumulative Global Color Histogram (CGCH) [Stricker and Orengo, 1995] In the extraction algorithm of this descriptor, the value of each bin is cumulated in the next bin. This makes the last bin have the sum of all the previous bins plus the actual bin. In our experiments, the color space was quantized into 64 bins and the $L1$ (Manhattan) distance function was used.

Color Bitmap [Lu and Chang, 2007] This descriptor analyzes image color properties globally and locally. Its extraction algorithm computes the mean and the standard deviation of the R, G, and B channels independently. After that, the image is splitted into m blocks and the mean of each block is computed for each channel. If the block mean is greater than the image mean, the correspondent feature vector position receives 1; otherwise, it receives 0. The implemented version used 100 blocks.

Color Structure (CSD) [Manjunath et al., 2001] The CSD extraction algorithm uses the HMMD (hue, max, min, diff) color space and scans the image with a 8×8 pixels structuring element. A histogram $h(m)$ is incremented if the color m is inside the structuring element, where m varies from 0 to $M - 1$ and M is the color space quantization.

Color Wavelet HSV (CWHSV) [Utenpattanant et al., 2006] This descriptor considers image color properties in the wavelet domain. The extraction algorithm uses the HSV color space quantized into 64 bins and computes a global color histogram for the image. After that, the Haar wavelet coefficients are hierarchically computed. This is done recursively by dividing the histogram in the middle: if the sum of the values from the first half are greater than the sum of the values from the second half, the correspondent feature vector position receives 1; otherwise, 0. The process is repeated until the last possible level of division, what leads to 64 bits in the feature vector.

Color Wavelet LUV (CWLUV) [Nallaperumal et al., 2007] Similar to Color Wavelet HSV, but use the LUV color space quantized into 127-bit binary Haar color histogram, which is used as an index of the images.

Global Color Histogram (GCH) [Swain and Ballard, 1991] This descriptor uses an extraction algorithm which quantizes the color space in a uniform way and it scans the image computing the number of pixels belonging to each color (bin). The size of the feature vector depends on the quantization used. In this work, the color space was split into 64 bins, thus, the feature vector has 64 values.

Joint Auto-Correlogram (AUTOCORRJOIN/JAC) [Williams and Yoon, 2007] This descriptor follows the same principle used by Color Auto-Correlogram Color (ACC). However, its extraction algorithm computes the autocorrelogram for more than one image property. The properties considered are: color, gradient magnitude, *rank*, and *texturedness*. Color is extracted in RGB color space and the other properties

are extracted from the gray level image. The joint autocorrelogram indicates, for each distance considered, the probability of simultaneously occurring the four properties considered. The implemented version used the HSV color space quantized into 64 bins, 5 bins for the other three properties, a 5×5 pixel neighborhood and 4 distance values (1, 3, 5, and 7).

Local Color Histogram (LCH) [Swain and Ballard, 1991] LCH is one of the most popular descriptors that is based on fixed-size regions to describe image properties. Its extraction algorithm splits the image into fixed-size regions and computes a color histogram for each region. After that, the histograms of each region are concatenated to compose one single histogram. The implemented version splitted the image into 16 regions (4×4 grid) and quantized the RGB color space into 64 bins. The L1 distance function was used.

Local Activity Spectrum (LAS) [Tao and Dickinson, 2000] This descriptor captures texture spatial activity in four different directions separately: horizontal, vertical, diagonal, and anti-diagonal. The four activity measures are computed for a pixel $(i; j)$ by considering the values of neighboring in the four directions. The values obtained are used to compute a histogram that is called local activity spectrum. Each component g_i is quantized independently. In our experiments, each component was non-uniformly quantized into 4 bins, leading to a histogram with 256 bins. Distance is computed by $L1$ function.

Local Binary Pattern (LBP) [Wang and He, 1990; Ojala et al., 1996, 2002] LBP is a simple texture descriptor that is invariant to rotation and variations in the gray scale values. Its extraction algorithm defines a window with radio R and a quantity of neighbors P and scans the image counting the quantity of positive and negative variations between the gray values of the neighbor pixels and the central pixel of the window. For gray scale invariance, only the signal of the variation is considered, being 1 for positive and 0 for negative variation. After that, the number of 0/1 and 1/0 transitions are computed, what guarantees the rotation invariance. If the number of transitions is less than 2, the LBP value for that window position is equal to the quantity of 1 signals in the neighborhood. Otherwise, the LBP value is $P + 1$. After all the image is scanned, a histogram of LBP values is computed. In our experiments, $R = 1$ and $P = 8$ values. The distance function used was the L1 distance.

Quantized Compound Change Histogram (QCCH) [Huang and Liu, 2007] This descriptor uses the relation between pixels and their neighbors to encode texture information. This descriptor generates a representation invariant to rotation and translation. Its extraction algorithm scans the image with a square window. For each position in the image, the average gray value of the window is computed. Four variation rates are then computed by taking into consideration the average gray values in four directions: horizontal, vertical, diagonal, and anti-diagonal directions. The average of these four variations is calculated for each window position. They are then grouped into 40 bins and a histogram of these values is computed.

Statistical Analysis of Structural Information (SASI) [Carkacioglu, 2001; Çarkacıoglu and Yarman-Vural, 2003] SASI defines a set of windows to extract and measure various structural properties of texture by using a spatial multiresolution method. SASI measures the spectral information, while it works in the spatial domain. Like Gabor Filter [Jain and Farrokhnia, 1990] descriptor, SASI employs different orientation and size of the moving windows. However, implementation of SASI is more robust compared to Gabor Filters.

UNSER [Unser, 1986] This descriptor is based on co-occurrence matrices. The extraction algorithm computes a histogram of sums H_{sum} and a histogram of differences H_{dif} . The histogram of sums is incremented considering the sum, while the histogram of differences is incremented by taking into account the difference between the values of two neighbor pixels. As well as gray level co-occurrence matrices, measures such as energy, contrast, and entropy can be extracted from the histograms. In our experiments, eight different measures were extracted from histograms and four angles are used (0°, 45°, 90°, and 135°). The final feature vector is composed of 32 values.

Edge Orientation Autocorrelogram (EOAC) [Mahmoudi et al., 2003] This descriptor classifies image edges based on their orientation and correlation between neighboring edges. It includes information of continuous edges and lines of images and describes major shape properties of images. This scheme tolerates translation, scaling, color illumination, and viewing position variations.

Spherical Pyramid-Technique (SPYTEC) [Lee and Kim, 2001] This descriptor extracts the shape feature of images automatically using edge detection and wavelet transform. SPYTEC is based on a special space partitioning strategy which divides the d-dimensional data space first into $2D$ spherical pyramids, and then cuts the single

sperical pyramid into several spherical slices. This particion provides a transformation of d-dimensional space into 1-dimentional space. It uses a B^+-tree to manage the transformed 1-dimensional data.

2.2 Mid-level Representation

In general, hundreds of local features are extracted from each single image. Their robustness to the change of view point, illumination and scale makes them attractive for many computer vision applications. However, their high dimensionality inhibit their use in real-time computer vision tasks involving large scale data processing.

To take advantage of both the robustness of local features and the efficiency of representing each image with a single vector, different mid-level features [Boureau et al., 2010], sometimes referred to image feature aggregation models or mid-level representation, have been proposed. These mid-level image features aim generating a higher level image representation from a low-level feature set by employing different feature aggregation approaches.

Since the Bag of Visual Words (BoVW) model has borrowed from text retrieval area in the work of Sivic and Zisserman [2003], more and more similar mid-level features have been developed. Examples are Fisher Vector [Perronnin et al., 2010], Vector of Locally Aggregated Descriptor (VLAD) [Jégou et al., 2010b], BOSSANova (Bag Of Statistical Sampling Analysis) [Avila et al., 2013] and WSA (Visual Word Spatial Arrangement) [Penatti et al., 2014]. These methods achieve state-of-art performance in visual search area.

The Bag of Visual Words model converts the set of local descriptors into the final image representation vector by a succession of four steps (see Figure 2.8): 1) sampling strategy (selection of regions (patches) into the image), 2) local feature descriptor (non-binary or binary image representation for each patch in the image), 3) coding (assigning each local descriptor to visual words) and 4) pooling (summarizing the local descriptor projections using average or maximum operations, for example). In the following two subsection, we detail each of the four steps.

2.2.1 Sampling Strategies

According to Tuytelaars [2010], the patch selection can be based on two approaches: (i) using points of interest (sparse sampling): in this case an algorithm is applied to find such a region to be described; or (ii) dense sampling, where fixed-size regions are

allocated on a regular grid size. An example of dense sampling is show in Figure 2.1. Examples of sparse sampling are shown on Figures 2.2, 2.3, 2.4, 2.5, 2.6 and 2.7.

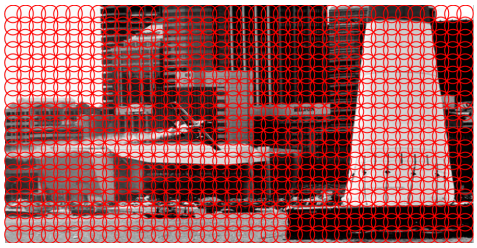
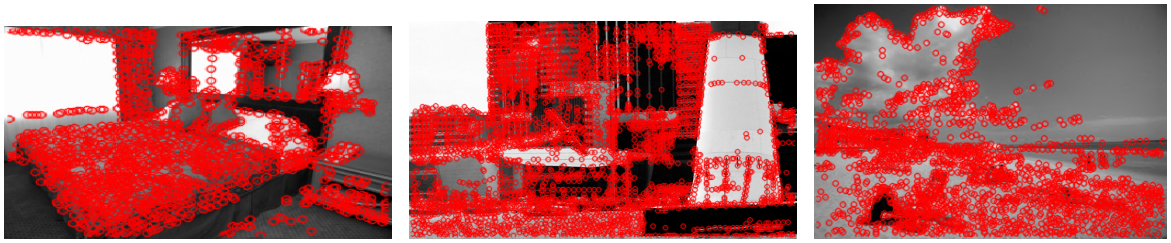
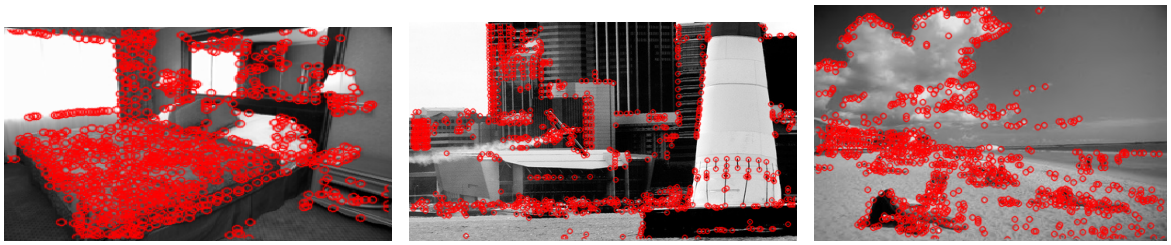


Figure 2.1: Dense Sampling 6×6 pixels.



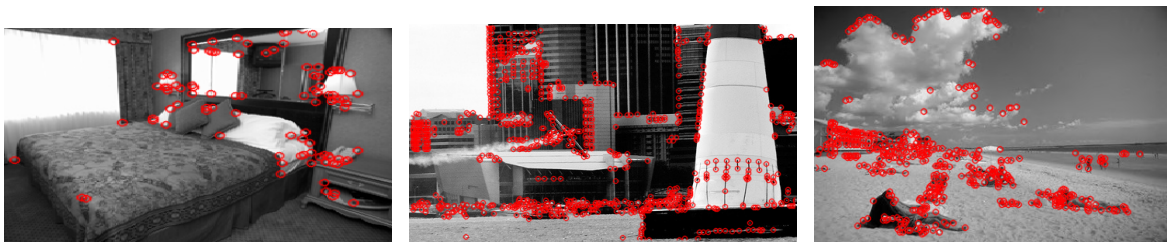
(a) Dataset: 15Scenes (b) Dataset: VOC2007 (c) Dataset: WANG

Figure 2.2: Keypoints using FAST sparse sampling.



(a) Dataset: 15Scenes (b) Dataset: VOC2007 (c) Dataset: WANG

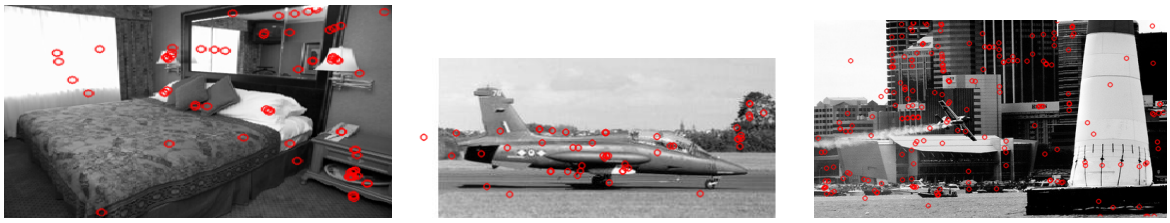
Figure 2.3: Keypoints using GFTT sparse sampling.



(a) Dataset: 15Scenes (b) Dataset: VOC2007 (c) Dataset: WANG

Figure 2.4: Keypoints using GFTTHarris sparse sampling.

In this thesis we use six sparse sampling: 1) FAST (Features from Accelerated Segment Test), 2) GFTT (Good Features to Track), 3) GFTTHarris (Good Features to Track using Harris), 4) Maximally Stable Extremal Regions (MSER), 5) Oriented FAST and Rotated BRIEF (ORB Detector) and 6) Speeded-Up Robust Features (SURF Detector).

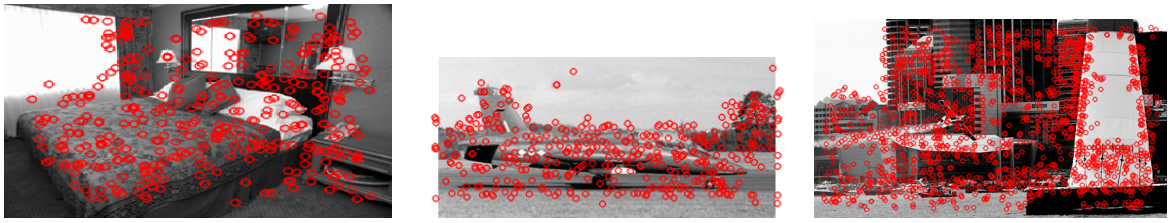


(a) Dataset: 15Scenes (b) Dataset: Caltech101 (c) Dataset: VOC2007

Figure 2.5: Keypoints using MSER sparse sampling.

(a) Dataset: 15Scenes (b) Dataset: Caltech101 (c) Dataset: VOC2007

Figure 2.6: Keypoints using ORBDetector sparse sampling.



(a) Dataset: 15Scenes (b) Dataset: Caltech101 (c) Dataset: VOC2007

Figure 2.7: Keypoints using SURFDetector sparse sampling.

FAST (Features from Accelerated Segment Test) [Rosten and Drummond, 2006] In Trajković and Hedley [1998], the authors developed a detector in which the value of the central pixel of a specific region is compared with the values of other pixels within this same region. Rosten et al. [Rosten and Drummond, 2006] improved this idea with a machine learning-based approach to create decision trees to enable the detector to classify a candidate point with less comparisons between pixels, giving rise to the FAST detector.

Harris Affine Detector [Mikolajczyk et al., 2005] The Harris affine detector can identify similar regions between images that are related through affine transformations and have different illuminations. The Harris affine detector relies on the combination of corner points detected through Harris corner detection, multi-scale analysis through Gaussian scale space and affine normalization using an iterative affine shape adaptation

algorithm.

Good Features to Track (GFTT) [Tomasi and Shi, 1994] Good Features to Track is a corner detector which shows better results compared to Harris Corner Detector [Tomasi and Shi, 1994]. The scoring function in Harris Corner Detector was given by: $R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$. Instead of this, Tomasi and Shi [1994] proposed: $R = min(\lambda_1, \lambda_2)$. If it is a greater than a threshold value, it is considered as a corner.

Maximally Stable Extremal Regions (MSER) [Matas et al., 2002] is an algorithm that extracts from an image I a number of co-variant regions, called MSERs. An MSER is a stable connected component of some level sets of the image I. Optionally, elliptical frames are attached to the MSERs by fitting ellipses to the regions.

Oriented FAST and Rotated BRIEF (ORB Detector) [Rublee et al., 2011] ORB is basically a fusion of FAST keypoint detector and BRIEF (Binary Robust Independent Elementary Features [Calonder et al., 2010]) descriptor with many modifications to enhance the performance. First it use FAST to find keypoints, then apply Harris corner measure to find top N points among them. It also use pyramid to produce multiscale-features. But one problem is that, FAST does not compute the orientation. To solve this problem, the authors came up with following modification. It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this corner point to centroid gives the orientation. To improve the rotation invariance, moments are computed with x and y which should be in a circular region of radius r, where r is the size of the patch.

Speeded-Up Robust Features (SURF Detector) [Bay et al., 2008] SURF rely on determinant of Hessian matrix [Beaudet, 1978] for both scale and location. For orientation assignment, SURF uses wavelet responses in horizontal and vertical direction for a neighbourhood of size 6. Adequate guassian weights are also applied to it. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window of angle 60 degrees. Interesting thing is that, wavelet response can be found out using integral images very easily at any scale.

2.2.2 Bag of Visual Words

The mid-level representation is useful to convert a set of local features into an unique global representation for each image, which is called Bag of Visual Words (BoVW).

This process is divided into offline and online steps, as shown in Figure 2.8. In the offline phase (steps 1, 2, 3, 4, and 5), after local feature descriptor are obtained on the image database, the features are clustered to create the vocabulary of visual words (also known as codebook or dictionary) [Csurka et al., 2004; van Gemert et al., 2010]. The codebook is usually built by partitioning the low-level feature space. It can be defined by the set of visual words, corresponding to the centroids of clusters. In the online phase, the same process (dense sampling, local feature descriptor, clustering features into visual words) is done using the codebook as a dictionary to do assignment and pooling steps. Assignment (or coding) is a step that associates the feature vector of a point detected in the image with the visual words in the dictionary. A pooling strategy is used for summarizing/selecting the assignment values from the coding/assignment step, generating the image feature vector [Penatti et al., 2014].

In this thesis, we have evaluated **two word assignment strategies** with different pooling approaches (average and maximum):

- **Hard assignment** [Sivic and Zisserman, 2003]: the local feature descriptors of the image are matched with visual words of the vocabulary (the nearest one). A histogram of the visual descriptors is populated by the corresponding bins.

- **Soft assignment** [van Gemert et al., 2010]: in this case, instead of assigning a descriptor to a single corresponding visual word, we assign it to k bins in a soft manner. More specifically, for every descriptor, we add a quantity q to the bins of the k top nearest visual words. This quantity q is the Gaussian kernel (Radial Basis Function) distance of the descriptor and the visual word.

The differences among each assignment/pooling strategy can be observed in the example presented in Figure 2.9. Basically, in hard-assignment, just one bin is activated per feature and in the soft-assignment, a group of nearest bins are activated in a soft manner. We used two pooling strategies (average or maximum) to summarize the assignment vectors. After the assignment and pooling steps, the Bag of Visual Words final vector representation is created and can be used in visual pattern recognition tasks, such as content-based image retrieval or categorization/classification of images.

Normalization In the bag of visual words using soft assigment with **maximum** pooling the distances are smoothed by a Gaussian, which gives less weights for farther re-

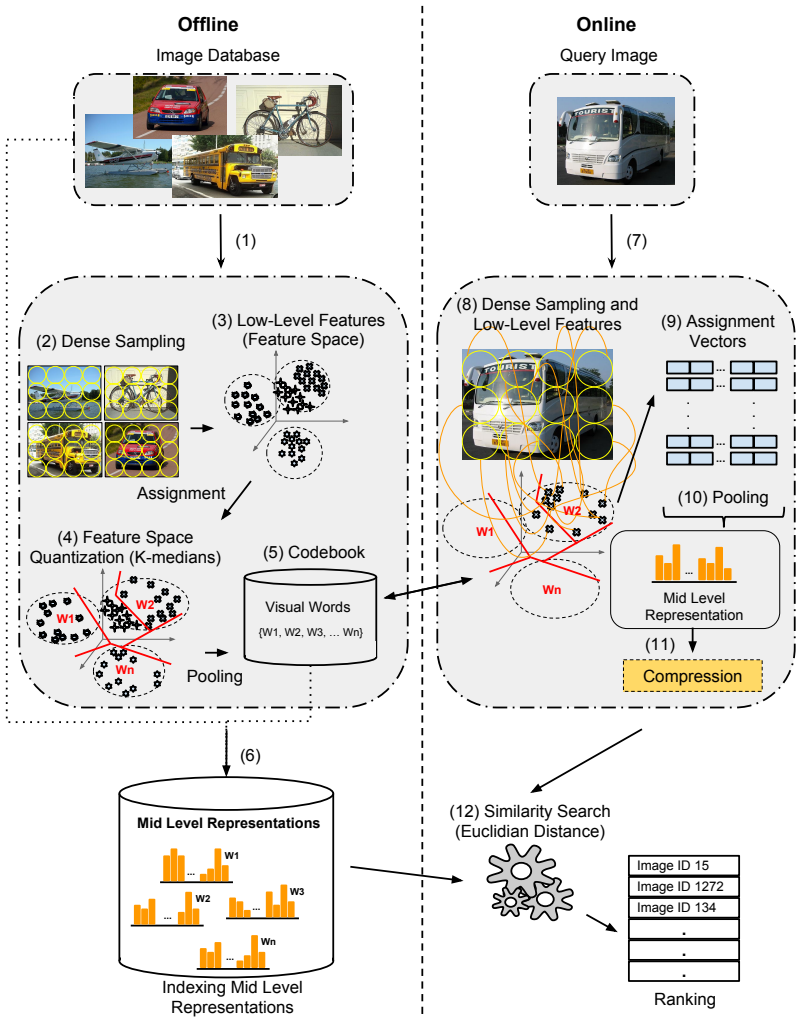


Figure 2.8: The content-based image retrieval process used to evaluate low-cost feature representation. **Offline phase**: after extracting local feature vectors from the image dataset (steps 1 to 3), the feature space is quantized (step 4) and each region corresponds to a visual word (step 5). We use the codebook to create bag-of-word representations for all images in the database (step 6). **Online phase:** given a query image (step 7), its local feature vectors are computed (step 8) and then assigned to the visual words in the dictionary (step 9). Finally, the local assignment vectors are summarized by a pooling strategy, which creates the bag-of-visual-words representation (step 10). A compression step may be processed to reduce the feature vector size (step 11). In the similarity search, a distance function (e.g., Euclidean or Manhattan) is used as similarity measure to rank the database images (step 12).

gions and higher importance for closer ones, as in [Penatti et al., 2011b]). The equation for the aforementioned soft assignment used in this thesis is the following:

$$\alpha_{i,j} = \frac{K_\sigma(D(v_i, w_j))}{\sum_{l=1}^{k} K_\sigma(D(v_i, w_l))} \tag{2.1}$$

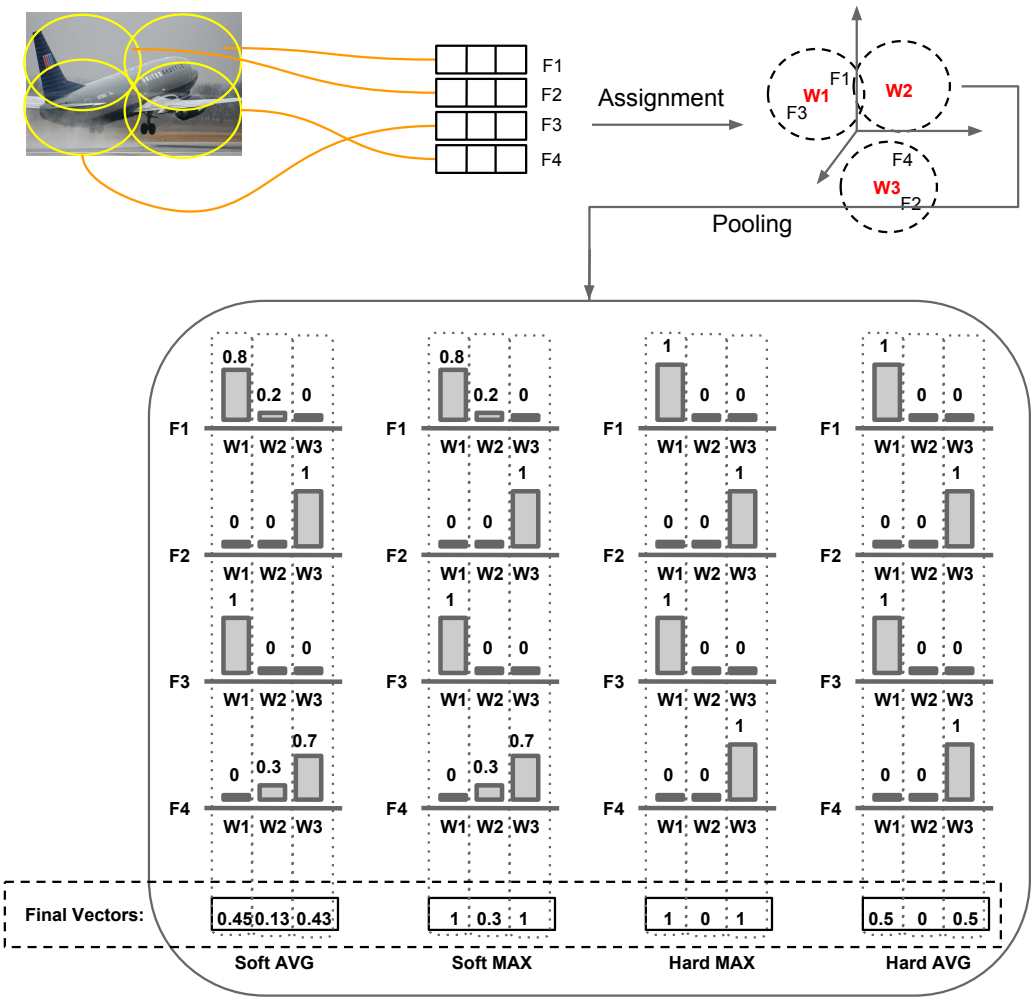


Figure 2.9: Mid-level representation with different assignment/pooling strategies. In hard-assignment just one bin is ativated per feature. In soft-assignment, a group of nearest bins are ativated in a soft manner. We used two pooling strategies: average and maximum

where j varies from 1 to the dictionary size (k), v_i is the feature vector of patch i, w_j is the vector corresponding to visual word j, $K_\sigma(D(x)) = \frac{1}{\sqrt{2\pi}\times\sigma} \times exp(-\frac{1x^2}{2\sigma^2})$, and $D(a,b)$ is the distance between vectors a and b. The σ parameter indicates the smoothness of the Gaussian function: the higher the value, the larger the number of neighboring regions considered [Penatti et al., 2011b].

We proposed to normalize the distances of bag of visual words using soft assignment with **average** pooling using min-max normalization. The equation for min-max normalization is the following:

$$MinMax_{norm}(bin_i) = \frac{bin_i - BoWVector_{min}}{BoWVector_{max} - BoWVector_{min}} \tag{2.2}$$

where i varies from 1 to the dictionary size (k), bin_i is the value of the bag of words vector in the position i, $BoWVector_{max}$ is the maximum value of the bag of words vector and $BoWVector_{min}$ is the minimum value of the bag of words vector.

2.3 Image Feature Compression

Data compression techniques have arisen due to the need of reducing the space required for storage and the time required for the transmission of images [Pedrini and Schwartz, 2008]. Compact representations of image features are important in mobile visual search because of the bandwidth limitations like 2G, 3G and 4G networks and this is a method that help to save battery energy consumption for mobile systems that needs perform content-based image retrieval transferring compact image feature vectors to be processed on the server side (see the mobile visual search architecture used in this work in the Figure 1.1).

The data compression is classified into two categories: lossless and lossy. In lossless compression, the resulting feature vector after decompression process is exactly equal to the original feature vector. In lossy compression, not all information is recovered after unpacking the data.

In this thesis, we have used three approaches of lossless compression: Huffman encoding, Error encoding followed by Huffman and Run-length encoding.

1. **Huffman encoding:** based on frequency of occurrence for each possible value of common symbols which are generally represented using fewer bits than less common symbols. We use the variation of Huffman called byte-oriented Huffman Code [Silva de Moura et al., 2000] where a sequence of bytes is assigned to bin values of a BoW representation. In practice, byte processing is much faster than bit processing (original Huffman code [Huffman et al., 1952]) because bit shifts and masking operations are not necessary in this compression and decompression.

2. **Error enconding:** the difference between bin values. The first bin is the maximum error and have their own value. In this thesis, we applied this technique to try get repeated differences of bin values and get a better compression using Huffman encoding.

3. **Run-length encoding:** A very simple form of data compression that represents sequences of the same data value as a single data value and count. This is most useful on data that contains long sequences of repeated values. This

approach were applied only in the 0's and 1's sequence of Bag of Words using Hard-Assignment with Max Pooling approach (BoW Hard-MAX).

Two approaches of lossy compression were tested in the Bag of Words using Soft-Assignment with Maximum Pooling approach (BoW Soft-MAX):

1. **Soft-MAX Truncated:** Instead of send float values, all numbers expressed in floating-point were truncated and transformed in integer values.

2. **Soft-MAX Truncated using Ranges:** In this approach, we created fixed features vectors with values in ranges of 5 or 10 or 15 or 20 or 25 or 30. We choose values after looking for the minimum and maximum in all values of the features of Soft-MAX. The idea is activate a bin if the value of the Soft-MAX Truncated is next to the central point in the range.

2.4 Image Segmentation

Traditionally, the objective of image segmentation is to partition the image into groups of pixels [Gonzalez and Woods, 2002], called regions. Conventional approaches for image segmentation are usually based on the basic properties of the image of the gray levels, trying to detect discontinuity or similarity image. Thresholding is one of the simplest techniques of the segmentation which classifies pixels in an image according to the specification of one or more thresholds. Some methods seek to partition the image based on abrupt changes in gray levels, characterized by the presence of isolated point, lines or edges in the image (for example, Canny algorithm, Hough transform). Other methods attempt group image points that have similar values for a given set of features (growth regions, dividing regions and watershed techniques) [Pedrini and Schwartz, 2008].

Recently, superpixels have become an essential tool to the vision community. These algorithms group pixels into perceptually meaningful regions, which can be used to replace the rigid structure of the pixel grid. Algorithms for generating superpixels are categorized as graph-based or gradient ascent methods. To create superpixel using a graph-based approach, each pixel is treated as a node in a graph and the similarity between neighboring pixels is the edge weights. After mapping an image in a graph, superpixels are created by minimizing a cost function defined over the graph [Shi and Malik, 2000]. Alternatively, gradient-ascent-based algorithms start from an initial clustering of pixels and iteratively refine the clusters until some convergence criterion is met to form superpixels. In superpixels, the following properties are generally desirable: 1)

they should adhere well to image boundaries, 2) they should be fast to compute, memory efficient, and simple to use, 3) they should both increase the speed and improve the quality of the results. In this thesis, we use a recent superpixel algorithm called SLIC.

2.4.1 Simple Linear Iterative Clustering (SLIC)

Simple Linear Iterative Clustering (SLIC) [Achanta et al., 2012] algorithm simply performs k-means clustering approach in the 5D space of color information and image location to efficiently generate superpixels. The SLIC idea is similar to the approach used as a preprocessing step for depth estimation described in Zitnick and Kang [2007]. According to Achanta et al. [2012], despite its simplicity, SLIC shows the ability to adhere to image boundaries, speed, memory efficiency as well as or better than previous state-of-the-art superpixel algorithms: Shi and Malik [2000]; Felzenszwalb and Huttenlocher [2004]; Vedaldi and Soatto [2008]; Siddiqi [2009]; Veksler et al. [2010]. In summary, SLIC is an adaptation of k-means for superpixel generation where the search space is dramatically reduced and a weighted distance measure combines color and spatial proximity. The main parameters of SLIC are n (number of approximately equally-sized superpixels) and their compactness (c). SLIC uses the $CIElab$ color space. $CIElab$ originates from the three dimensions l, a, and b. l represents lightness whose values run from 0 (black) to 100 (white). a and b are color axes where each axis the values run from positive to negative. On the a axis, positive values indicate amounts of red while negative values indicate amounts of green. On the b axis, yellow is positive and blue is negative. For both axes, zero is neutral gray.

The SLIC clustering begins with k initial cluster centers which are sampled on a regular grid spaced of S pixels. To produce roughly equally sized superpixels, the grid interval is $S = \sqrt{\frac{N}{k}}$, where N is the number of pixels. The centers are moved to seed locations corresponding to the lowest gradient position in a 3×3 neighborhood. Once each pixel has been associated to the nearest cluster center, an update step adjusts the cluster centers to be the mean vector of all the pixels belonging to the cluster. The $L2$ norm is used as a similarity measure to compute the distance in the adaptated k-means proposed by SLIC. By limiting the size of the search region, SLIC algorithm significantly reduces the number of distance calculations. The complexity of SLIC is $O(N)$. SLIC only computes distances from each cluster center to pixels within a $2S \times 2S$ region. On the other hand, the conventional k-means algorithm has the complexity of $O(kNI)$ where I is the number of iterations.

SLIC superpixels correspond to clusters in the $5D$ ($labxy$) color-image plane

space. To combine pixel color information (l, a and b components) with pixel position information (x and y positions) into a single measure, it is necessary to normalize color proximity and spatial proximity by their respective maximum distances within a cluster. The maximum spatial distance expected within a given cluster should correspond to the sampling interval, $N_S = \sqrt{\frac{N}{k}}$. However, determining the maximum color distance N_C is not so straightforward because color distances can vary significantly from cluster to cluster and image to image. This problem can be avoided by fixing N_C to a constant c, where c weighs the relative importance between color similarity and spatial proximity. When c is small, the resulting superpixels adhere more tightly to image boundaries. When c is large, the resulting superpixels have more regular size and shape. The compactness (c) of the superpixels can be controlled in the SLIC algorithm. To sum up, the distance measure D which computes the distance between a pixel i and cluster center C_k is the following:

$$d_C = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \tag{2.3}$$

$$d_S = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{2.4}$$

$$D = \sqrt{\left(\frac{d_C}{N_C}\right)^2 + \left(\frac{d_S}{N_S}\right)^2} \implies D = \sqrt{\left(\frac{d_C}{\mathbf{c}}\right)^2 + \left(\frac{d_S}{N_S}\right)^2} \tag{2.5}$$

Examples of image labels create after use the SLIC algorithm are shown in Figure 2.10. To show the effects of the SLIC parameters, we define the value of n (number of superpixels) as 10 and the value of c (compactness) as 0, 5, 20 or 50.

2.5 Evaluation Metrics

The feature extraction in the mobile devices has a triple trade-off among effectiveness, efficiency, and the feature representation compactness. We use four measures in this thesis to evaluate several algorithms: 1) Curve of Precision versus Recall, 2) Mean Average Precision (MAP), 3) Precision at Top N images, and 4) Compression Ratio (CR). Below, we describe each of these metrics and other key concepts.

Figure 2.10: Image labels of the SLIC algorithm. SLIC has two parameters: n = Number of superpixels and their compactness (c).

Precision is the fraction of the retrieved images which is relevant. High precision means that an algorithm returned substantially more relevant results than irrelevant.

$$\text{Precision} = \frac{|\{\text{Relevant Images}\} \cap \{\text{Retrieved Images}\}|}{|\{\text{Retrieved Images}\}|} \tag{2.6}$$

Recall is the fraction of the relevant images which has been retrieved. High recall means that an algorithm returned most of the relevant results.

$$\text{Recall} = \frac{|\{\text{Relevant Images}\} \cap \{\text{Retrieved Images}\}|}{|\{\text{Relevant Images}\}|} \tag{2.7}$$

The definition of precision and recall assumes that all docs in the "answer set" have been examined. Thus, precision and recall vary as the user proceeds with their examination of the "answer set".

Curve of Precision versus Recall is an appropriate way to analyze the precision and recall together. A high area under the curve represents both high recall and high precision, where high precision relates to a low false positive rate, and high recall

relates to a low false negative rate. In this thesis, we interpolate the precision $p(r)$ as a function of recall r as follows. Let r_j (recall at level j) where $j \in 0, 1, 2, ..., 10$, be a reference to the j-th standard recall level. Then, the precision at the 11 standard recall levels are $P(r_j) = max_{\forall_r | r_j \leq r} p(r)$.

Mean Average Precision (MAP) is the average precision at each recall level r_j for a set of Q queries as follows.

$$\text{MAP} = \frac{\sum_{q=1}^{Q} \text{AveP}(\text{r}_\text{j})}{Q} \tag{2.8}$$

where Q is the number of queries and $AveP(r_j)$ is the precision at recall level r_j for the i-th query. Average Precision ($AveP(r_j)$) computes the average value over the interval from $r = 0$ to $r = 1$. $AveP$ is the area under the precision-recall curve which plots the precision $p(r)$ as a function of recall r.

$$\text{AveP} = \int_0^1 p(r)dr \tag{2.9}$$

This integral is in practice replaced with a finite sum over every position in the ranked sequence of images.

$$\text{AveP} = \frac{\sum_{k=1}^{n} (P(k) \times \text{relevant}(k))}{\text{number of relevant documents}} \tag{2.10}$$

where k is the rank in the sequence of retrieved images, n is the number of retrieved images, $P(k)$ is the precision at level k in the list and $relevant(k)$ is an indicator function that returns 1 if the item at rank k is a relevant image or 0 otherwise.

Precision at Top N images (P@N) provides a single measure of quality across the recall levels. Higher the number of relevant images at the top of the ranking, more positive is the impression of the users. As we are studying representations in the CBIR context, achieving a high precision on the initial images retrieved is important (Figure 2.11). Therefore, most of the time we considered the top 10 images to calculated the $P@N$.

Compression Ratio (CR) is defined as the ratio between the uncompressed size and compressed size. If CR is high, the compression ratio is better because the resulting image representation is smaller. In this aspect, our baseline is the size of all images

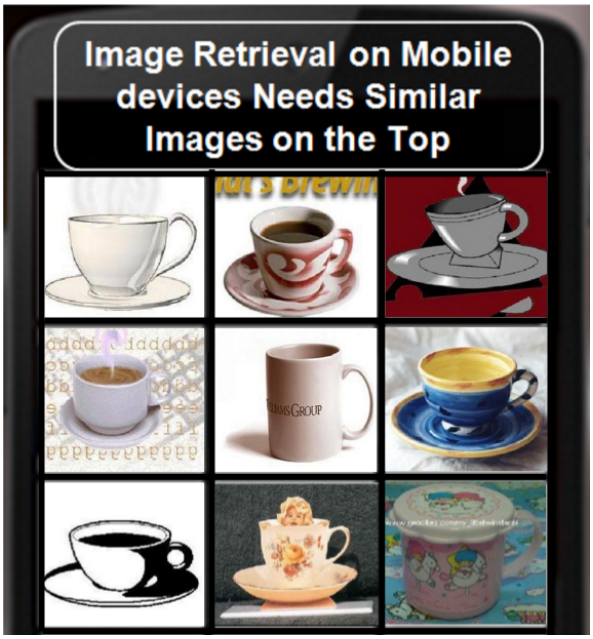


Figure 2.11: In image retrieval on mobile devices is very important to have good precision on the top N images.

in the datasets. Thus, we always divide images size (I) per image representation size ($IRepr$) as follows.

$$CR = \frac{size(I)}{size(IRepr)} \tag{2.11}$$

In conclusion, we have used the $P@5$, $P@10$, $P@15$ metric to evaluate **effectiveness**. To give an overall precision, we report the **effectiveness** using the MAP (Mean Average Precision) metric. The **efficiency** was evaluated by computing the feature extraction and representation time, in seconds. We have used the representation size (in bytes) and the Compression Ratio (CR) as measures for evaluating the **compactness**. Finally, we use Precision $\times$ Recall curves to evaluate image retrieval strategies.

Chapter 3

Benchmark Datasets

In this thesis, we use several benchmark image datasets to compare our methods of image retrieval. These benchmarks provide a common ground for researchers to compare their methods. This chapter provides details regarding each dataset used in this work and discusses how they differ among themselves. Hence, we comment on its challenges, such as intra-class variability, single and multiple label, occlusion, changes on viewpoint, illumination and background clutter. The final objective is to provide realistic scenaries with diferents types of datasets that have been used in the community of CBIR over the last years. The datasets are splitted in four categories: 1) Scenes, 2) Mobile Visual Search (Mobile), 3) Single-label and 4) Multi-label. Table 3.1 summarizes all datasets used in this thesis.

Category	**Dataset**	**N. images**	**N. Classes**	**Measure**
Scenes	01: 15Scenes	4,485	15	MAP, P@10
	02: OxBuild11	567	11	MAP, P@10
	03: Paris	6,392	12	MAP, P@10
	04: ZuBuD	1,005	201	MAP, P@5
Mobile	05: SMVS692	3,460	692	MAP, P@5
Single-label	06: WANG	1,000	10	MAP, P@10
	07: caltech101	9,144	102 (with Background)	MAP, P@10
	08: caltech256	30,607	257 (with Cluster)	MAP, P@10
Multi-label	09: VOC2007	9,963	20	MAP, P@10
	10: UWDataset	1,109	20	MAP, P@10

Table 3.1: Summary of all datasets used in this thesis.

3.1 Scenes Datasets

3.1.1 Fifteen Scene Categories (15Scenes)

The Fifteen Scene Categories (15Scenes) dataset [Lazebnik et al., 2006] contains 4,485 images of 15 indoor and outdoor scene categories. 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. We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Example images are shown in Figure 3.1 and the number of images and labels for each class in 15Scenes dataset are shown on Table 3.2.

Class	**N. images**	**Class**	**N. images**
1: bedroom	216	9: MITinsidecity	308
2: CALsuburb	241	10: MITmountain	374
3: industrial	311	11: MITopencountry	410
4: kitchen	210	12: MITstreet	292
5: livingroom	289	13: MITtallbuilding	356
6: MITcoast	360	14: PARoffice	215
7: MITforest	328	15: store	315
8: MIThighway	260		

Table 3.2: Number of images and labels for each class in Fifteen Scene Categories (15Scenes) dataset [Lazebnik et al., 2006].

3.1.2 Oxford Buildings (OxBuild11)

The Oxford Buildings datset [Philbin et al., 2007] consists of 5,062 images collected from Flickr and divided on seventeen classes (1: All Souls, 2: Balliol, 3: Christ Church, 4: Hertford, 5: Jesus, 6: Keble, 7: Magdalen, 8: Oriel, 9: New Oxford, 10: Trinity, 11: Radcliffe Camera, 12: Cornmarket, 13: Bodleian, 14: Pitt Rivers, 15: Ashmolean, 16: Worcester, 17: Oxford). The collection has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries: 1) Good - A nice, clear picture of the object/building, 2) OK - More than 25% of the object is clearly visible, 3) Bad - The object is not present, 4) Junk - Less than 25% of the object is visible, or there are very high levels of occlusion or distortion.

We used the annotation of the queries "Good" and "OK" to create a subset of the dataset with images contain more than 25% of the interest object. After that, a new dataset with 567 images and 11 classes were created. We call this dataset as

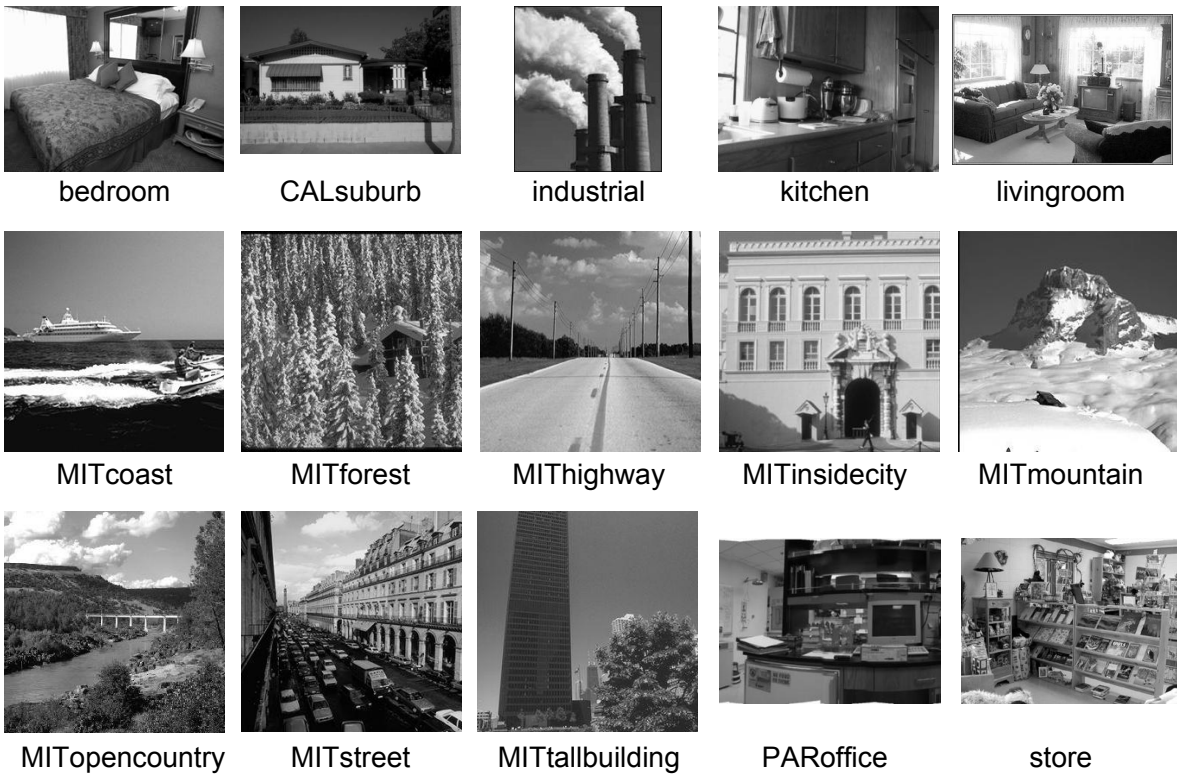


Figure 3.1: Example images from Fifteen Scene Categories (15Scenes) dataset [Lazebnik et al., 2006] with their associated class.

OxBuild11. We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Example images are shown in Figure 3.2 and the number of images and labels for each class in OxBuild11 dataset are shown on Table 3.3.

Class	**N. images**	**Class**	**N. images**
1: All Souls	77	7: Hertford	54
2: Ashmolean	25	8: Keble	7
3: Balliol	12	9: Magdalen	54
4: Bodleian	24	10: Pitt Rivers	6
5: Christ Church	78	11: Radcliffe Camera	221
6: Cornmarket	9		

Table 3.3: Number of images and labels for each class in Oxford Buildings (OxBuild11) dataset [Philbin et al., 2007].

Figure 3.2: Example images from 11 classes annotated from The Oxford Buildings (OxBuild11) dataset [Philbin et al., 2007] with their associated class.

3.1.3 Paris Landmarks (Paris)

Paris Landmark (Paris) dataset is composed of 6,392 images divided into 12 categories of different sizes. Each category represents a monument in the city of Paris, France. We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Example images are shown in Figure 3.3 and the number of images and labels for each class in Paris dataset are shown on Table 3.4.

Class	**N. images**	**Class**	**N. images**
1: defense	381	7: museedorsay	739
2: eiffel	147	8: notredame	444
3: general	1,497	9: pantheon	659
4: invalides	452	10: pompidou	436
5: louvre	266	11: sacrecoeur	377
6: moulinrouge	444	12: triomphe	550

Table 3.4: Number of images and labels for each class in Paris Landmarks (Paris) dataset (11 classes annotated) [Philbin et al., 2008].

3.1.4 Zurich Building (ZuBuD)

The Zurich Building dataset (ZuBuD) database [Shao and Gool, 2004] is a subset of 1,005 images of Zurich city building which have been manually selected and which form 201 classes of 5 images each. The five images in each class have different views with changes on scale such as rotation and translation transformations. We used the precision in the top 5 images (P@5) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Example images are shown in Figure 3.4.

Figure 3.3: Example images from ParisLandmarks (Paris) dataset [Philbin et al., 2008] with their associated class.

Figure 3.4: Example images from Zurich Building (ZuBuD) dataset [Shao and Gool, 2004] with their associated class.

3.2 Stanford Mobile Visual Search (SMVS)

The Stanford Mobile Visual Search (SMVS) dataset [Chandrasekhar et al., 2011] was released in 2011, this dataset not only has the advantage to be obtained from low and high-end camera phones but also has several key properties that turn it into a good data set for mobile applications: The dataset contains rigid objects, this means, is possible to estimate a transformation between reference image and query image. The images were captured within a wide range of lighting conditions and present a foreground-background clutter. Finally, the data set has common perspective distortions (rotation,

scale changes, viewpoint changes).

The SMVS dataset has 3,300 query images for 1,200 distinct classes (objects) grouped into 8 categories as showed in Table 3.5. Due to the variety of categories, this dataset can be used in a wide range of visual search applications like product recognition (CD, DVD, Books, etc), landmark recognition, augmented reality, text recognition or even video recognition.

Category	**Database**	**Query**
1: CD	100	400
2: DVD	100	400
3: Books	100	400
4: Video Clips	100	400
5: Landmarks	500	500
6: Business Cards	100	100
7: Text Documents	100	400
8: Paintings	100	400

Table 3.5: Number of query and database images in the Stanford Mobile Visual Search (SMVS) dataset [Chandrasekhar et al., 2011] for diferent categories.

This dataset has a reference image and query images. Query images were captured with several different camera phones, including some digital cameras. The list of companies and models used is as follows: Apple (iPhone4), Palm (Pre), Nokia (N95, N97, N900, E63, N5800, N86), Motorola (Droid), Canon (G11) and LG (LG300). Query images present wide variations in lighting conditions and foreground and background clutter.

In this work, we divide the classes of the dataset per type of image capture by a different camera. We eliminate the "Landmarks" class, because the landmarks class has just two images per class where one image is the query image and the other is the reference image. On the contrary, the others seven classes have five images per class. So, we selected seven classes (CD, DVD, Books, Video, Business Cards, Text Documents and Paintings) where each image class has five images per class (1 reference image and 4 different acquisition from digital cameras). After all these steps, the new dataset, called here as SMVS692, has 692 classes and 5 images per class with a total of 3,460 images. We used the precision in the top 5 images (P@5) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Examples of query and database images are shown in Figure 3.5.

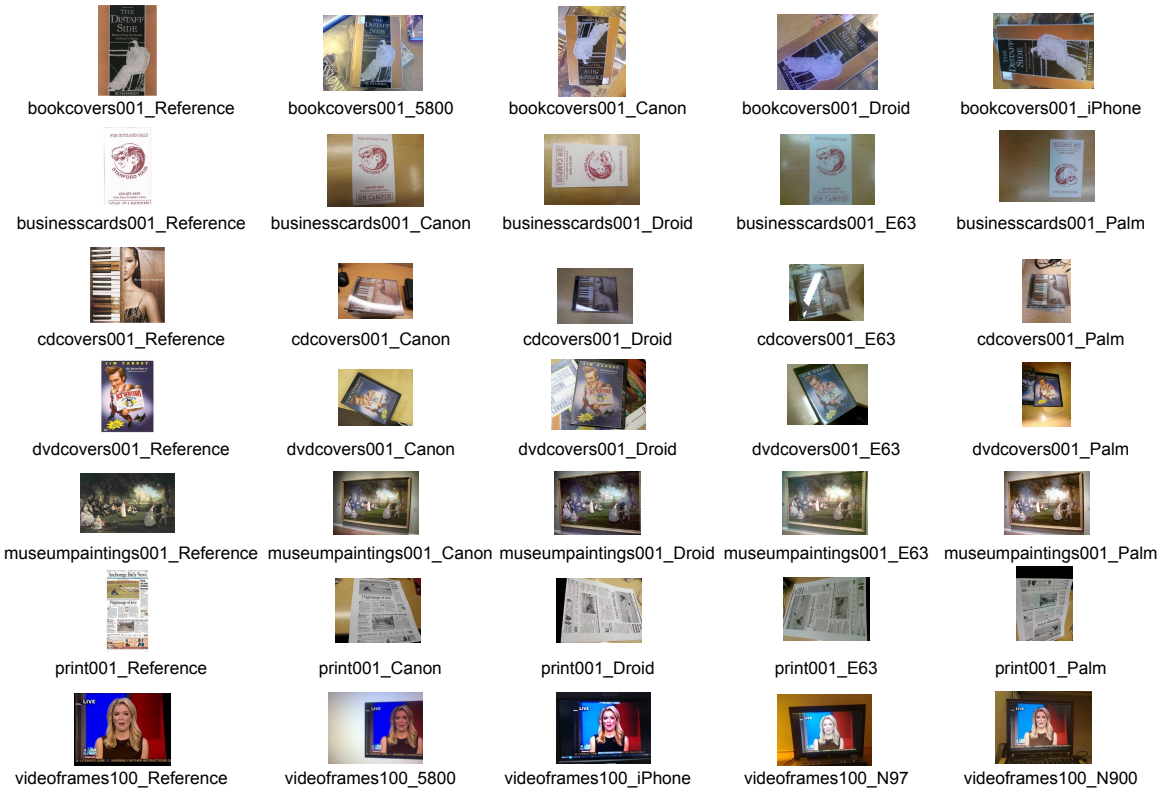


Figure 3.5: Example images from Stanford Mobile Visual Search (SMVS) dataset [Chandrasekhar et al., 2011].

3.3 Single-label Datasets

3.3.1 WANG

The WANG database is a subset of 1,000 images which have been manually selected and which form 10 classes of 100 images each. The WANG database can be considered similar to common stock photo retrieval tasks with several images from each category and a potential user having an image from a particular category and looking for similar images. The ten classes are used for relevance estimation: given a query image, it is assumed that the user is searching for images from the same class, and therefore the remaining 99 images from the same class are considered relevant. We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Some example images are shown in Figure 3.6. Table 3.6 summarizes the number of images for each class.

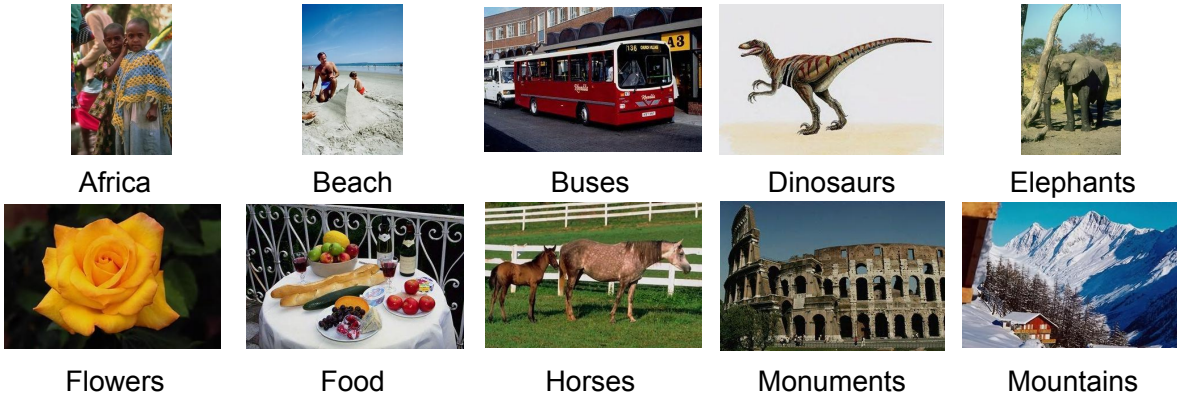


Figure 3.6: Example images from WANG dataset [Wang et al., 2001] with their associated class.

Class	N. images	Class	N. images
1: Africa	100	6: Flowers	100
2: Beach	100	7: Food	100
3: Buses	100	8: Horses	100
4: Dinosaurs	100	9: Monuments	100
5: Elephants	100	10: Mountains	100

Table 3.6: Number of images and labels for each class in WANG dataset [Wang et al., 2001].

3.3.2 Caltech 101

Caltech 101 [Fei-Fei et al., 2007] is a challenge dataset with pictures of objects belonging to 101 categories and a "Background" class to difficult the retrieval. This database was obtained using Google Image Search, and the images contain significant clutter, occlusions, and intra-class appearance variation. Caltech 101 has about 40 to 800 images per category. Most categories have about 50 images. Caltech101 has 9,144 images and just one object per image. We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Some images are shown in 3.7.

3.3.3 Caltech 256

Caltech 256 [Griffin et al., 2007] is a challenge dataset with pictures of objects belonging to 256 categories and a "Cluster" class to difficult the retrieval. The main problem of this dataset is that it has a very large intraclass variability. In addition, this dataset represent a diverse set of lighting conditions, poses, backgrounds, image sizes and camera systematics. Caltech 256 has 30,607 images and just one object per image.

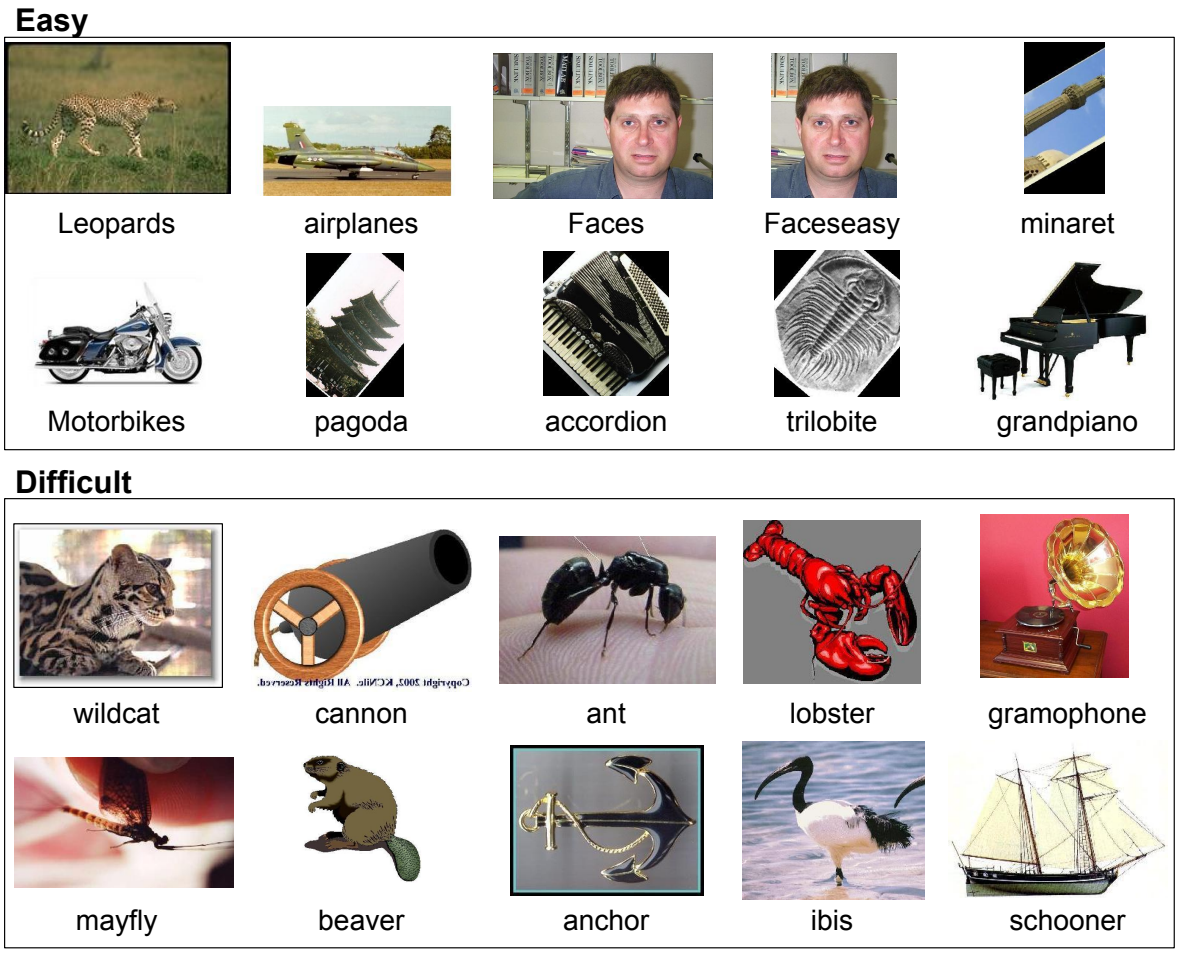


Figure 3.7: Example images from Caltech101 dataset [Fei-Fei et al., 2007]. The top 10 images got high precision (P@10) with the the representation DEOBSM (DEnse sampling, ORB descriptor, bag of visual words with Soft assignment and Maximum pooling). The botton 10 images got low precision (P@10) with the representation DEOBSM.

We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Some images are shown in 3.8.

3.4 Multi-label Datasets

3.4.1 PASCAL Visual Object Classes 2007 (VOC 2007)

The PASCAL Visual Object Classes 2007 (VOC2007) [Everingham et al., 2010] 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. Those object classes are categorized as person, animal, vehicle, and indoor objects. In

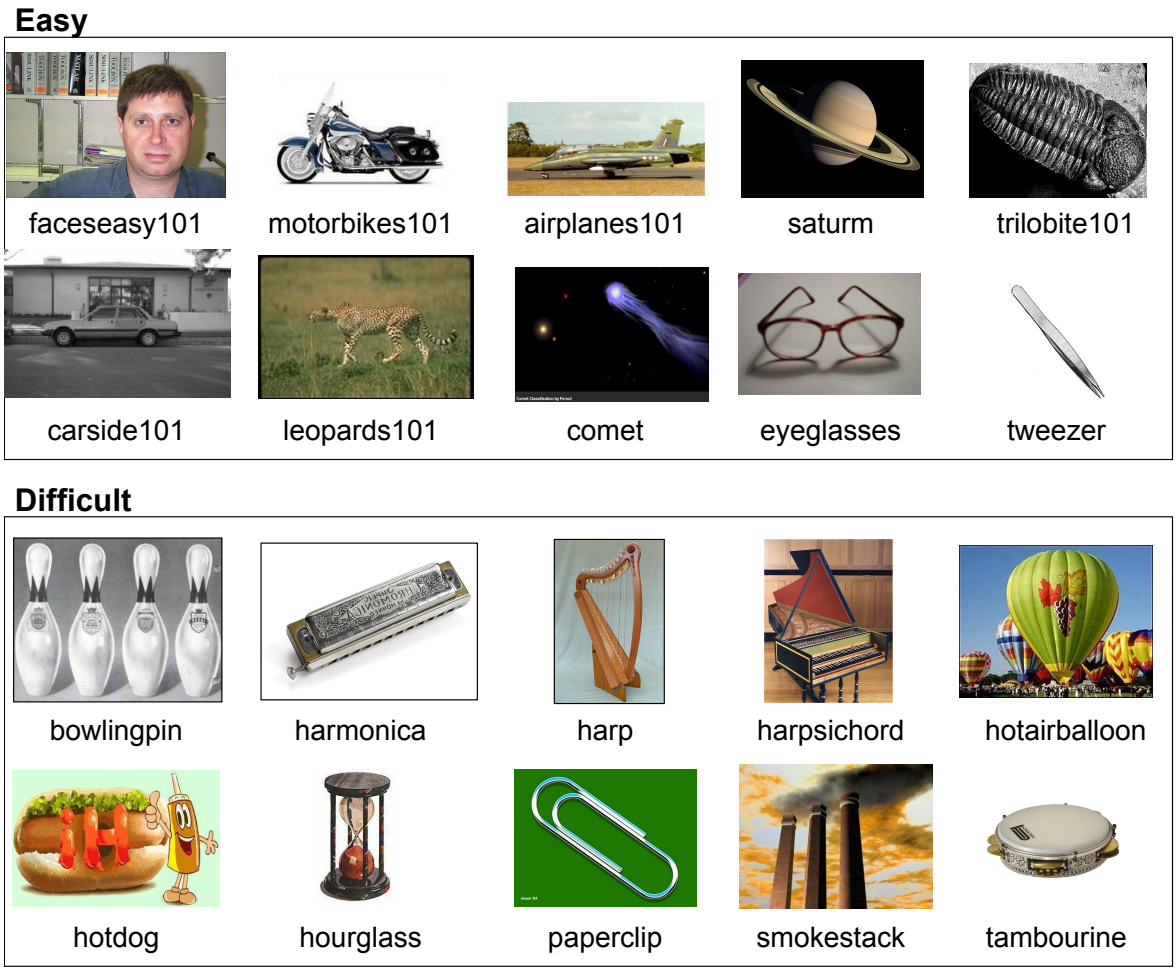


Figure 3.8: Example images from Caltech256 dataset [Griffin et al., 2007]. The top 10 images got high precision (P@10) with the the representation DEOBSM (DEnse sampling, ORB descriptor, bag of visual words with Soft assignment and Maximum pooling). The botton 10 images got low precision (P@10) with the representation DEOBSM.

total, there are 9,963 images. We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. Some example images are shown in Figure 3.9. VOC2007 dataset contains significant variability in terms of object size, orientation, pose, illumination, position and occlusion.

The PASCAL VOC 2007 dataset is an image classification benchmark, separated in images for tranning, validation and test, but we use this dataset to validate our image retrieval application on images with multiple labels. There are others recent versions of the PASCAL Visual Object Classes (VOC 2009, VOC 2010, VOC 2011 and VOC 2012), but just the version 2007 has all images from traning, validation and test available. The other aforementioned versions provide just tranning and validation

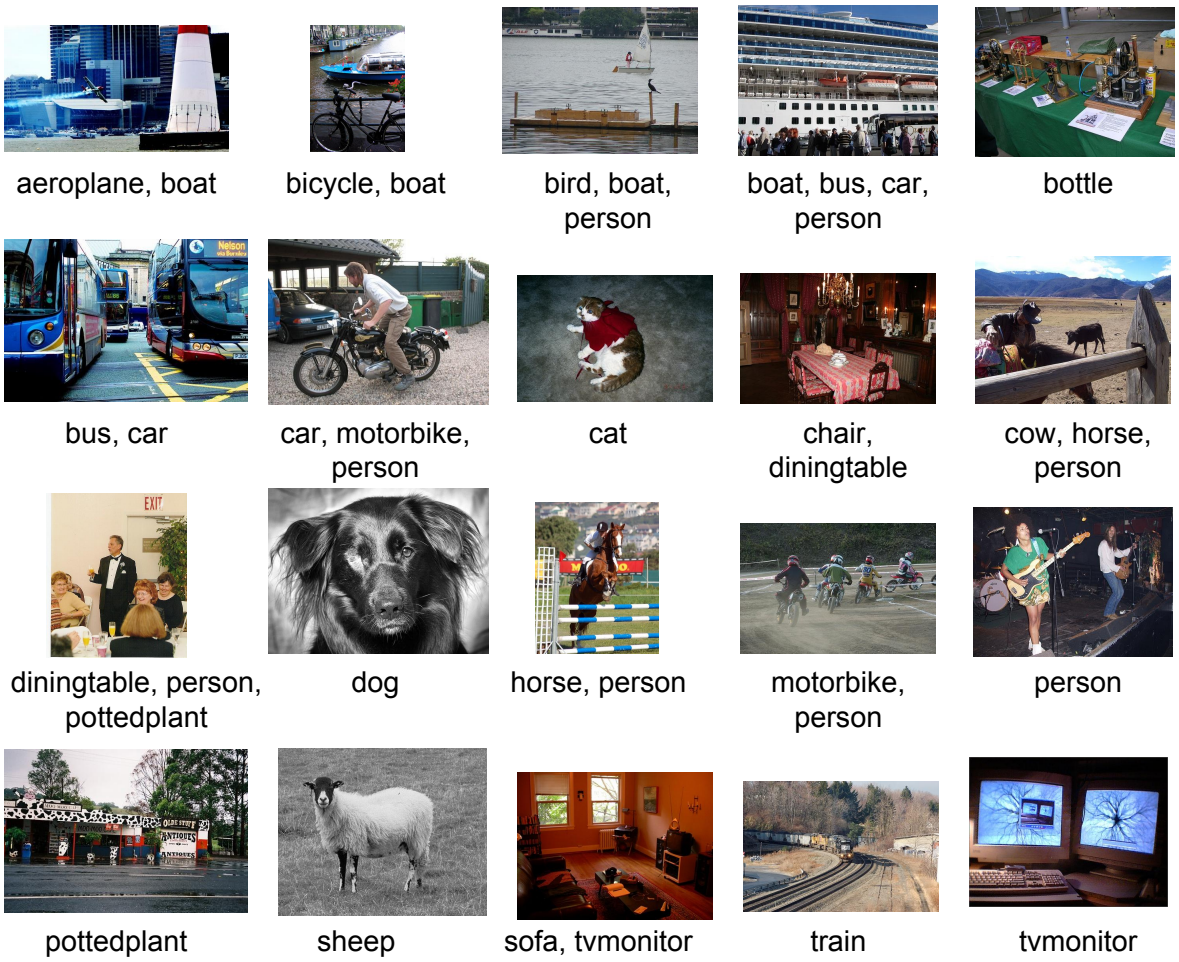


Figure 3.9: Example images from PASCAL Visual Object Classes 2007 (VOC2007) dataset [Everingham et al., 2010] with their associated classes.

images freely. The dataset is freely available[1]. Table 3.7 summarizes the number of images for each class.

3.4.2 University of Washington dataset (UWdataset)

The University of Washington dataset [Deselaers et al., 2008] consists of a collection of 1,109 images. These images are partly annotated using keywords and the remaining images were annotated by a group of the University of Washington. The annotations are publicly available[2]. The complete annotation consists of 6,383 words with a vocabulary of 328 unique words (eliminating stop words like "and", "of"). On the average, each image has about 5.8 words of annotation. The maximum number of key-words per

[1] Acessed on November 2015: `http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/`

[2] Acessed on November 2015: `http://www-i6.informatik.rwth-aachen.de/~deselaers/uwdb/index.html`

Class	N. images	Class	N. images
1: aeroplane	445	11: diningtable	93
2: bicycle	505	12: dog	748
3: bird	622	13: horse	527
4: boat	350	14: motorbike	379
5: bottle	493	15: person	1,275
6: bus	353	16: pottedplant	213
7: car	1,308	17: sheep	151
8: cat	654	18: sofa	124
9: chair	919	19: train	396
10: cow	254	20: tvmonitor	154

Table 3.7: Number of images and labels for each class in PASCAL Visual Object Classes 2007 (VOC 2007) dataset [Everingham et al., 2010]

image is 23 and the minimum is 1. The smallest category contains 22 images and the largest contains 255 images. Details of the labels are on the Table 3.8.

Class	N. images	N. labels	Class	N. images	N. labels
1: arborgreens	47	196	11: greenland	255	1286
2: australia	30	105	12: indonesia	36	123
3: barcelona	48	209	13: iran	49	159
4: campusinfall	48	382	14: italy	22	77
5: cannonbeach	48	360	15: japan	45	136
6: cherries	55	418	16: leaflesstrees	48	261
7: columbiagorge	83	379	17: sanjuans	48	333
8: football	48	826	18: springflowers	48	262
9: geneva	25	138	19: swissmountains	30	127
10: greenlake	48	295	20: yellowstone	48	360

Table 3.8: Number of images and labels for each class in University of Washington (UWdataset) dataset [Deselaers et al., 2008]

The images have several sizes and include vacation pictures from various countries. There are 20 categories, for example "aborgreens", "indonesia", and "yellowstone". We used the precision in the top 10 images (P@10) and the Mean Average Precision (MAP) to evaluate the precision on this dataset. To calculate the P@10 and MAP we use just the image class, and not their tags. Some example images with annotations are shown in Figure 3.10. This dataset is commonly used for personal photo retrieval task, where the object is looking for images from the same vacation, scenes, building, and so on.

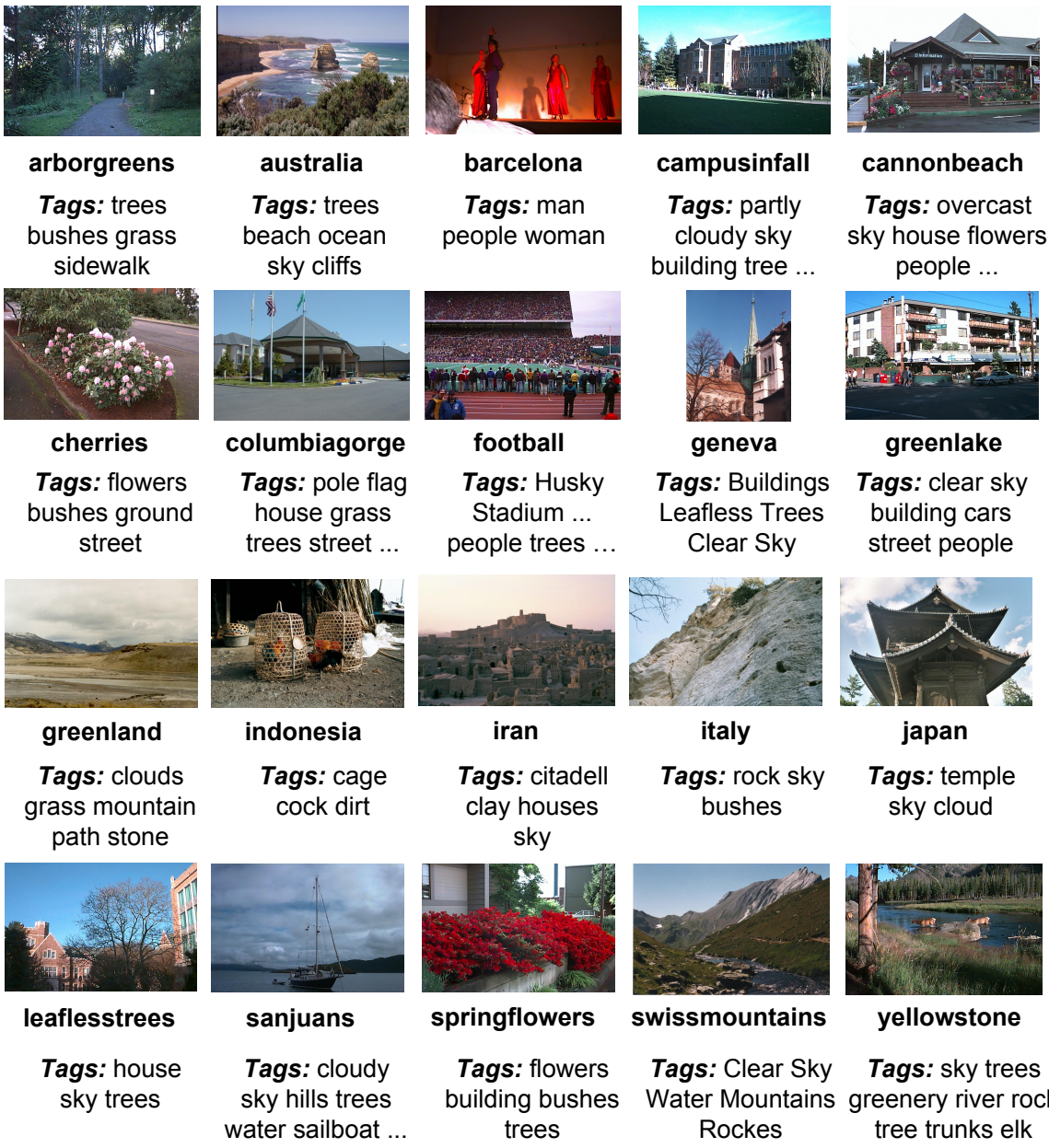


Figure 3.10: Example of images from University of Washington (UWdataset) dataset [Deselaers et al., 2008] and their tags.

Chapter 4

Low-Cost Representation for Mobile Image Search

In this chapter, we deal with the *feature extraction triple trade-off problem* (efficiency, effectiveness and compactness) in mobile devices by evaluating low-cost feature representations. We concentrate our efforts in four main fronts: (1) binary low-level descriptor selection; (2) mid-level representation; (3) low-level global representation analysis and (4) feasibility analysis of data compression techniques. We are interested in balancing computational cost, precison, and feature representation size. In this sense, binary descriptors are considerable options because they provide effective and compact representation [Ascenso and Pereira, 2013]. Mid-level representations based on Bag of Visual Words (BoVW, or just BoW) are good alternatives since it provides effective features and compacted in comparison with the amount of local features extracted [Chatzilari et al., 2013]. Finally, we present global descriptors (color, texture, and shape) analysis as an alternative for mid-level representation, as well as, image features compression techniques.

4.1 Related Work

Content-Based Image Retrieval (CBIR) applications on mobile devices has been typically modeled using a client-server architecture [Girod et al., 2011; Chen et al., 2013; Zhou et al., 2014]. In Girod et al. [2011], the authors present the Stanford Product Search system, a low latency interactive visual search system. They use interest point detection, Compressed Histogram of Gradients (CHoG) descriptor [Chandrasekhar et al., 2009] and a mid-level representation. A CHOG descriptor with 60 bits matches the performance of a SIFT descriptor with 4,096 bits (128-dimensional

SIFT descriptor × 32) [Chandrasekhar et al., 2010a]. However, due to the complexity of spatial sub-block assignment scheme, the extraction of the CHOG is not fast enough. In addition, the quality of features is also influenced by the detection of interest points, which does not receive much attention in CHOG.

In Chen et al. [2013], the authors develop a discriminative image signature called the Residual Enhanced Visual Vector (REVV). They have utilized REVV to design and construct a mobile augmented reality system for landmark recognition. REVV uses features of SURF [Bay et al., 2008] or CHoG [Chandrasekhar et al., 2009] descriptors. They also use mid-level representation and aggregation methods (Mean and Median). In Zhou et al. [2014], they propose a codebook algorithm for large scale mobile image search. Their method, first employs a scalable cascaded hashing scheme to ensure the recall rate of local feature matching. They use the algorithm in a CBIR application where PCA is used for dimension reduction on SIFT features. Different feature quantization and hashing schemes are used in BoW approach.

Due to the high computational cost of traditional approaches as SIFT, SURF, CHOG descriptors, it has been made a great efforts in the evaluation of low-cost descriptors, such as, binary descriptors. Chatzilari et al. [2013] perform a comparative study of approaches for mobile visual recognition and use machine learning in a mobile environment. They use keypoint detection, two binary descriptors (BRIEF, ORB) and two non-binary descriptors (SURF, SIFT). They also use mid-level representations on PASCAL VOC 2007 dataset. In Ascenso and Pereira [2013], lossless compression of binary image features (BRIEF, BRISK, FREAK, ORB) is proposed to be used in a mobile CBIR enviroment. The coding solution exploits the redundancy between descriptors of an image by sorting the descriptors and applying Differential Pulse Coded Modulation (DPCM) and arithmetic coding. They do not use mid-level representation, just apply lossless compression on binary features before sending them to a server side. Each binary descriptor is computed from a patch around a detected keypoint. They propose a lossless predictive coding scheme for binary features. Zhuang et al. [2014] address the problem of Content-Based Image Retrieval system by using binary features (BRIEF, ORB, BRISK, and FREAK). They use spatial context information in image retrieval process in the local region around interest points and use BoW representations.

4.2 Analyzes of Low-Cost Representations

Our work differs from the literature in several aspects. First, it is worthwhile mentioning that to the best of our knowledge, there are no works in the literature that

evaluate (or use) low-cost mid-level representation based on dense sampling in mobile applications. It has been shown that dense sampling is more accurate than interest point detection to compute bag-of-visual-word features [Ken Chatfield and Zisserman, 2011; Penatti et al., 2014]. Also, there is no work that evaluate the state-of-the-art binary descriptor called BinBoost [Trzcinski et al., 2013] for image feature extraction on mobile devices. We use ten color descriptors, five texture descriptors and two shape descriptors in an approach of image retrieval on mobile devices – an extensive comparative study of the global descriptors used in this work is shown on Penatti et al. [2012]. Finally, we evaluate compression techniques and different assignment/pooling strategies in order to obtain compact representations. Figure 4.1 shows a summary of the analyzes of low-cost representations for mobile image search which will be presented in the next sections.

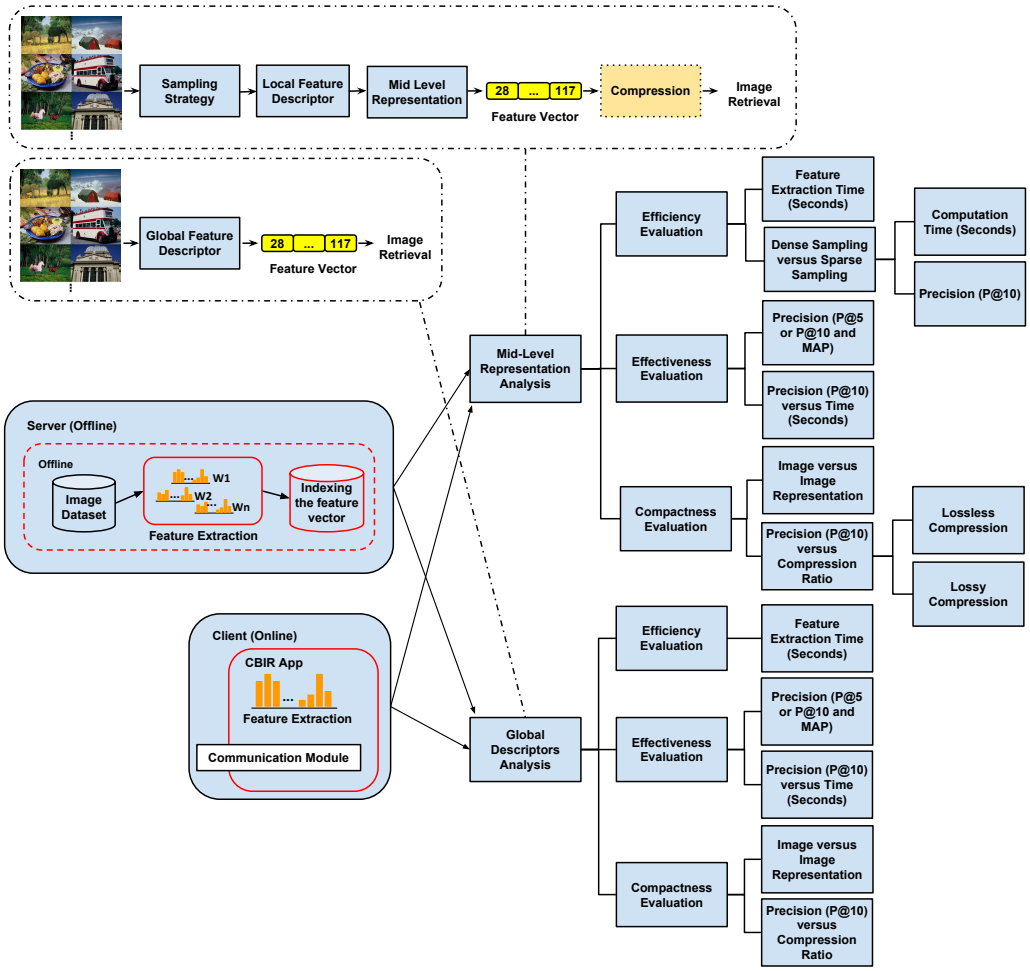


Figure 4.1: Analyzes of low-cost representations for mobile image search.

4.3 Mid-level Representation Analysis

We aim at evaluating binary low-level descriptors in different mid-level representations to find the most suitable configurations in terms of effectiveness, efficiency, and compactness. Although the descriptors could be used for other applications, we have adopted a content-based image retrieval (CBIR) application protocol.

4.3.1 Experimental Setup

We have used a **dense sampling** strategy to select points for low-level feature extraction, five binary descriptors are used to encode low-level local properties: (1) BRIEF [Calonder et al., 2010], (2) ORB [Rublee et al., 2011], (3) BRISK [Leutenegger et al., 2011], (4) FREAK [Ortiz, 2012] and (5) BinBoost [Trzcinski et al., 2013] and bag of words representation using two word assignment strategies (hard and soft assignment) with two different pooling approaches (average and maximum). The experiments were conducted in ten public available image datasets: (1) 15Scenes [Lazebnik et al., 2006], (2) Caltech 101 [Fei-Fei et al., 2007], (3) Caltech 256 [Griffin et al., 2007], (4) OxBuild11 [Philbin et al., 2007], (5) Paris [Philbin et al., 2008], (6) SMVS [Chandrasekhar et al., 2011], (7) UWDataset [Deselaers et al., 2008], (8) VOC2007 [Everingham et al., 2010], (9) WANG [Wang et al., 2001] and (10) ZuBuD [Shao and Gool, 2004].

For all binary descriptors, except for BinBoost[1], we obtained the code from the OpenCV repository [Bradski, 2000] – version 2.4.10. We use the default setting for the OpenCV descriptors (BRIEF, BRISK, FREAK[2], ORB). In addition, we set the length in bytes of the BinBoost descriptor to 16 bytes (it can be equal to 8, 16 or 32 bytes). All these configurations are also used in Caetano et al. [2014b].

To learn the codebooks, we apply a k-medians clustering algorithm with Hamming distance over all sampled descriptors, as Caetano et al. [2014b]. We created one dictionary for each binary descriptor (BRIEF, BRISK, FREAK, ORB, BinBoost) without compression and with lossless and lossy compression. The parameters for dictionary generation and image representation are the same: dense sampling (6 pixels) as in van de Sande et al. [2010]; Penatti et al. [2014]; Caetano et al. [2014a], and 1024 visual words, as in Zhou et al. [2010b]; Ken Chatfield and Zisserman [2011]; Caetano et al. [2014b] (See Attachment B).

[1]The reference implementation of BinBoost is available at `http://cvlab.epfl.ch/research/detect/binboost` - Last access on: December, 2015

[2]For FREAK descriptor which we have to use 22 as the 'Scaling of the description pattern' because when we use the default value, this descriptor is not enable to describe some images from Caltech-101 and Pascal VOC 2007 which have low height.

We have implemented two approaches of pooling (average and maximum) in two different approaches of assignment (hard and soft). The retrieval performance is measured by P@10 (precision at top 10) for all datasets, except SMVS692 and ZuBuD where we use the P@5 (precision at top 5), because they have just 5 images per classe. The ranking of the CBIR process is created using the Manhattan distance ($L1$ distance) – (See Appendix A).

4.3.2 Efficiency Evaluation

We divide the efficiency analysis of mid-level representation with binary descriptors in two parts: (1) time spent for feature extraction and (2) sampling strategies [3].

4.3.2.1 Time spent for feature extraction

Figure 4.2 presents the computational time required for each descriptor to extract features for all images in Caltech101 and VOC2007 datasets combined. According to the results, while BinBoost and BRISK are the most expensive descriptors, BRIEF, FREAK and ORB can be considered good choices since they are very fast. For this experiment, we extracted each descriptor five times per image and the results are reported with a confidence of 95% (α=0.05).

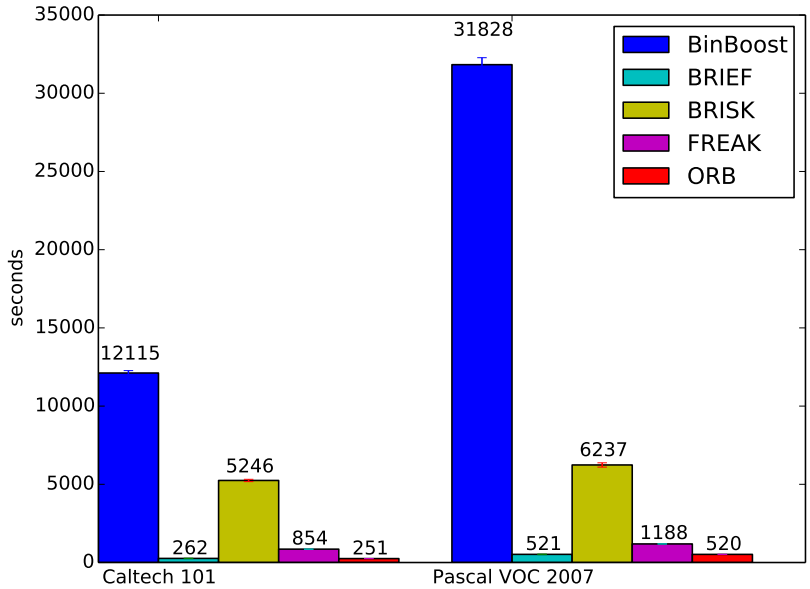


Figure 4.2: Time (in seconds) spent for feature extraction of all images of Caltech 101 (9,144 images) and VOC 2007 (9,963 images) using different descriptors.

[3] Although we are in the efficiency subsection, we analyze the time (efficiency) and also the accuracy (effectiveness) to make the comparison of sampling strategies.

4.3.2.2 Dense Sampling Versus Sparse Sampling

The first step in the matching process and image representation using low-level and mid-level representations is the sampling step (dense or sparse). Keypoint detectors look for points in images with properties such as repeatibility, which may create less ambiguity and are in discriminative regions [Tuytelaars and Mikolajczyk, 2008]. On the other hand, dense sampling, gives a better coverage of the entire object or scene and a constant amount of features per image area.

In this subsection, we analyze dense and sparse sampling. To find the best sampling strategy to use in a mobile visual retrieval approach, we use a dense configuration against six keypoint detectors (FAST, GFTT, GFTTHarris, MSER, ORB detector, SURF detector). We use two images datasets: WANG (1,000 images) and 15Scenes (4,485 images). In the experiment, we use soft assignment with maximum pooling to create bag of words representation of five binary descriptors (BinBoost, BRIEF, BRISK, FREAK, ORB) using 1024 visual words and compare them with bag of words using dense sampling.

We performed several tests with dense and sparse sampling. Figure 4.3 (15Scenes dataset) and Figure 4.4 (WANG dataset) shows the best combination of binary descriptor with a keypoint detector and compare them with the results of dense sampling. In both graphs (Figures 4.3 and 4.4), we show just the best combination of binary descriptor with keypoint detector. We have analized all binary descriptors (BinBoost, BRIEF, BRISK, FREAK, ORB) with all keypoint detector (GFTT, GFTTHarris, MSER, ORBDetector, SURFDetector). In the graphs, "ALL" (that means all keypoints) may have more points than 100% of the points in the dense sampling. The labels “25%”, “50%”, “75%” and “100%” means that the keypoint detector is using the number of points of the respective percentage of dense. When the line in the graph is not complete (like “GTBKSM” line), it means that the sparse sampling points is less than the points in the dense sampling.

In conclusion, dense sampling showed to be better than sparse sampling in the dataset WANG and 15Scenes. Other alternatives are use FTOBSM (FAST using bag of words of ORB descriptors using Soft assignment and Maximum pooling) or FTBKSM (FAST using bag of words of BKISK descriptors using Soft assignment and Maximum pooling). However, as we showed in the previous analysis of time spent for feature extraction, BRISK descriptor is time consuming. Therefore, we have two options: DEOBSM (dense sampling) and FTOBSM (sparse sampling). Paired statistical test (t-student distribution) with 95% of confidence showed that DEOBSM (bag of words using DEnse sampling, ORB descriptor, Soft assignment and Maximum pooling) is

better then FTOBSM (bag of words using FAST sparse sampling, ORB descriptor, Soft assignment and Maximum pooling).

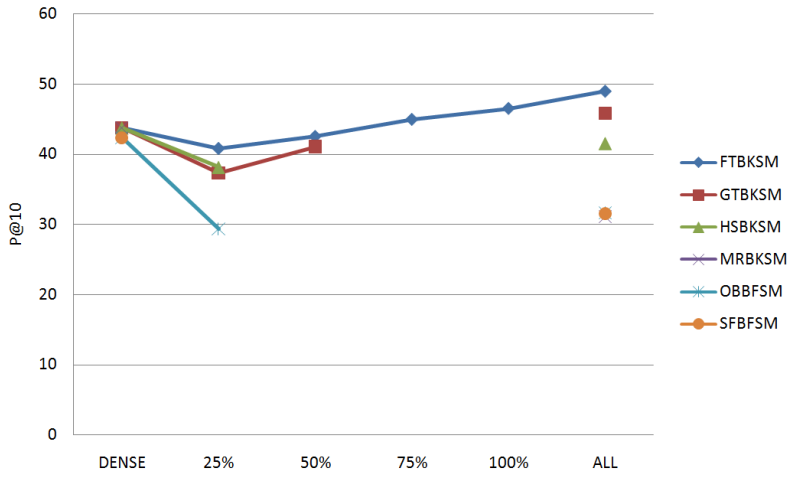


Figure 4.3: 15Scenes dataset: Best combination of binary descriptor with a keypoint detector versus binary descriptor with dense sampling. Keypoint detectors: FT = FAST, GT = GFTT, HS = GFTTHarris, MR = MSER, OB = ORBDetector, SF = SURF. Descriptors: BK = BRISK, OB = ORB, BF = BRIEF. SM = BoW Soft-MAX.

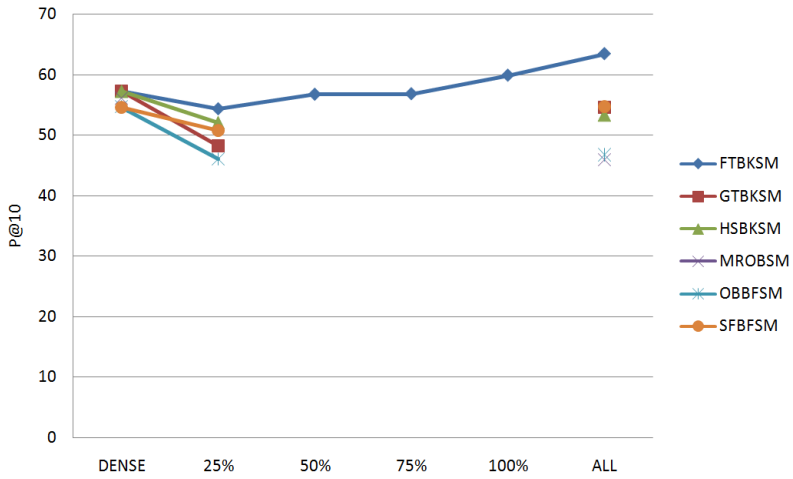


Figure 4.4: WANG dataset: Best combination of binary descriptor with a keypoint detector versus binary descriptor with dense sampling. Keypoint detectors: FT = FAST, GT = GFTT, HS = GFTTHarris, MR = MSER, OB = ORBDetector, SF = SURF. Descriptors: BK = BRISK, OB = ORB, BF = BRIEF. SM = BoW Soft-MAX.

Figure 4.5 shows the time to compute the bag of words DEBFSM (DEnse sampling, BRIEF, Soft-MAX), DEOBSM (DEnse Sampling, ORB, Soft-MAX), FT25dBKSM (FAST 25% of dense sampling, BRISK, Soft-MAX), FT50dBKSM (FAST 50% of dense sampling, BRISK, Soft-MAX), FTBKSM (FAST sparse sampling,

BRISK, Soft-MAX) on 15Scenes (4,485 images) and WANG (1,000 images) dataset. For this experiment, we extracted each descriptor five times per image in a total of 1,000 images for WANG dataset and a total of 4,485 images for 15Scenes dataset and the results are reported with a confidence of 95% (α=0.05). In summary, dense sampling (BRIEF or ORB) is fast than FAST sparse sampling (BRISK).

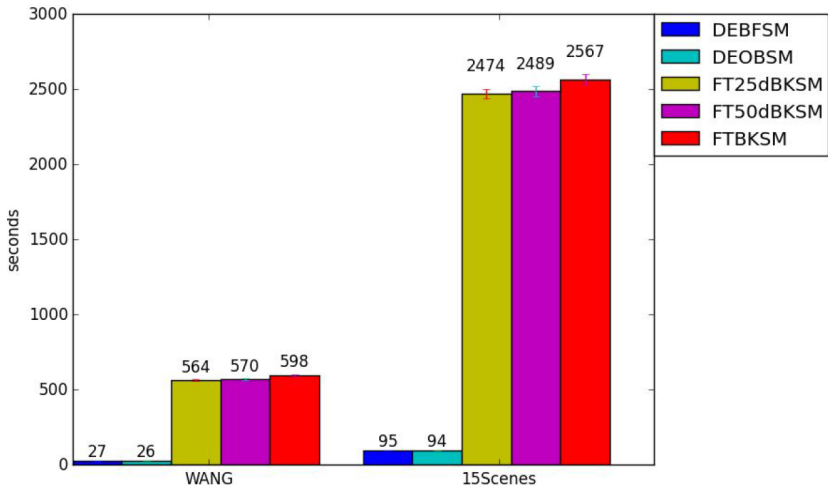


Figure 4.5: Time to compute DEBFSM, DEOBSM, FT25dBKSM, FT50dBKSM, FTBKSM descriptors on 15Scenes (4,485 images) and WANG (1,000 images) dataset.

Restriction of dense sampling To use dense sampling on mobile devices, the images should not be so big, because if the image is big the dense sampling is slow. For this reason, is important resize images extremely big. Another option is to perform a more sparse dense sampling. The Table 4.1 shows the small, large and average images size of all datasets used in this thesis. We also show the average size reported in the dataset papers (if exists) and if we resize the images.

4.3.3 Effectiveness Evaluation

Tables 4.2 (Scenes and Mobile datasets) and 4.3 (Single-label and Multi-label datasets) show the P@10 (or P@5), with a confidence of 95%, for each binary descriptor with four different BoW-based mid-level representations. We use P@5 just to SVMS692 and ZuBuD. The best descriptor in terms of effectiveness, efficiency, compactness is DEOBSM (bag of words using DEnse sampling, ORB descriptor, Soft assignment and Maximum pooling). To see the complete experimental results (P@5, P@10, MAP), see Appendix C (MAP) and Appendix D (P@5/P@10).

For almost all datasets, the Soft assignment with Max Pooling (Soft-MAX) achieved the best results for all tested descriptors, except for the FREAK descrip-

Dataset	Small	Dimension	Large	Dimension	Avg size reported	Resize?
15Scenes	4 KB	256 x 256	93 KB	509 x 220	-	No
caltech101	3 KB	243 x 170	338 KB	1024 x 1024	300 x 200	No
caltech256	2 KB	110 x 110	3.45 MB	1537 x 2181	-	No
OxBuild11	18 KB	300 x 200	18 KB	283 x 212	1024 x 768	Yes
Paris	5 KB	199 x 301	577 KB	200 x 300	1024 x 768	Yes
SMVS	4 KB	226 x 265	101 KB	200 x 300	-	Yes
UWdataset	31 KB	768 x 504	93 KB	765 x 504	-	No
VOC2007	10 KB	500 x 214	93 KB	460 x 500	-	No
WANG	7 KB	384 x 256	56 KB	384 x 256	384 x 256	No
ZuBuD	21 KB	320 x 240	165 KB	640 x 480	-	No

Table 4.1: Images size of all datasets used in this thesis.

tor on the 15Scenes, caltech256, UWDataset and WANG datasets. Soft-AVG and Hard-AVG mid-level representation also yield very high results for some descriptors.

Although, in the 15Scenes, caltech256, UWDataset and WANG datasets, FREAK descriptor with Hard-AVG or Soft-AVG has high precision (P@10) compared to others bag of words strategies of FREAK, all of them yields low precision (P@10) comparing with the other binary descriptors.

To evaluate each descriptor in both effectiveness and efficiency aspects, we present the scatter plots which show the relation between P@10 and time in seconds (in log scale) for the best descriptors in the Caltech 101 (Figure 4.6) and VOC 2007 (Figure 4.7) datasets. In this scenario, the most suitable representations are “BRIEF + Soft-MAX" and “ORB + Soft-MAX" for both Caltech 101 and VOC 2007. In Pessoa et al. [2015a], we obtained similar results using Euclidean distance (Figures 4.8 (caltech101) and 4.9 (VOC2007)).

4.3.4 Compactness Evaluation

In this section, we evaluate the feature representation compactness aiming at finding the most suitable descriptors concerning their feature vector size. As it can be seen in Figure 4.10, it is better to transfer mid-level representation to be processed in the server side instead of images or low-level features because mid-level representation are more compact.

Lossless Compression: Figures 4.11 (caltech101) and 4.12 (VOC2007) present the relation between P@10 and Compression Ratio (CR) for the most suitable feature representations in the Caltech 101 and Pascal VOC 2007 datasets, respectively. According

Dataset	Descriptor	Soft-AVG	Soft-MAX	Hard-MAX	Hard-AVG
15Scenes (P@10)	BRIEF	28.25 ± 0.03	$\mathbf{42.33 \pm 0.08}$	28.99 ± 0.07	33.55 ± 0.007
	BRISK	40.05 ± 0.06	$\mathbf{43.74 \pm 0.09}$	30.96 ± 0.06	35.08 ± 0.07
	FREAK	$\mathbf{29.87 \pm 0.05}$	27.81 ± 0.05	24.64 ± 0.05	28.47 ± 0.07
	ORB	28.18 ± 0.04	$\mathbf{43.88 \pm 0.08}$	29.12 ± 0.06	35.54 ± 0.07
OxBuild11 (P@10)	BRIEF	23.45 ± 0.09	$\mathbf{45.21 \pm 0.14}$	31.61 ± 11.2	31.72 ± 0.12
	BRISK	36.58 ± 0.13	$\mathbf{50.97 \pm 0.12}$	34.41 ± 0.11	33.72 ± 0.11
	FREAK	32.9 ± 0.11	$\mathbf{38.67 \pm 0.1}$	30.67 ± 0.1	25.81 ± 0.1
	ORB	23.87 ± 0.09	$\mathbf{46.29 \pm 0.13}$	31.24 ± 0.1	31.34 ± 0.11
Paris (P@10)	BRIEF	21.26 ± 0.03	$\mathbf{34.5 \pm 0.05}$	23 ± 0.04	25.93 ± 0.04
	BRISK	27.79 ± 0.04	$\mathbf{34.46 \pm 0.04}$	22.75 ± 0.03	25.34 ± 0.04
	FREAK	25.34 ± 0.03	$\mathbf{27.67 \pm 0.04}$	21.98 ± 0.03	22.85 ± 0.04
	ORB	22.05 ± 0.03	$\mathbf{32.9 \pm 0.05}$	22.18 ± 0.03	25.7 ± 0.04
ZuBuD (P@5)	BRIEF	39.18 ± 0.02	$\mathbf{71.18 \pm 0.03}$	27.32 ± 0.02	57.23 ± 0.03
	BRISK	62.33 ± 0.03	$\mathbf{70.11 \pm 0.03}$	23.52 ± 0.01	57.77 ± 0.03
	FREAK	59.12 ± 0.03	$\mathbf{62.11 \pm 0.03}$	25.57 ± 0.01	61.07 ± 0.03
	ORB	37.77 ± 0.02	$\mathbf{70.01 \pm 0.03}$	25.01 ± 0.01	55.48 ± 0.03
SMVS692 (P@5)	BRIEF	11.21 ± 0.01	$\mathbf{19.09 \pm 0.01}$	13.66 ± 0.01	13.66 ± 0.01
	BRISK	14.1 ± 0.01	$\mathbf{19.47 \pm 0.01}$	13.23 ± 0.13	13.39 ± 0.01
	FREAK	15.5 ± 0.01	$\mathbf{19.92 \pm 0.01}$	13.75 ± 0.01	13.1 ± 0.01
	ORB	11.28 ± 0.01	$\mathbf{18.6 \pm 0.01}$	12.15 ± 0.01	13.03 ± 0.01

Table 4.2: Scenes and Mobile datasets: P@5 (%) or P@10 (%) for each descriptor with different mid-level representations.

to the results “BRIEF + Soft-MAX" and “ORB + Soft-MAX" can be considered the most suitable approaches for the Caltech 101 and VOC2007 datasets.

In [Pessoa et al., 2015a], we got similar results using euclidean distance (Figures 4.13 and 4.14), however, for the Pascal VOC 2007 dataset, the best ones are “ORB + Soft-MAX", “BRIEF + Soft-MAX", “BRIEF + Hard-AVG" and “BRIEF + Hard-AVG + Huffman" (this approach uses lossless compression - see section 2.3). BinBoost and BRISK are time consuming and have been discarded.

Lossy Compression: Here, we show the our results of lossy compression (Soft-MAX Truncated and Soft-MAX Truncated using Ranges – see section 2.3) reported in Pessoa et al. [2015a]. Figures 4.15 (caltech101) and 4.16 (VOC2007) present the relation between P@10 and Compression Ratio (CR) for the lossy compression of Soft-MAX representation in the Caltech 101 and Pascal VOC 2007 datasets, respectively.

In the Caltech 101 dataset, we have included the best Soft-MAX representation to compare its performance with the compact ones. In the Pascal VOC 2007 dataset, we also have included the “BRIEF + Hard-AVG" and “BRIEF + Hard-AVG + Huffman".

Dataset	Descriptor	Soft-AVG	Soft-MAX	Hard-MAX	Hard-AVG
caltech101 (P@10)	BRIEF	17.3 ± 0.02	$\mathbf{27.51 \pm 0.04}$	18.07 ± 0.02	20.96 ± 0.03
	BRISK	22.09 ± 0.03	$\mathbf{27.46 \pm 0.04}$	17.23 ± 0.02	21.29 ± 18.47
	FREAK	19.4 ± 0.03	$\mathbf{21.4 \pm 0.03}$	17.6 ± 0.02	18.28 ± 0.03
	ORB	16.38 ± 0.02	$\mathbf{26.51 \pm 0.04}$	17.12 ± 0.02	20.76 ± 0.03
caltech256 (P@10)	BRIEF	11.57 ± 0.01	$\mathbf{13.01 \pm 0.01}$	11.69 ± 0.01	12.34 ± 0.01
	BRISK	12.7 ± 0.01	$\mathbf{14.22 \pm 0.01}$	11.85 ± 0.01	12.59 ± 0.01
	FREAK	$\mathbf{12.53 \pm 0.01}$	12.78 ± 0.01	11.71 ± 0.01	12.4 ± 0.01
	ORB	11.5 ± 0.01	$\mathbf{13.99 \pm 0.01}$	11.59 ± 0.01	12.56 ± 0.01
WANG (P@10)	BRIEF	37.81 ± 0.07	$\mathbf{54.65 \pm 0.12}$	34.93 ± 0.08	34.93 ± 0.08
	BRISK	54.94 ± 0.11	$\mathbf{57.25 \pm 0.13}$	36.77 ± 0.08	52.78 ± 0.12
	FREAK	$\mathbf{44.18 \pm 0.12}$	38.52 ± 0.15	33.89 ± 0.14	42.37 ± 0.13
	ORB	41.61 ± 0.12	$\mathbf{56.12 \pm 0.13}$	36.72 ± 0.12	51.11 ± 0.13
UWDataset (P@10)	BRIEF	23.37 ± 0.04	$\mathbf{33.58 \pm 0.08}$	19.63 ± 0.06	29.35 ± 0.07
	BRISK	31.78 ± 0.07	$\mathbf{33.09 \pm 0.07}$	18.31 ± 0.04	31.37 ± 0.07
	FREAK	30.9 ± 0.07	30.14 ± 0.06	18.42 ± 0.06	$\mathbf{31.68 \pm 0.08}$
	ORB	23.93 ± 0.04	$\mathbf{35.05 \pm 0.08}$	18.85 ± 0.04	31.63 ± 0.06
VOC2007 (P@10)	BRIEF	18.31 ± 0.03	$\mathbf{18.81 \pm 0.03}$	17.91 ± 0.03	18.01 ± 0.04
	BRISK	19.65 ± 0.03	$\mathbf{20.52 \pm 0.04}$	17.67 ± 0.03	19.15 ± 0.04
	FREAK	17.25 ± 0.03	$\mathbf{20.2 \pm 0.03}$	17.31 ± 0.03	18.74 ± 0.04
	ORB	18.69 ± 0.03	$\mathbf{20.83 \pm 0.04}$	18.18 ± 0.03	19.38 ± 0.04

Table 4.3: Scenes and Mobile datasets: P@5 (%) or P@10 (%) for each descriptor with different mid-level representations.

In both datasets, “ORB + Soft-MAX" and “BRIEF + Soft-MAX" have better P@10 compared with the lossy compression of Soft-MAX representations.

It is worth to observe that Soft-MAX Truncated achieves similar P@10 compared with raw Soft-MAX, but it is more compact (transformation of interger to float values). For example, the precision (P@10) of “ORB + Soft-MAX" is 45,6471% ($\approx$ 45.65%) with CR = 21.41 and the “ORB Soft-MAX Truncated" is 45.6492% ($\approx$ 45.65%) with CR = 85.83. In this case, the highest CR values were observed with the “ORB Soft-MAX Truncated" approach, which are more compact (See Table 4.4).

	Caltech 101	Pascal VOC 2007
BRIEF + Soft-MAX	3.69	21.3
BRIEF + Soft-MAX Truncated	14.9	85
ORB + Soft-MAX	3.75	21.41
ORB + Soft-MAX Truncated	15.07	85.83

Table 4.4: Compression Ratio (CR) of “BRIEF + Soft-MAX", “BRIEF + Soft-MAX Truncated", “ORB + Soft-MAX" and “ORB + Soft-MAX Truncated" in the datasets Caltech 101 and Pascal VOC 2007.

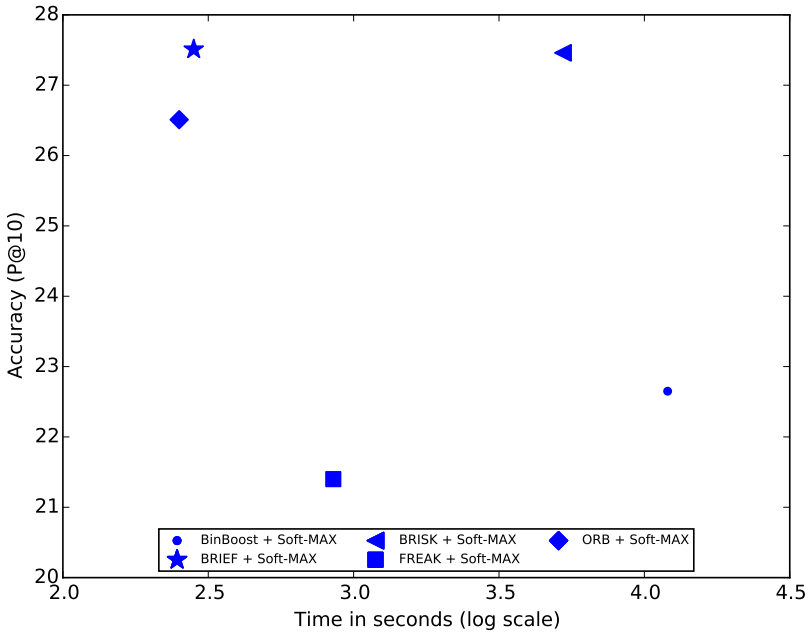


Figure 4.6: Manhattan distance (caltech101): Relationship of Accuracy (P@10) versus Time in seconds (log scale) for the most accurate descriptors (See Table 4.3).

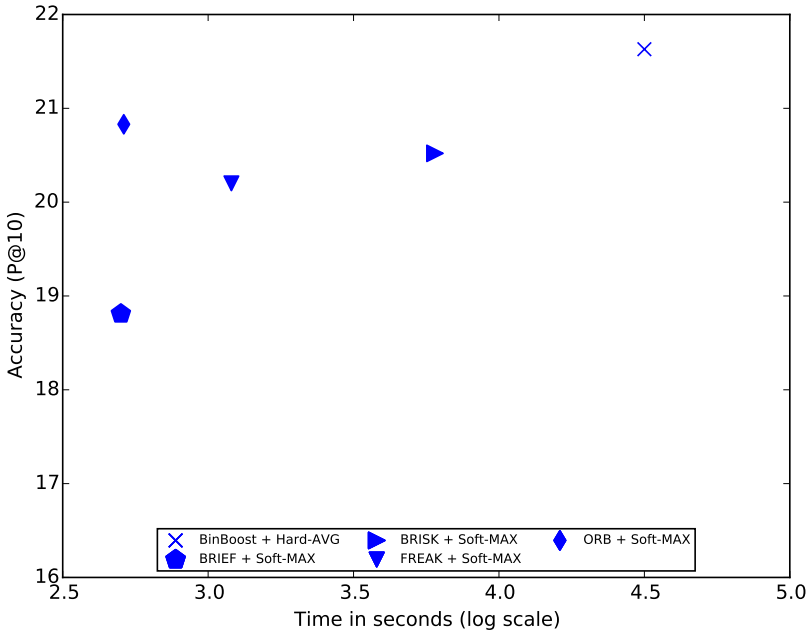


Figure 4.7: Manhattan distance (VOC2007): Relationship of Accuracy (P@10) versus Time in seconds (log scale) for the most accurate descriptors (See Table 4.3).

Even though the compression approaches that use ranges (lossy compression) are extremely compact, they produce low precision rates, which invalidates their use.

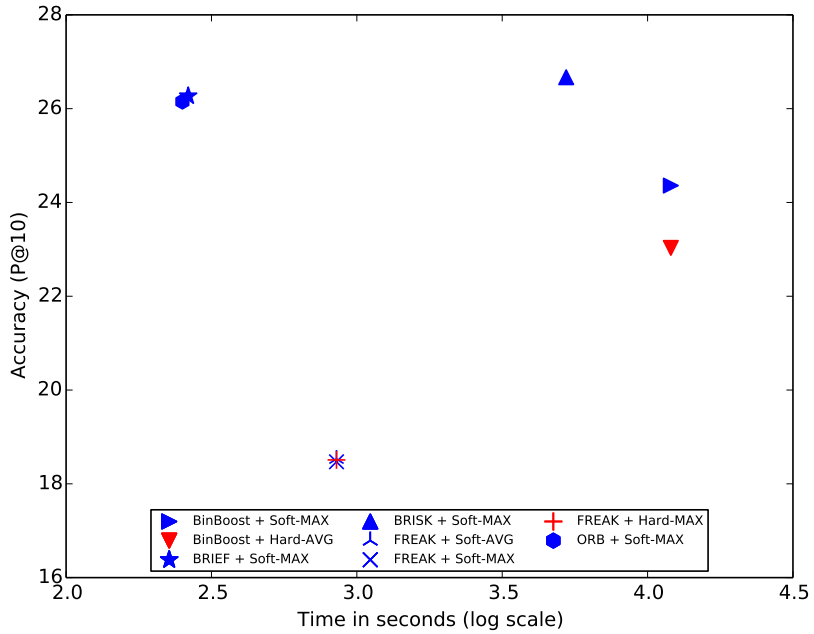


Figure 4.8: Euclidean distance (caltech101): Relationship of Accuracy (P@10) versus Time in seconds (log scale) for the most accurate descriptors (Results in Pessoa et al. [2015a]).

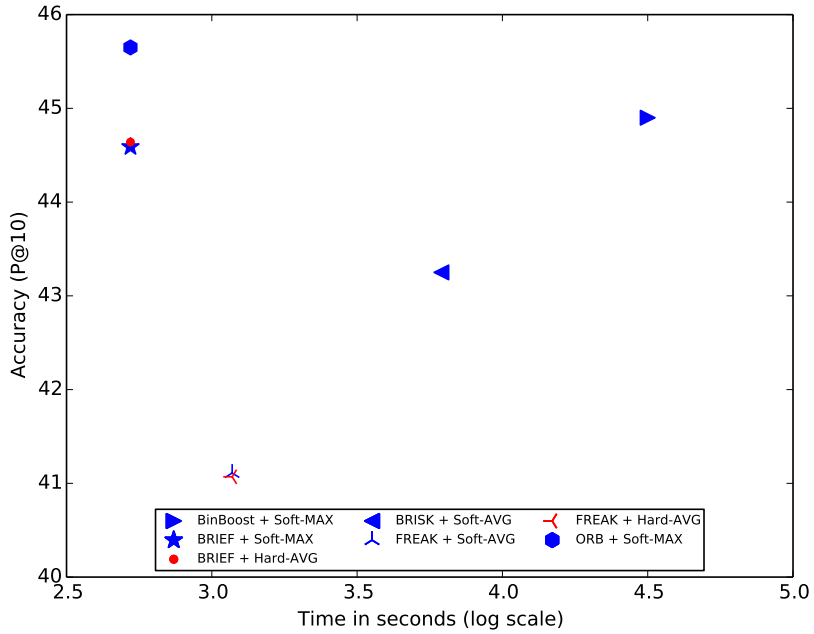


Figure 4.9: Euclidean distance (VOC2007): Relationship of Accuracy (P@10) versus Time in seconds (log scale) for the most accurate descriptors (Results in Pessoa et al. [2015a]).

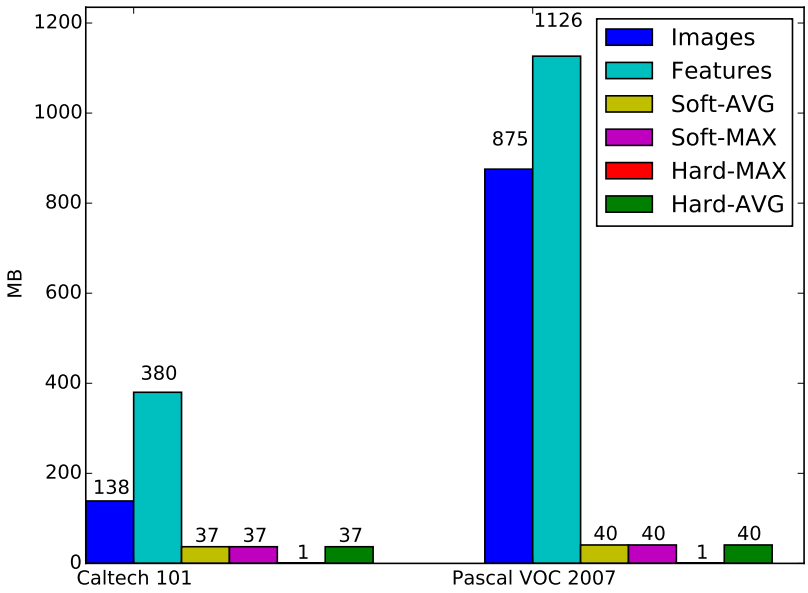


Figure 4.10: Size (MB) of the Images, Features and ORB' Mid-Level Representations in the Caltech 101 and Pascal VOC 2007 datasets.

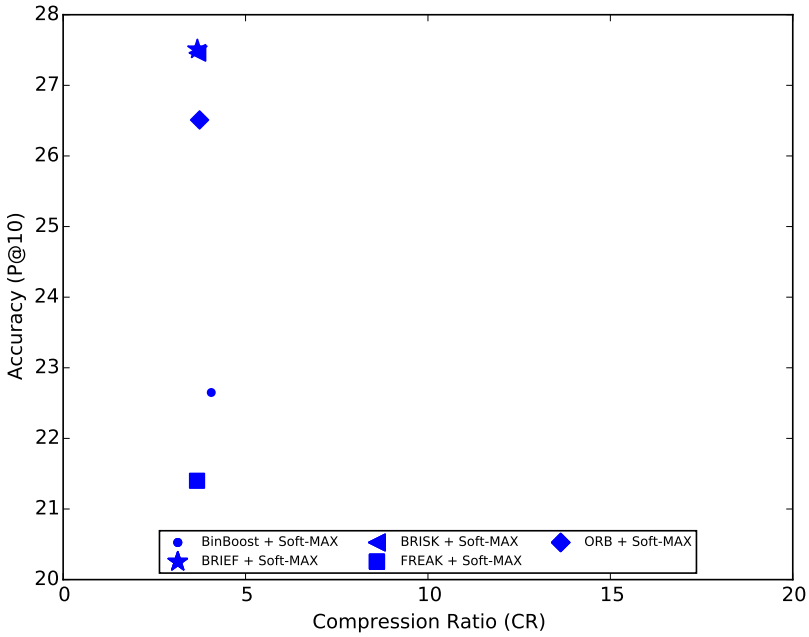


Figure 4.11: Manhattan distance (caltech101): Relationship between P@10 and Compression Ratio (CR) for the most suitable feature representations (See Table 4.3).

4.4 Low-level Global Representation Analysis

We aim at evaluating global low-level descriptors to find the most suitable configurations in terms of effectiveness, efficiency, and compactness. We use seventeen global

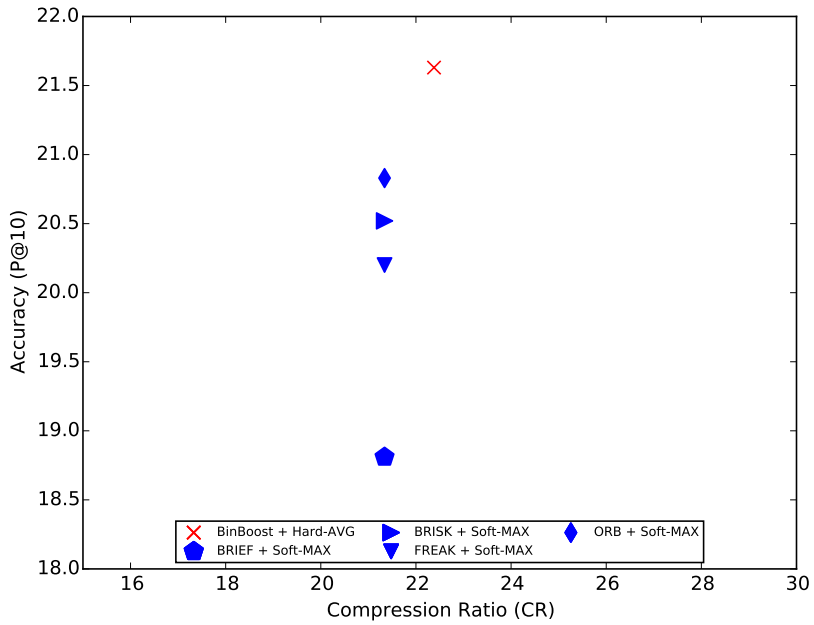


Figure 4.12: Manhattan distance (VOC2007): Relationship between P@10 and Compression Ratio (CR) for the most suitable feature representations (See Table 4.3).

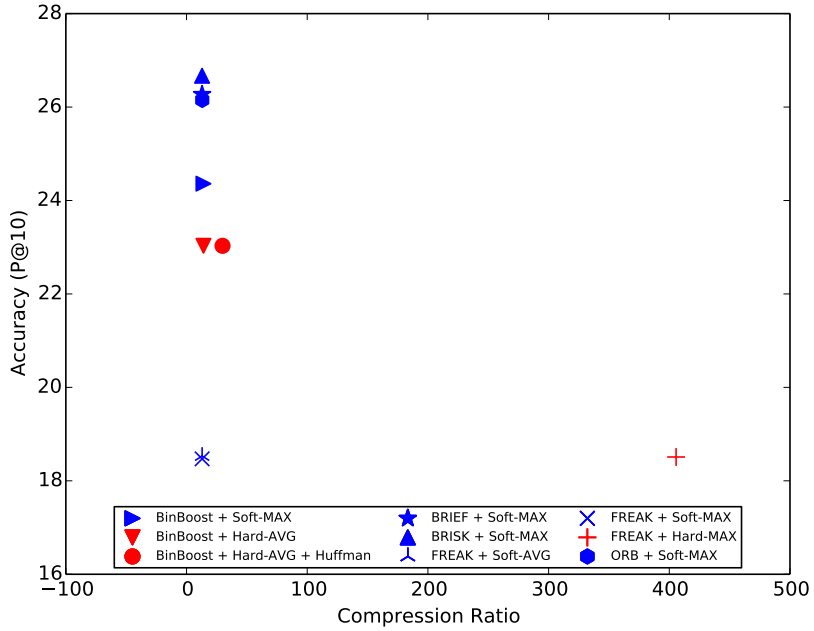


Figure 4.13: Euclidean distance (caltech101): Relationship between P@10 and Compression Ratio (CR) for the most suitable feature representations (Results in Pessoa et al. [2015a]).

descriptors in a content-based image retrieval (CBIR) application protocol. We have evaluated seventeen global descriptors (ten color descriptors, five texture descriptors and two shape descriptors). The ranking of the CBIR process is created by using the

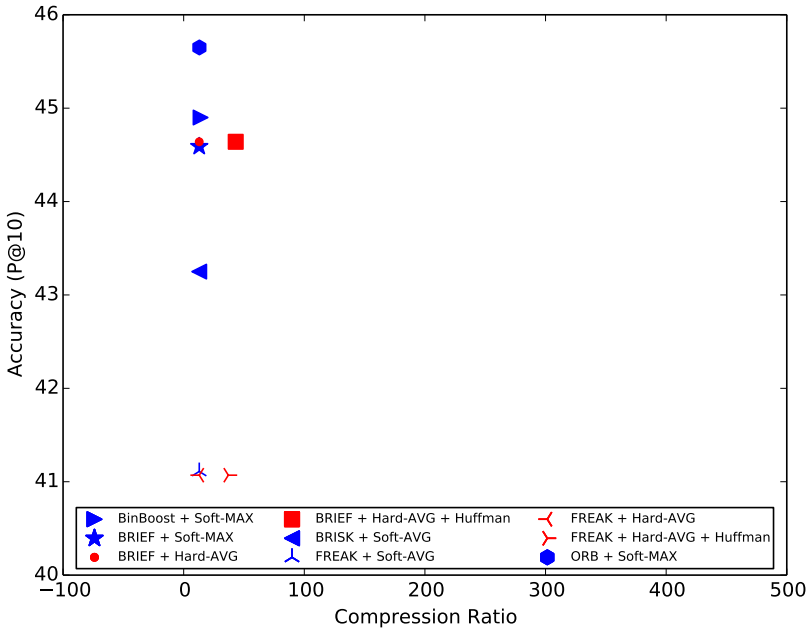


Figure 4.14: Euclidean distance (VOC2007): Relationship between P@10 and Compression Ratio (CR) for the most suitable feature representations (Results in Pessoa et al. [2015a]).

Manhattan Distance ($L1$ Distance).

4.4.1 Efficiency Evaluation

Figure 4.17 presents the computational time required for each global descriptor to extract features for all images in Caltech 101 (9,114 images) and Pascal VOC 2007 (9,963 images) combined. According to the results, SASI descriptor is the most expensive global descriptors and the others sixteen global descriptors can be considered good choices since they are very fast. For this experiment, we extracted each descriptor five times per image and the results are reported with a confidence of 95% (α=0.05). Complexity analysis of all global descriptors used in this thesis are found in Penatti and da Silva Torres [2008].

4.4.2 Effectiveness Evaluation

Table 4.5 shows the P@10 (or P@5) for the best six global descriptors analyzed using the results in Appendix C (MAP) and Appendix D (P@5/P@10). We use P@5 just to SVMS692 and ZuBuD. In five datasets, the BIC descriptor achieved the best results. Other satisfying results were obtained using LCH [Swain and Ballard, 1991], LAS [Tao and Dickinson, 2000] and SASI [Carkacioglu, 2001; Çarkacıoglu and Yarman-Vural,

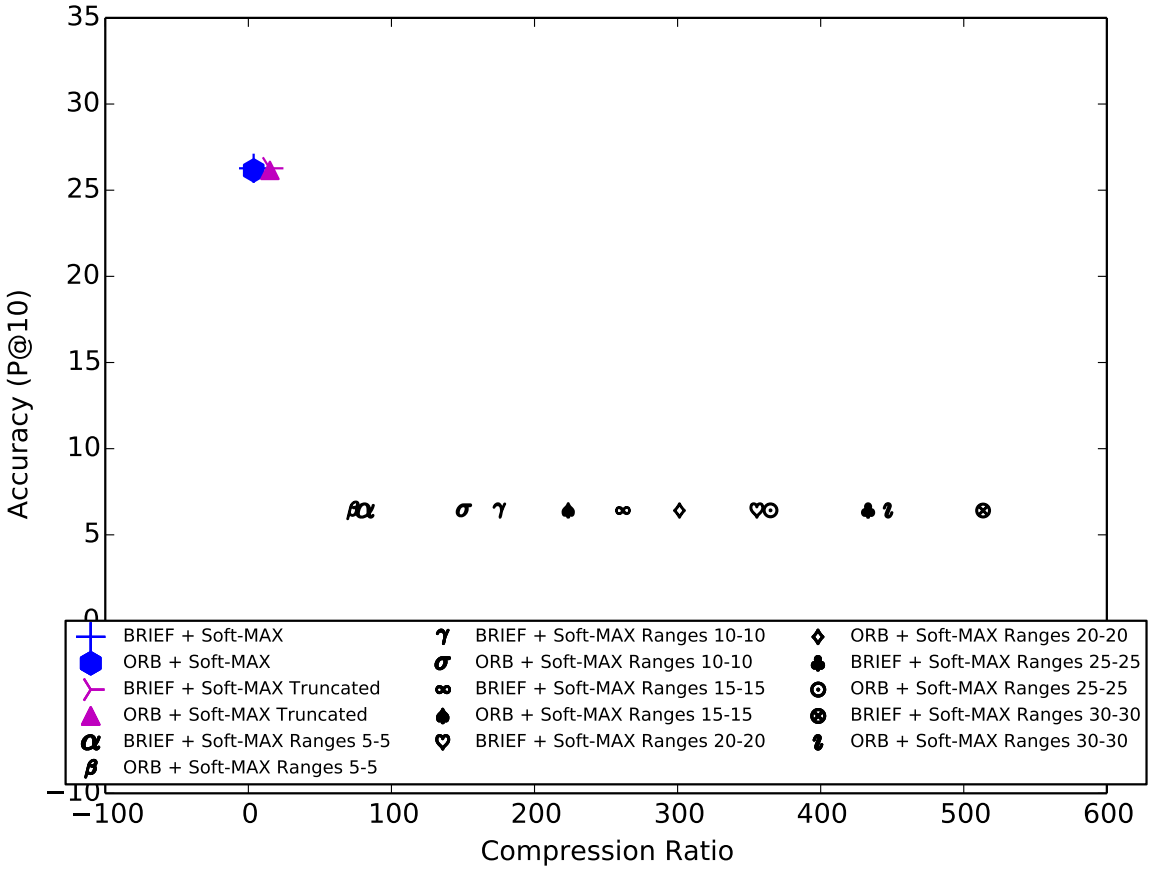


Figure 4.15: Manhattan distance (caltech101): Relationship between P@10 and Compression Ratio (CR) for the lossy compression of Soft-MAX representation. The best representations in Figure 4.13 have been included only for comparison.

2003]. A reason for color descriptors, such as BIC and LCH, performed well in our experiments is because several of the analyzed datasets have a lot of color information, which is a great advantage for the BIC and LCH descriptors. On the other hand, in datasets with grayscale images (for instance, 15Scenes dataset), texture or shape descriptors are the best options. The best descriptor in terms of effectiveness, efficiency, compactness is BIC (Border/Interior Pixel Classification) [Stehling et al., 2002].

To evaluate each global descriptor in both effectiveness and efficiency aspects, we present the scatter plots which show the relation between P@10 and time in seconds (in log scale) for the best global descriptors in the Caltech 101 (Figure 4.18) and VOC 2007 (Figure 4.19) datasets. In this scenario, the most suitable representations is BIC, LCH and LAS for both Caltech 101 and VOC 2007.

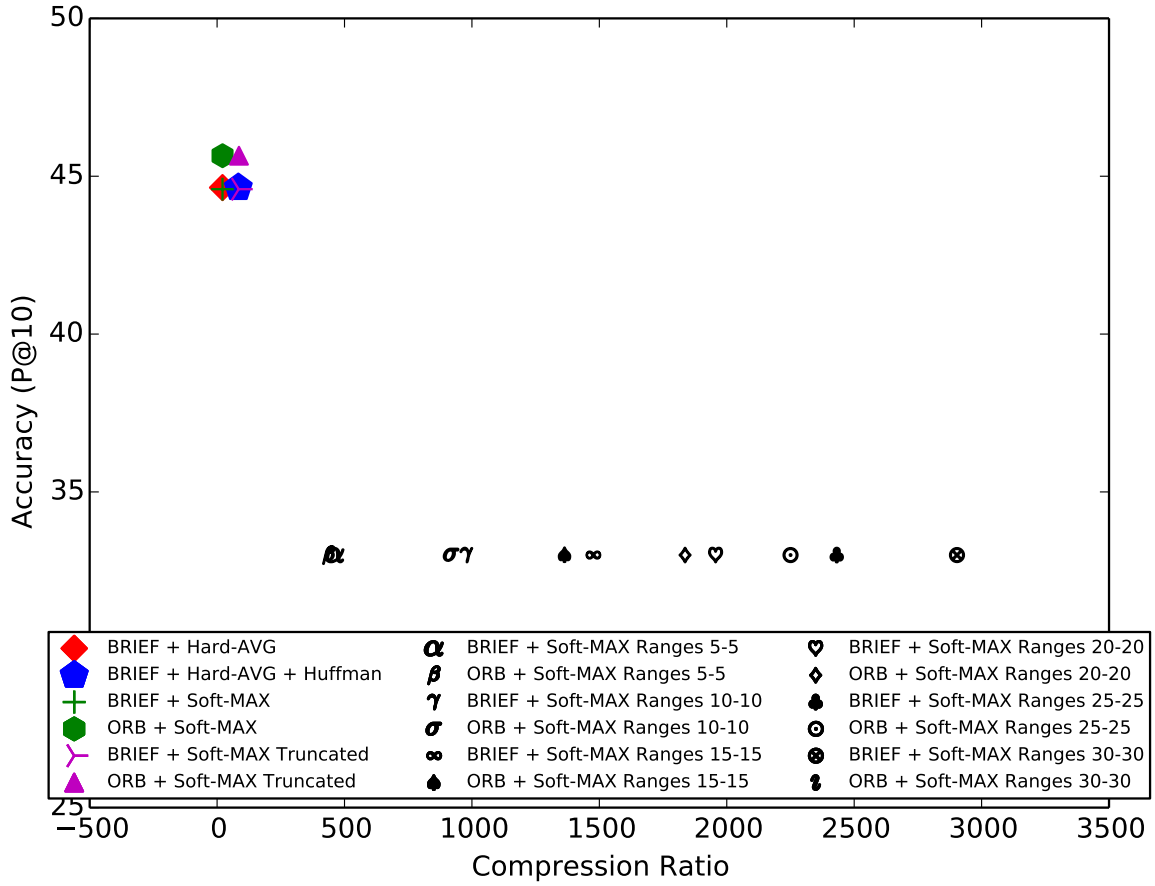


Figure 4.16: Manhattan distance (VOC2007): Relationship between P@10 and Compression Ratio (CR) for the lossy compression of Soft-MAX representation. The best representations in Figure 4.14 have been included only for comparison.

4.4.3 Compactness Evaluation

In this subsection, we evaluate the feature representation compactness aiming at finding the most suitable global descriptors concerning their feature vector size. As it can be seen in Figure 4.20, except JAC, it is better to transfer any global descriptor to be processed in the server side instead of images, binary low-level features or mid-level representations because global descriptors representation are more compact.

Figures 4.21 and 4.22 present the relation between P@10 and Compression Ratio (CR) for the most suitable feature representations in the Caltech 101 and Pascal VOC 2007 datasets, respectively. According to the results BIC, LCH and LAC can be considered the most suitable global descriptos for the Caltech 101 and VOC2007 datasets.

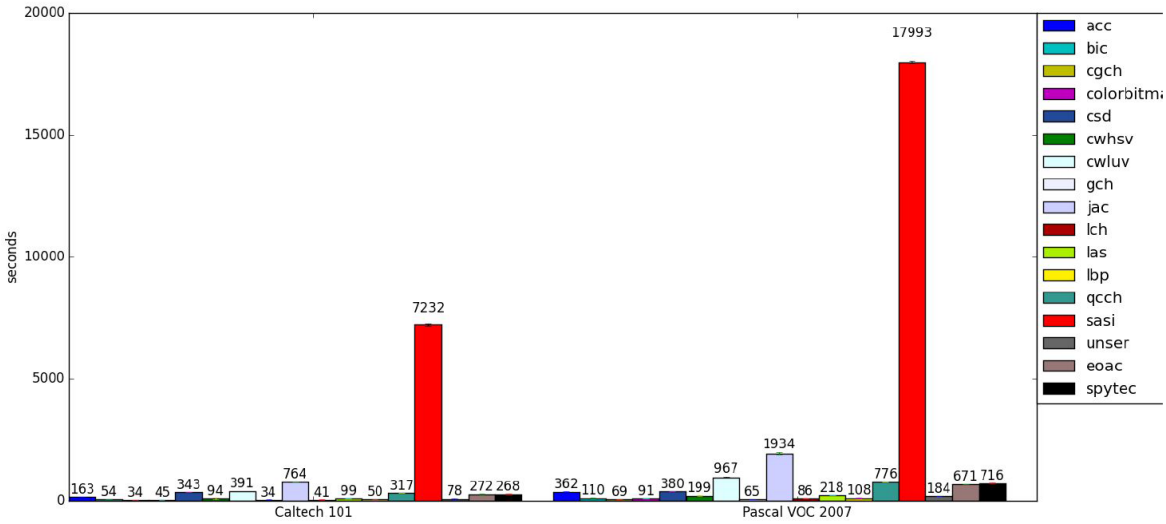


Figure 4.17: Time (in seconds) spent for global feature extraction of all images of Caltech 101 (9,144 images) and VOC 2007 (9,963 images) using different descriptors.

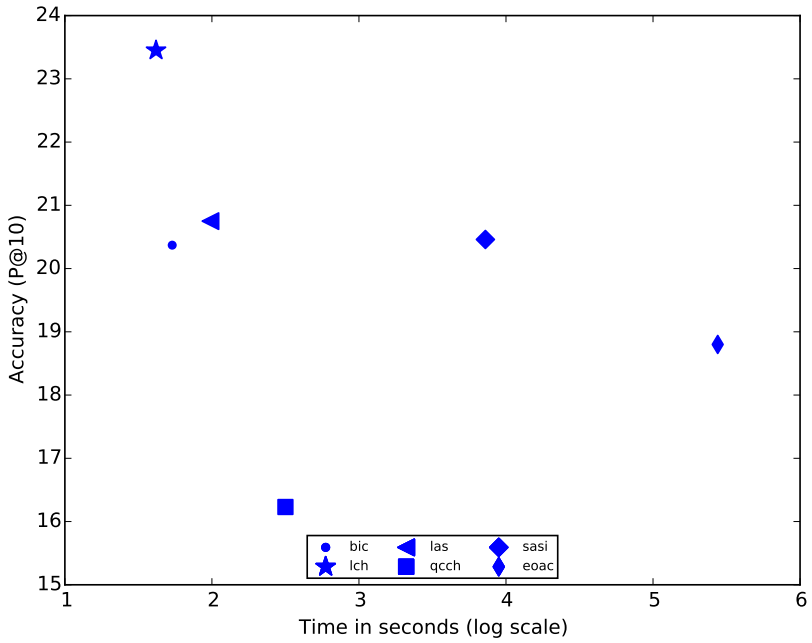


Figure 4.18: Manhattan distance (caltech101): Relationship of Accuracy (P@10) versus Time in seconds (log scale) for the most accurate global descriptors (See Table 4.5).

4.5 Discussion

Considering the analysis of effectiveness, efficiency and compactness of visual features presented, we can draw our conclusions about the best global descriptors and bag of words representations to be used in the mobile image search scenario: 1) BIC (Border/Interior Pixel Classification) [Stehling et al., 2002] and 2) DEOBSM (Bag of Words

Dataset	Descriptor		Descriptor	
15Scenes (P@10)	BIC	30.2 ± 0.05	QCCH	31.71 ± 0.06
	LCH	32.02 ± 0.06	SASI	$\mathbf{50.76 \pm 0.08}$
	LAS	49.32 ± 0.09	EOAC	42.23 ± 0.05
caltech101 (P@10)	BIC	20.37 ± 2.89	QCCH	16.23 ± 0.02
	LCH	$\mathbf{23.45 \pm 0.03}$	SASI	20.46 ± 0.03
	LAS	20.75 ± 0.03	EOAC	18.8 ± 0.03
caltech256 (P@10)	BIC	$\mathbf{15.31 \pm 0.01}$	QCCH	12 ± 0.01
	LCH	$\mathbf{15.3 \pm 0.01}$	SASI	13.92 ± 0.01
	LAS	13.81 ± 0.01	EOAC	13.06 ± 0.01
OxBuild11 (P@10)	BIC	28.43 ± 0.1	QCCH	25.46 ± 0.08
	LCH	32.19 ± 0.12	SASI	34.78 ± 0.11
	LAS	$\mathbf{36.18 \pm 0.13}$	EOAC	26.93 ± 0.1
Paris (P@10)	BIC	$\mathbf{32.79 \pm 0.07}$	QCCH	22.61 ± 0.03
	LCH	28.98 ± 0.05	SASI	28.46 ± 0.04
	LAS	30.81 ± 0.05	EOAC	26.44 ± 0.04
SMVS692 (P@5)	BIC	23.08 ± 0.01	QCCH	21.53 ± 0.01
	LCH	$\mathbf{23.46 \pm 0.01}$	SASI	21.6 ± 0.01
	LAS	23.17 ± 0.01	EOAC	21.32 ± 0.01
UWDataset (P@10)	BIC	$\mathbf{59.74 \pm 0.1}$	QCCH	26.48 ± 0.05
	LCH	42.65 ± 0.09	SASI	44.12 ± 0.08
	LAS	29.54 ± 0.06	EOAC	40.56 ± 0.07
VOC2007 (P@10)	BIC	21.05 ± 0.03	QCCH	16.87 ± 0.03
	LCH	19.4 ± 0.04	SASI	21.3 ± 0.04
	LAS	$\mathbf{22.16 \pm 0.04}$	EOAC	19.72 ± 0.03
WANG (P@10)	BIC	$\mathbf{77.73 \pm 0.1}$	QCCH	48.97 ± 0.14
	LCH	64.13 ± 0.13	SASI	64.62 ± 0.12
	LAS	59.84 ± 0.14	EOAC	60.44 ± 0.14
ZuBuD (P@5)	BIC	$\mathbf{72.62 \pm 0.02}$	QCCH	42.37 ± 0.02
	LCH	70.87 ± 0.03	SASI	47.82 ± 0.02
	LAS	51.14 ± 0.02	EOAC	42.47 ± 0.02

Table 4.5: P@5 (%) or P@10 (%) for the six best global descriptors.

using Dense Sampling, ORB descriptor, Soft assignment and Maximum pooling), respectively. We note that the BIC descriptor seems to outperform DEOBSM in almost cases.

Paired statistical tests (Table 4.6) show in which scenary BIC can be considered better then DEOBSM. In the table, P@N means "Precision on the top N images" and "Not different" mean that BIC and DEOBSM have no statistical difference with 95% confidence. We use P@5 just to SVMS692 and ZuBuD. Therefore, for mobile visual

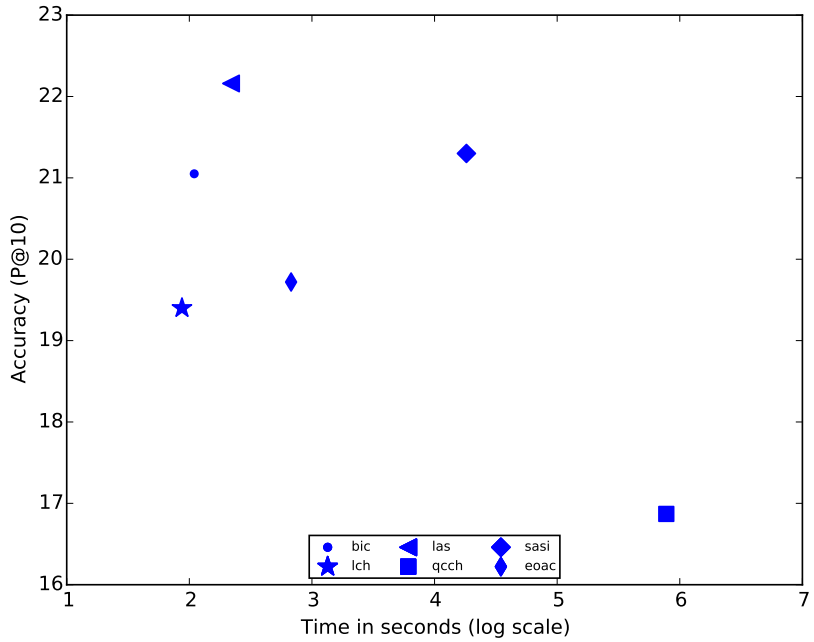


Figure 4.19: Manhattan distance (VOC2007): Relationship of Accuracy (P@10) versus Time in seconds (log scale) for the most accurate global descriptors (See Table 4.5).

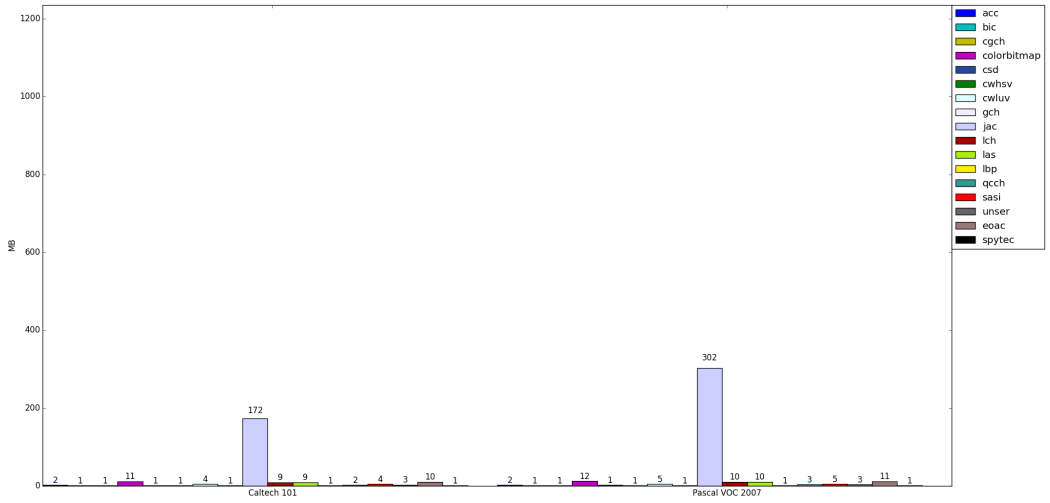


Figure 4.20: Size (MB) of the Global Representations in the datasets Caltech 101 (9,144 images) and Pascal VOC 2007 (9,963 images).

retrieval, we may consider using BIC descriptor as the best option considering the triple trade-off problem regarding efficiency, effectiveness and compactness.

Figures 4.23, 4.24, 4.25 and 4.26 show precision versus recall curves for 15Scenes (Scenes), SMVS692 (Mobile), UWDataset (Single-label) and WANG (Multi-label)

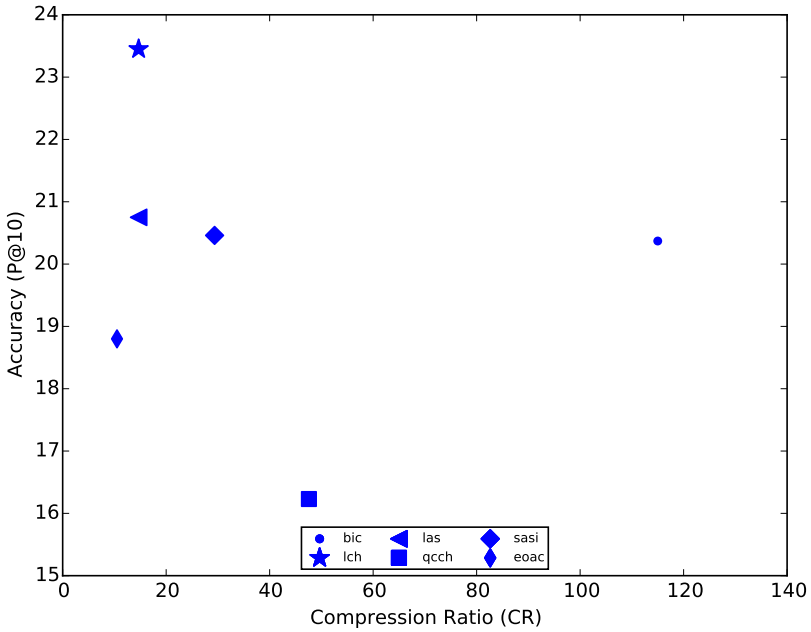


Figure 4.21: Manhattan distance (caltech101): Relationship between P@10 and Compression Ratio (CR) for the most suitable feature representations in Table 4.5.

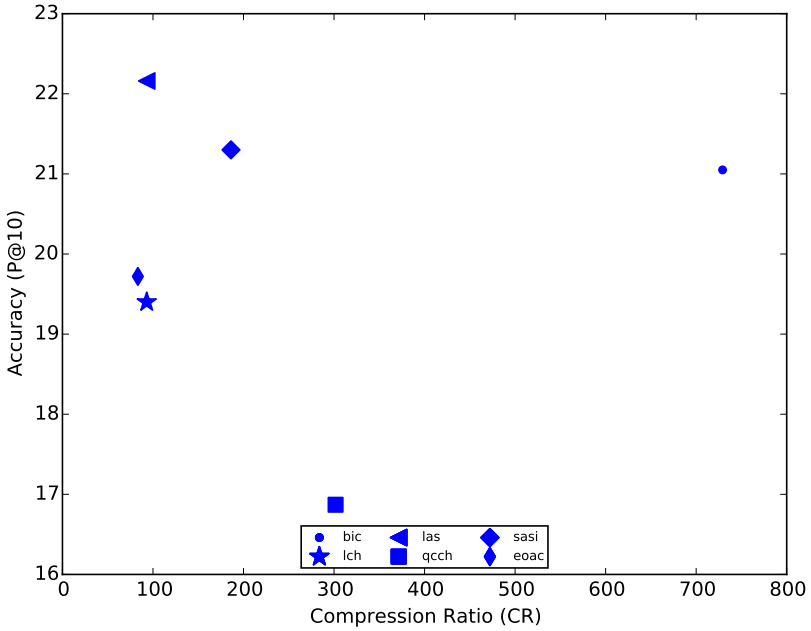


Figure 4.22: Manhattan distance (VOC2007): Relationship between P@10 and Compression Ratio (CR) for the most suitable feature representations in Table 4.5.

datasets, respectively. We show only the twelve best descriptors found out in our image retrieval precision analysis (detailed in Appendix C (MAP) and Appendix D (P@5/P@10)).

	Dataset	BIC	DEOBSM	Best, IC(95%)
MAP	15Scenes (P@10)	12.99 ± 0.02	$\mathbf{21.76 \pm 0.05}$	DEOBSM
	caltech101 (P@10)	6.68 ± 0.02	$\mathbf{10.62 \pm 0.03}$	DEOBSM
	caltech256 (P@10)	2.86 ± 0.01	$\mathbf{5.32 \pm 0.01}$	DEOBSM
	OxBuild11 (P@10)	20.57 ± 0.06	$\mathbf{33.53 \pm 0.06}$	DEOBSM
	Paris (P@10)	$\mathbf{11.65 \pm 0.03}$	11.38 ± 0.03	BIC
	SMVS692 (P@5)	24.14 ± 0.01	$\mathbf{35.98 \pm 0.01}$	DEOBSM
	UWdataset (P@10)	$\mathbf{36.24 \pm 0.08}$	18.89 ± 0.05	BIC
	VOC2007 (P@10)	$\mathbf{25.37 \pm 0.04}$	24.31 ± 0.04	BIC
	WANG (P@10)	$\mathbf{51.8 \pm 0.12}$	37.28 ± 0.14	BIC
	ZuBuD (P@10)	$\mathbf{78.99 \pm 0.02}$	71.15 ± 0.03	BIC
P@N	15Scenes (P@10)	30.2 ± 0.05	$\mathbf{43.88 \pm 0.08}$	DEOBSM
	caltech101 (P@10)	20.37 ± 0.03	$\mathbf{26.51 \pm 0.04}$	DEOBSM
	caltech256 (P@10)	$\mathbf{15.31 \pm 0.01}$	13.99 ± 0.01	BIC
	OxBuild11 (P@10)	28.43 ± 0.1	$\mathbf{46.29 \pm 0.13}$	DEOBSM
	Paris (P@10)	32.79 ± 0.07	32.90 ± 0.05	Not different (P@10)
	SVVM692 (P@10)	23.08 ± 0.01	$\mathbf{34.27 \pm 0.01}$	DEOBSM
	UWdataset (P@10)	$\mathbf{59.74 \pm 0.1}$	35.05 ± 0.08	BIC
	VOC2007 (P@10)	$\mathbf{21.05 \pm 0.03}$	20.83 ± 0.04	BIC
	WANG (P@10)	$\mathbf{77.73 \pm 0.1}$	56.12 ± 0.13	BIC
	ZuBuD (P@10)	$\mathbf{72.62 \pm 0.02}$	0.7001 ± 0.03	BIC

Table 4.6: Statistical test paired t-test, with 95% of confidence, between the BIC descriptor and the DEOBSM descriptor.

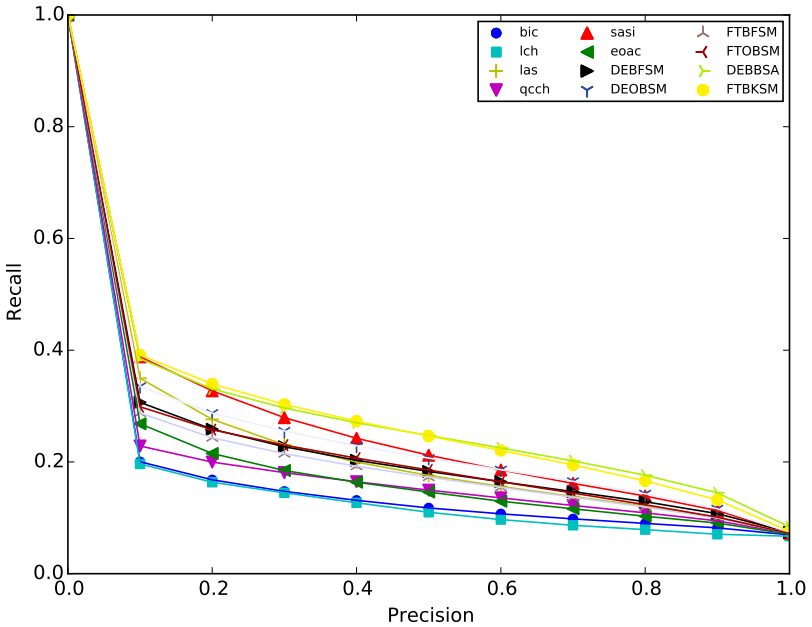


Figure 4.23: Precision × Recall of the best descriptors in 15Scenes dataset.

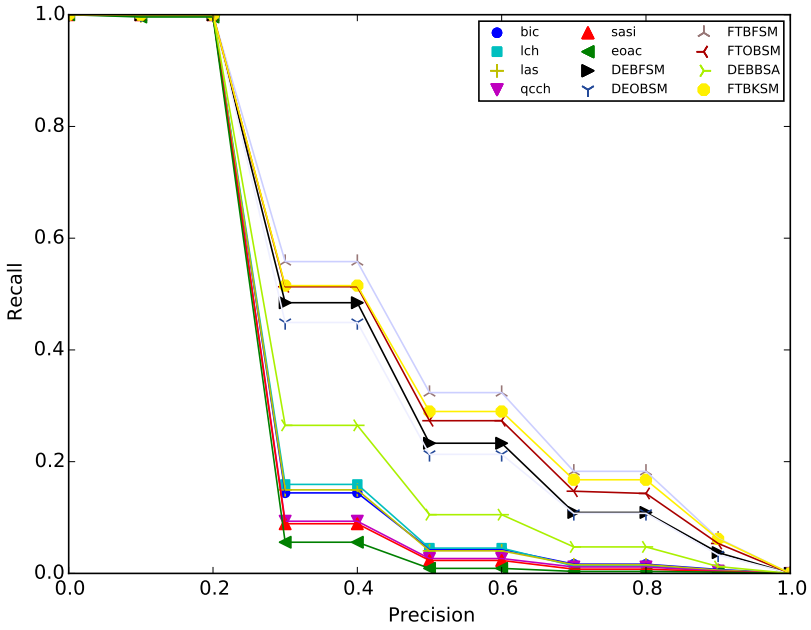


Figure 4.24: Precision × Recall of the best descriptors in SMVS692 dataset.

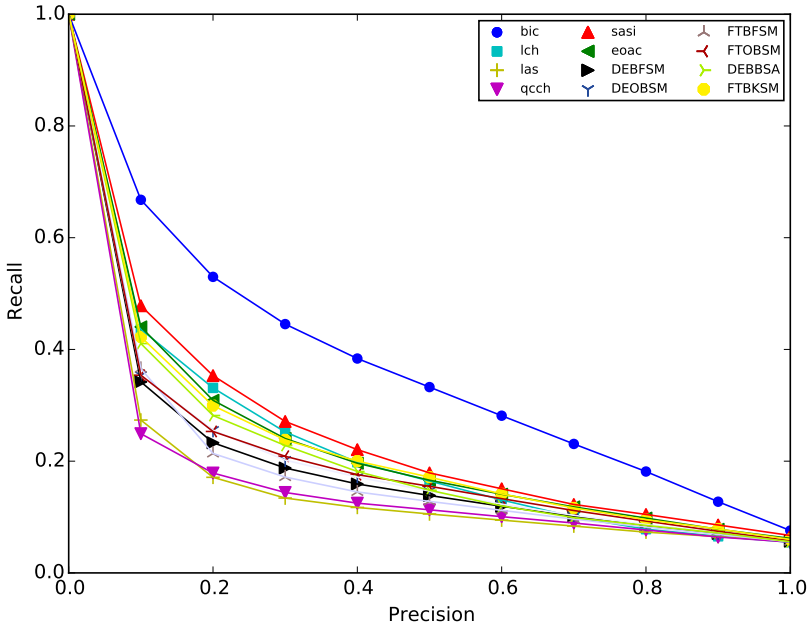


Figure 4.25: Precision × Recall of the best descriptors in UW dataset.

Image retrieval experiments on mobile devices We developed a prototype system using Android platform[4] for retrieving the content of general image by using mobile devices as query interface. We did some preliminary tests in the smartphone LG Nexus 5 using the best descriptors pointed out in this thesis (BIC and Bag of Words using

[4]Acessed on November 2015: Android platform - `http://developer.android.com/`

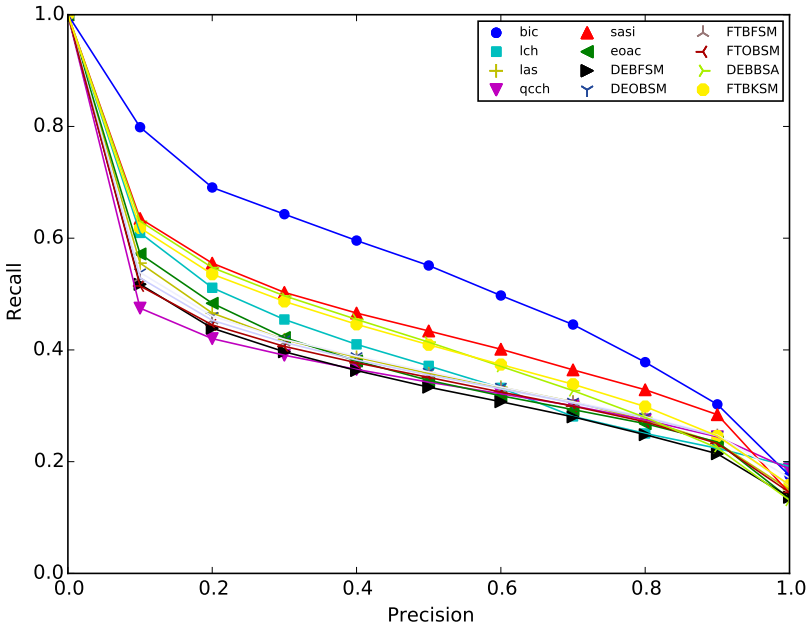


Figure 4.26: Precision × Recall of the best descriptors in WANG dataset.

Dense Sampling, ORB descriptor, Soft assignment and Maximum pooling). As a result, for images dimension of 300 × 500 using *LG Nexus 5*, the feature extraction of BIC takes about 300 milliseconds and DEOBSM (Denso + ORB (with bag of words of size 1024) + Soft-MAX) takes around 500 milliseconds. Using bag of words with size of 128 and the feature extraction takes about 300 milliseconds. It's hard to be less than 300 milliseconds because of the JNDI (*Java Naming and Directory Interface*) overhead.

General discussion In this chapter, we conducted a series of experiments to evaluate aspects of effectiveness, efficiency and compactness of extracted features of images to perform content-based image retrieval on mobile devices. In this case, the user decides the best triple trade-off configuration regarding effectiveness, efficiency, and compactness of visual features using more or less resources on the mobile devices. The experiments indicate that the descriptors BIC (Border/Interior Pixel Classification – a color descriptor), LAS (Local Activity Spectrum – a texture descriptor), DEOBSM (Bag of Words using Dense Sampling, ORB descriptor, Soft assignment and Maximum pooling) and FTBKSM (Bag of Words using FAST Algorithm Sampling, BRISK descriptor, Soft assignment and Maximum pooling) are the best options. BIC descriptor was point out as the best option considering the triple trade-off problem regarding efficiency, effectiveness and compactness.

Chapter 5

Spatial Feature Representation for Mobile Image Search

This chapter refers to the problem of extracting spatial information from images to improve the quality of image representation on mobile devices. We proposed two new bag-of-visual words-based approaches to encode spatial information. Section 5.1 gives an overview of the challenges in represent visual content by considering the spatial configuration of visual words of part of images in the image space. Section 5.2 describe differences between existing approaches and the two proposed methods in this thesis. Then, we present two proposed spatial pooling methods based on fixed parts and segmented parts in Sections 5.3.1 and 5.3.2, respectively. Section 5.4 presents the experiments for image retrieval on mobile devices using four different spatial bag of words. Finally, we present some conclusions on section 5.5.

5.1 Introdution

In a very recent past, bag-of-visual words representations are successfully used in many applications of computer vision such as image retrieval and classification. However, the traditional pooling methods usually discard the spatial configuration for visual words in the image and this kind of information is important to distinguish types of object and arrangements in the image. Therefore, the research community has been very active in order to propose new approaches of bag of visual words which encodes the spatial information of visual words to improve image semantics and distinguish different classes of scenes or objects [Penatti et al., 2014].

To encode spatial information, some of the existing approaches compute bag of words in different parts and levels of the image as the tradicional Spatial Pyramid

Matching (SPM) [Lazebnik et al., 2006] or just use geometrical information encoded by the traditional bag-of-words representation [Cao et al., 2010]. Some other methods include spatial information in the assignment step as Penatti et al. [2011a] or on the pooling step as Avila et al. [2013]. Finally, some works compute a bag of words representation and then trying to find the matching of visual words between images computing the spatial information and leaving the statial verification as a post-processing [Jégou et al., 2010a; Zhou et al., 2010a].

A contribution of this thesis is to address the problem of extracting spatial information from parts of images to improve the quality of image representation on mobile devices which encodes a spatial information of bag of visual parts of images. In Chapter 4, we point out the BIC (Border/Interior Pixel Classification) as one of the best descriptors analyzed. Thus, we use this descriptor to create representations of part of images and the vectors representation are used on mid-level strategies.

We have proposed two approaches: (1) BOBGrid and (2) BOBSlic. BOBGrid encodes a graph of BIC representations on images. The graph encodes relationships of nine fixed quadrants on the image and use them on a bag of words protocol. Similarly, BOBSlic use the superpixel algorithm SLIC (Simple Linear Iterative Clustering) to segment more homogenious regions on the images. After that, BOBSlic encodes BIC representations on parts of the images and create bag of words representations. We have conducted experiments by comparing the two proposed spatial representations against Word spatial arrangement (WSA) Penatti et al. [2011a] and BossaNova Avila et al. [2013].

5.2 Related Work

One of the most famous works on encoding spatial information of visual words is the Spatial Pyramid Matching (SPM) [Lazebnik et al., 2006]. SPM splits the image into fixed-size tiles and generates one BoW representation for each tile. For a pyramid level of 2, for example, 21 bags are generated. The first bag is the original bag of words representations without splitting. In the next level, the image is split into 4 tiles of the same size. The next level splits each of the 4 tiles into another set of 4 tiles. Therefore, there is 1 bag for level 0, 4 bags for level 1 and 16 bags for level 2. All the bags are concatenated to create the image feature vector. The main problem of this strategy is related to the large feature vector size.

Cao et al. [2010] present an approach to spatial pooling of visual words based on creating linear and circular projections of the image. The linear projections consider

the horizontal axis as reference. The image is split into vertical tiles and a BoW representation is generated for each tile. The axis is then rotated by an angle of θ and each of the tiles generates another set of bags. This is performed by a predefined number of angles. The circular projections consider a set of points to be the center of the image and then splits the image into sectors. A BoW representation is computed for each sector. The final feature vector is a concatenation of all bags generated by linear and circular projections. The method reorder bags in the feature vector to achieve rotation, translation, and scale invariance. Its main disadvantage is the large feature vector size.

Aiming at encode the spatial relationship among every pair of points in the image, Zhou et al. [2010a] use binary spatial maps. The spatial verification is a post-processing step in the retrieval framework, applied only for matching visual words between query and database images. A horizontal spatial map is an $N \times N$ binary matrix, where each row i says if the feature i is at right (1) or at left (0) of each other feature. The vertical spatial map is analogous, having value 1, in row i, for points which i is above and value 0, otherwise. The effect of the spatial maps calculations consists in splitting the image into 4 quadrants, using each point in the image as the origin. The method also considers rotation and scale issues.

Word spatial arrangement (WSA), proposed by Penatti et al. [2011a], is an approach to represent the spatial arrangement of visual words under the bag-of-visual-words model and it is based on the idea of dividing the image space into quadrants using each point as the origin of the quadrants and counting the number of words that appear in each quadrant. It lies in a simple idea which encodes the relative position of visual words by splitting the image space into quadrants using each detected point as origin. WSA generates compact feature vectors and is flexible for being used for image retrieval and classification, for working with hard or soft assignment, requiring no pre/post processing for spatial verification. WSA counts how many times a visual word w_i appears in each quadrant in relation to all other points in a specific image. This counting will tell us the spatial arrangement of the visual word w_i. Intuitively, the counting will measure the positioning of a word in relation to the other points in the image. It reveals, for example, that a word w_i tends to be below, at right, or surrounded by other points. By counting w_i position in relation to the other points in the images, without considering the labels of other points (visual words assigned to them), WSA generates a not-too-precise representation, which is interesting for the semantic-search application. Figure 5.2 presents the whole procedure of the WSA approach.

Althout is not a spatial mid-level representation, the BossaNova representation [Avila et al., 2013] is an extension of BoW model which provides an improvement

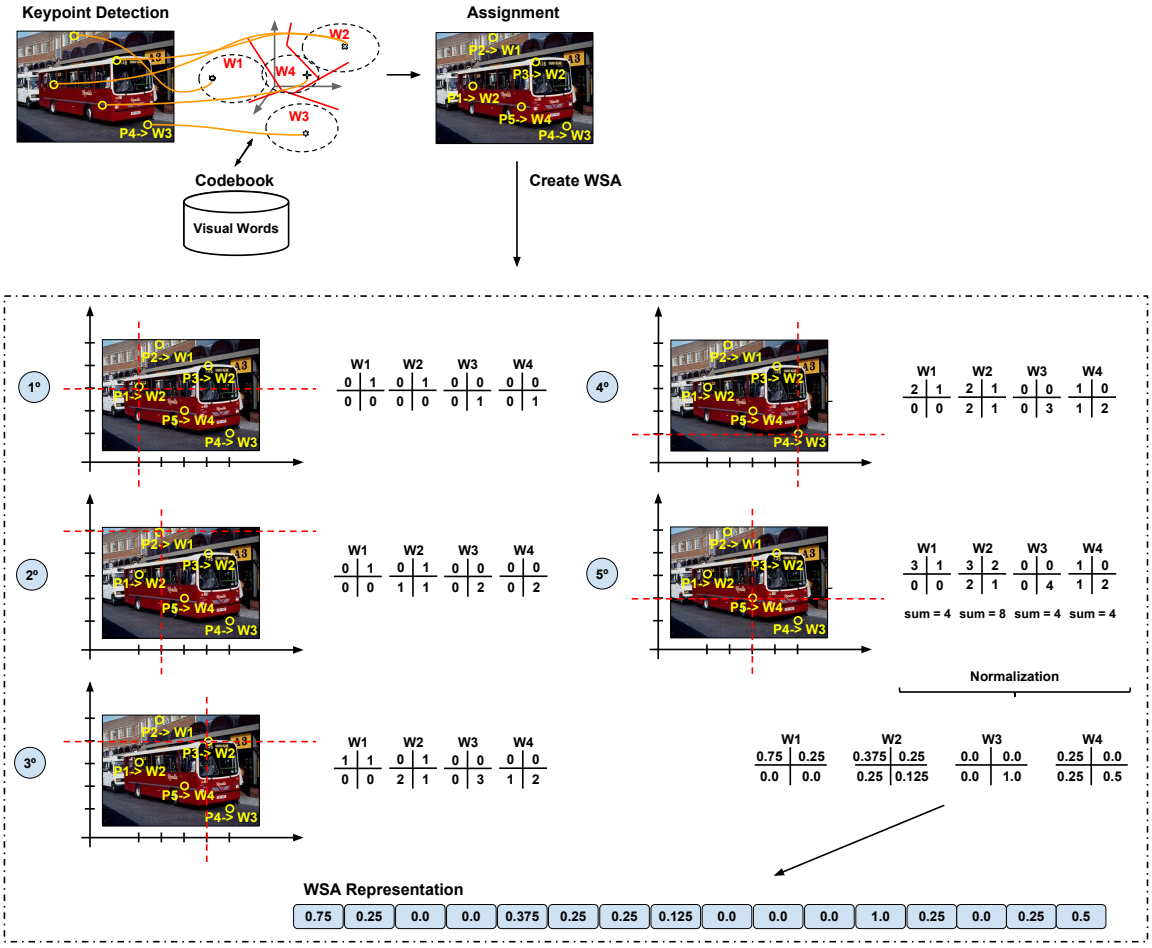


Figure 5.1: Word spatial arrangement (WSA) Ilustration: First, the image space is divided and each point p_i is detected in the image by a sparse sample, for example. Then, the space is divided into 4 quadrants, putting the point p_i in the quadrant's origin. For every other detected point p_j , WSA increments the counters of the visual word associated with p_j in the position that corresponds to the position of p_j in relation to p_i. For instance, if w_j is the visual word associated with p_j and p_j is at top-left from p_i, the counter for top-left position of w_j is incremented. After all points are analyzed in relation to p_i, the quadrant's origin goes to the next point p_i+1, and the counting in relation to p_i+1 begins. When all points have been the quadrant's origin, the counting finishes and each 4-tuple is normalized by its sum [Penatti et al., 2011a].

in the pooling stage, to preserve a more rich way the information obtained during the encoding step. BossaNova differentiates from the BoW approach at the coding/pooling stage, resulting in a new representation that better preserves the information from the encoded local descriptors by using a density-based pooling step. Their coding function activates the closest codewords to the descriptor, which corresponds to a localized soft coding over the visual codebook. The pooling step estimates the distribution of the descriptors around each codeword, while the BoW estimates the distribution around

one or determined number of codewords. The BossaNova pooling process is a density-based estimation of the descriptors distribution generating a histogram of distances between the descriptors found in the image and each codeword. In the encoding step of the BossaNova, Avila et al. [2013] use a strategy of soft coding, considering the k-nearest visual words location for encoding of a local descriptor. The pooling step can be modeled by function g, that estimates the probability density function. The function g represents a density distribution of the distances (B bins) between the codeword c_m and the local descriptors of an image. BossaNova representation consists of three main parameters: 1) the distance range ($[\alpha_m^{min}, \alpha_m^{max}]$), 2) the local histogram number of bins (B) and 3) the size of the visual dictionary (K). BossaNova is illustrated in Figure 5.2.

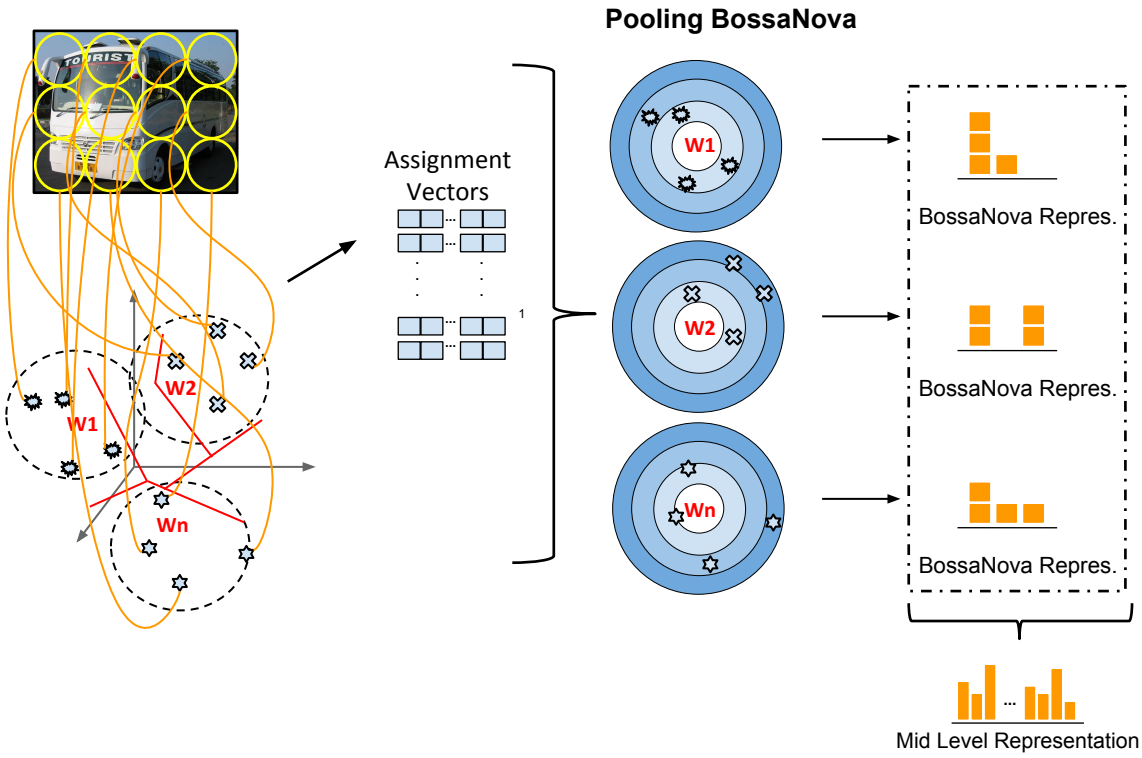


Figure 5.2: BossaNova Ilustration: After the assignment step all the assignment vectors are summarized by a pooling step. For each center c_m, BossaNova obtains a local histogram z_m. The differents blue colors indicate the discretized distances from the center c_m to the local descriptors shown by the symbols in the circles. For each colored bin $z_{m,b}$, the height of the histogram is equal to the number of local descriptors x_j, whose discretized distance to the visual words c_m fall into the b^{th} bin [Avila et al., 2013].

5.3 Proposed Approaches

In the section, we detail the two proposed spatial mid-level approaches: BOBGrid and BOBSlic.

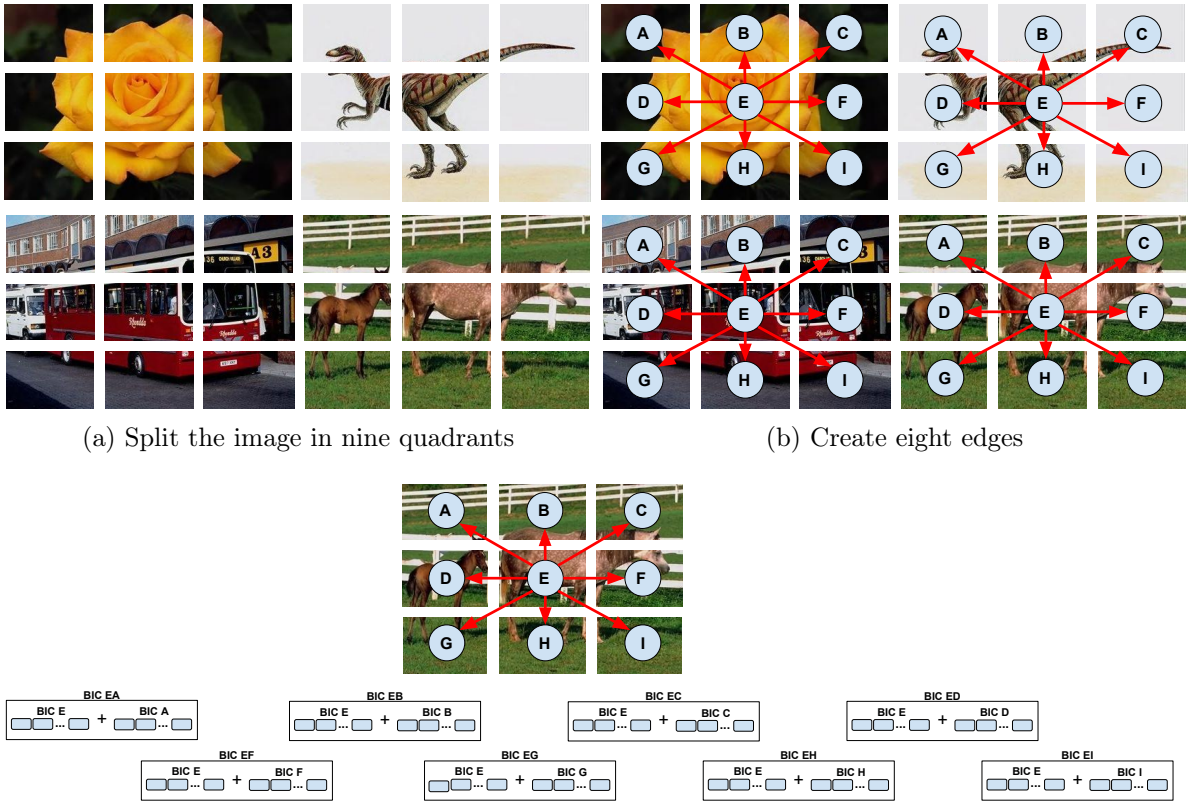
5.3.1 BOBGrid Representation

To encode spatial information on bag-of-visual words representation, we propose to split the image in nine similar quadrants. For each tile, we compute visual features by using the BIC descriptor [Stehling et al., 2002] – Border/Interior Pixel Classification, which were the most suitable descriptor as presented in Chapter 4. We encode the spatial information by creating a graph with edges starting from the the center quadrant. In summary, we use a directed graph with eight edges. The BOBGrid (Bag Of BIC Grid) is divided into two steps: Offline and Online. In the offline step, we used the nine splited parts of all images on the dataset to create a dictionary (or codebook) using k-means algorithm. On the Online step, we use the dicionary created to generate BOBGrid representations. A dictionary of 128 visual words was constructed selecting points in the feature space to create BOBGrid representations.

Figure 5.3 presents the process of splitting the image in nine quadrants and the creation of edges generating a graph in the imagem. Figure 5.4 presents the offline and online processes to create a codebook based on nine quadrants in the image and the process which uses the dictionary created to generate BOBGrid representations.

5.3.2 BOBSlic Representation

The second spatial algorithm proposed in this work is the BOBSlic (Bag Of SLIC the BIC) representation. BOBSlic use the SLIC algorithm [Achanta et al., 2012] – Simple Linear Iterative Clustering – to segment the image in parts segmented. As shown in section 2.4.1, SLIC has two parameters: n = Number of superpixels and their compactness (c). We tested several values for the parameters n (n = 10, 20 or 30) and c (c = 0, 5, 20, 50). The best results were obtained with n = 10 and c = 50. Therefore, we use this setting to segment images with SLIC. After SLIC, we compute BIC descriptors [Stehling et al., 2002] for each segmented part. We have created a **complete graph** where the edge is a concatenation of two BIC vectors. The graph is undirected, therefore we have two edges round trip (edge = $[BIC_a + BIC_b]$ and edge = $[BIC_b + BIC_a]$). Again, we have an offline and online steps where a codebook (dictionary) is created and used to create spatial bag of words from images on the dataset and the query image.



(a) Split the image in nine quadrants

(b) Create eight edges

(c) BIC Representations of Edges

Figure 5.3: BIC in 9 Quadrants Graph Flow: Process of split the image in nine quadrants (a) and the creation of edges generating a graph in the imagem (b). In (c), we illustrate the concatenation order of the vectors generated for each quadrant of the image which represent an edge.

Figure 5.5 presents the offline and online processes to create a codebook based on SLIC-BIC in the image and the process which uses the dictionary created to generate BOBSlic representations.

5.4 Experiments

To evaluate the proposed approaches considering a CBIR scenario, we have used the WANG dataset. Details of the datasets are shown in subsection 3.3.1. The dataset can be classified as different types of images with scenes (like monuments), object (like buses) and high intra-class variation like (africa). For each category, the same object appears in different rotation and viewpoints. In the experiments, we try to discern if the accuracy of BOBGrid and BOBSlic are comparable to the best methods from literature.

In our experimental setup, we compare BOBGrid and BOBSlic with our baseline

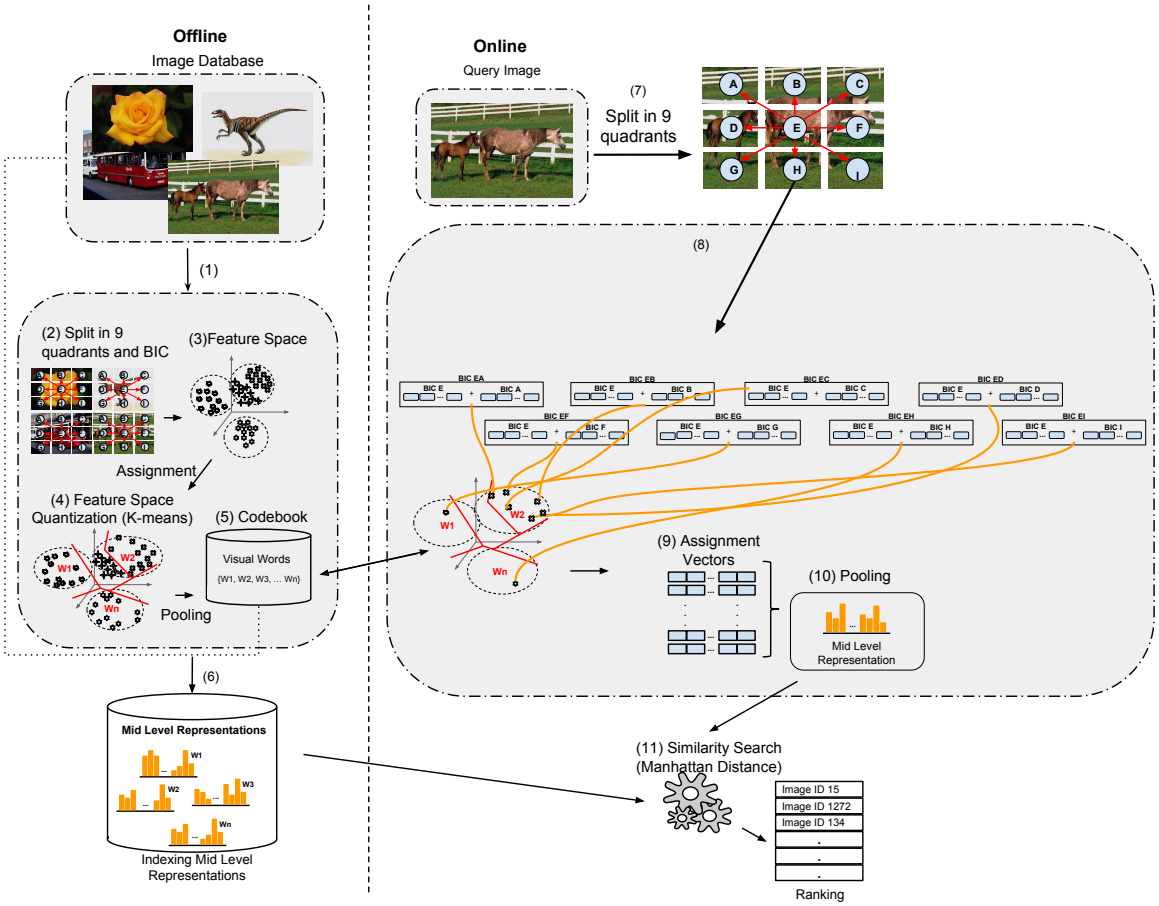


Figure 5.4: BoBGrid Representation: In the offline step, eight edges of nine quadrants are clustered and a codebook is created. On the online step, given a query image a BOBGrid representation is created and then assign to the dictionary to create a bag of words representation.

WSA. We also compare with the BossaNova approach. We use the best configuration described on the baselines papers of WSA [Penatti et al., 2011a] and BossaNova [Avila et al., 2013]. We use the best binary descriptors (ORB) founded out on the experimental analysis presented in Chapter 4. We select interest points for WSA by using sparse sampling as Penatti et al. [2011a] and use dense sampling for BossaNova suggested in Avila et al. [2013]. We use two sampling strategies to select interest points to WSA and BossaNova: 1) Good Features to Track using Harris (GFTTHarris) [Mikolajczyk et al., 2005; Tomasi and Shi, 1994] and 2) ORB detector [Rublee et al., 2011].

WSA configuration Penatti et al. [2011a] point out the best configuration of WSA used as WSA using soft assignment and maximum pooling (Soft-MAX), where the Harris-Laplace detector were used. In the work, we use the WSA using the Soft-MAX and Hard-MAX bag of words approaches. The selection of points were made using ORB

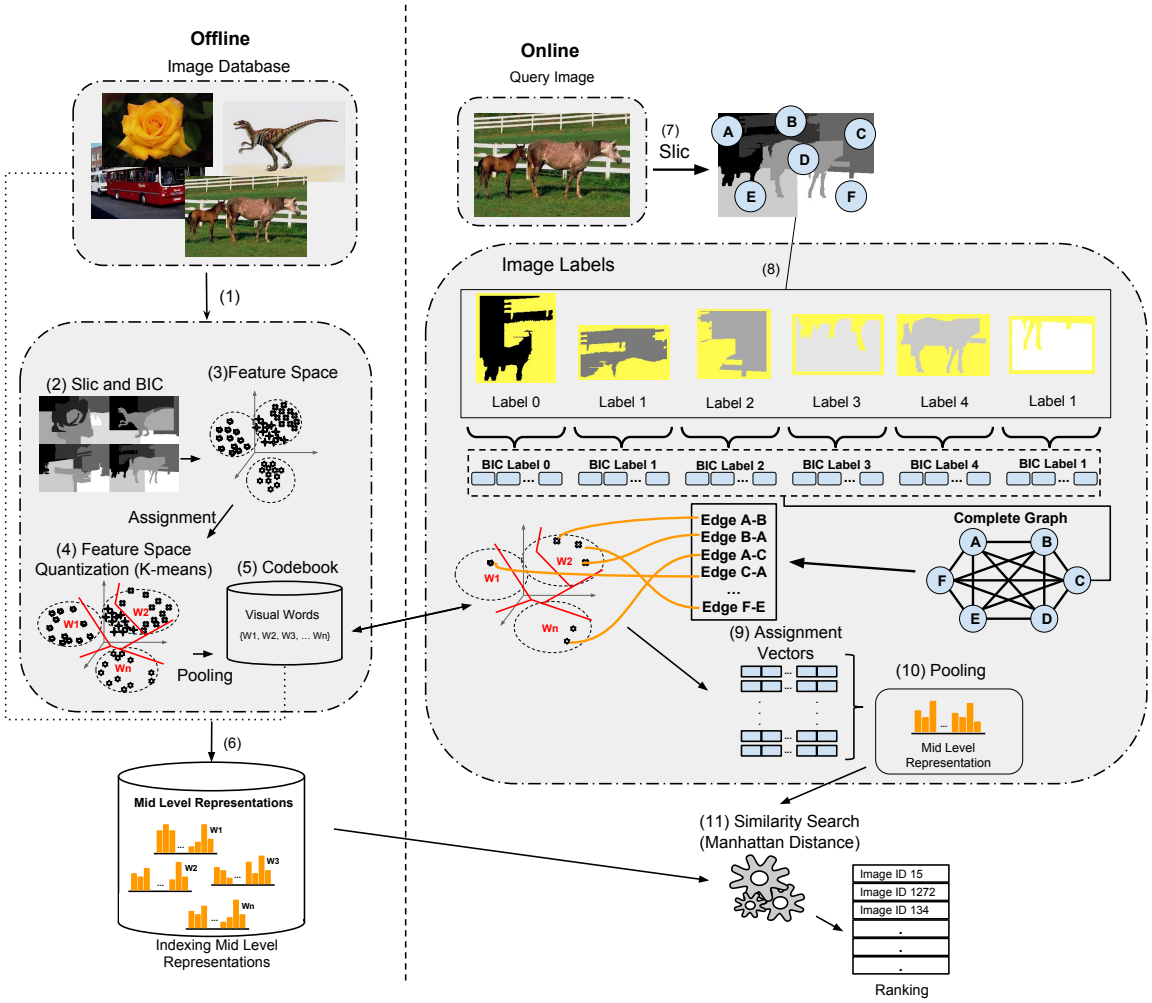


Figure 5.5: BOBSlic Representation: In the offline step, the segmented regions created by the superpixel algorithm SLIC are clustered and a codebook is created. On the online step, given a query image a BOBSlic representation is created and then assign to the dictionary to create a bag of words representation.

detector and GFTTHarris (to details of these sparse sampling, see: Section 2.2.1) and ORB descriptor was used to extract local feature vectors from images. A dictionary of 1024 visual words (as in [Caetano et al., 2014a]) were constructed by selecting points in the feature space [Viitaniemi and Laaksonen, 2008] and they were used in the WANG dataset. We used the σ soft parameter with the value 60 ($\sigma = 60$). For WSA, we have used the standard version (WSA) available on WSA info page [1].

BossaNova configuration In order to study the best binary descriptor behavior found (ORB descriptor) together with the BossaNova representation, we used the best configuration pointed out by Avila et al. [2013]. We extracted features from patches in

[1] Acessed on November 2015: `http://www.recod.ic.unicamp.br/~otavio/pr_wsa/index.htm`

a dense sampling every 6 pixels, as Avila et al. [2013] and use default parameters used Avila et al. [2013]: the distance range $[\alpha_m^{min}, \alpha_m^{max}]$ was defined as $\alpha_m^{min} = \lambda_{min} \times \sigma_m$ and $\alpha_m^{max} = \lambda_{max} \times \sigma_m$, where σ_m is the standard deviation of each cluster c_m obtained by clustering algorithm $k - medians$ [Caetano et al., 2014a]. Thus, the other parameters were defined as: 1) $[\alpha_m^{min} = 0.4, \alpha_m^{max}] = 2.0$, 2) the local histogram number of bins ($B = 2$) and 3) the size of the visual dictionary ($K = 1024$). A dictionary of 1024 visual words was constructed by randomly selecting points in the feature space [Viitaniemi and Laaksonen, 2008] and they were used in the WANG dataset. We used the Soft-MAX approach with $\sigma = 60$. For BossaNova, we have used the version (BossaNova) available on BossaNova info page [2] and modified in the work Caetano et al. [2014a] to be used with binary descriptors.

Results Table 5.1 shows the experimental results in WANG database [Wang et al., 2001]. Note that the proposed approaches (BOBGrid and BOBSlic) outperformed our baseline WSA, and also BossaNova and the tradicional Bag-of-Words models (BoW).

	Approach	**P@5(%)**	**P@10(%)**	**P@15(%)**	**MAP(%)**
Traditional BoW's	DEBBSA	70.92	64.37	60.35	41.29
	DEBBHA	73.32	65.71	61.35	37.93
	DEBFSM	62.94	54.65	50.55	34.65
	DEOBSM	63.52	56.12	52.51	37.28
	FTBKSM	70.74	63.47	59.65	41.63
Spatial BoW's	DEOBBN	66.02	58.67	54.99	38.93
	HSOBBN	55.02	46.95	43.11	29.33
	OBOBBN	48.18	40.31	37.17	27.43
	HSOBHMWSA	27.22	18.60	15.59	11.96
	HSOBSMWSA	32.04	23.16	20.01	13.67
	OBOBHMWSA	27.08	18.45	15.53	11.90
	OBOBSMWSA	29.40	20.54	17.58	17.58
Proposed	BOBGrid	**77.90**	**71.57**	**67.73**	**48.06**
	BOBSlic	**78.20**	**71.43**	**68.07**	**48.81**

Table 5.1: Precision results using P@5, P@10, P@15, MAP of BOBGrid and BOBSlic, our baseline (WSA), BossaNova approaches and traditional Bag of Words (BoW's) strategies on WANG dataset [Wang et al., 2001]. DE = Dense, FT = FAST, HS = GFTT using Harris, OB = ORB detector / ORB descriptor, BB = BinBoost, BF = BRIEF, BK = BRISK, SA = Soft-AVG, SM = Soft-MAX, HA = Hard-AVG, HM = Hard-MAX, BN = BossaNova (BossaNova), BOBGrid = Bag of BIC Grid, BOBSlic = Bag of Slic BIC.

[2] Acessed on November 2015: `https://sites.google.com/site/bossanovasite/`

It is important to point out that the proposed algorithms provide more compact representations than our baseline WSA and BossaNova, which are suitable for applications in mobile devices. Both vectors of BOBGrid and BOBSlic have length of 128, while WSA and BossaNova use vectors of length 1024. For retrieval experiments, which are generally based on computing distances between vectors, with the Euclidean or Manhattan distance, for example, vectors should be compact, or embedded in an index structure, to avoid the curse of the dimensionality [Traina Jr et al., 2002; Hoàng et al., 2010; Kang et al., 2011; Weber et al., 1998]. Table 5.2 summarizes all methods proposed/analyzed in this work and their feature vector sizes.

Method	**Feature Vector Size**
Traditional BoW's	K
Word Spatial Arrangement (WSA)	4K
BossaNova	$K \times (B + 1)$
BOBGrid	128
BOBSlic	128

Table 5.2: Feature vector sizes for all methods being evaluated in the experiments for image retrieval. K is the dictionary size. B is a BossaNova parameter which indicates the local histogram number of bins (Default: B=2)

To determine the statistical significance of results, a statistical test for differences between means was done using paired t-test, paired about the classes of the database. In Table 5.3, we present statistical test results between the BOBGrid descriptor and other representations results. In Table 5.4, we present statistical test results between the BOBSlic descriptor and other representations. The confidence interval (CI) for the mean difference is calculated using the Student t-test model and the difference is considered significant if the interval does not include zero (marked with $\surd$). For all tests, we used a confidence of 95%. In conclusion, BOBGrid and BOBSlic performs statistically better than our baseline WSA with a confidence of 95% on WANG dataset. BOBGrid and BOBSlic also showed greater accuracy in relation to BossaNova on WANG dataset.

5.5 Discussion

In this chapter, we introduced two approaches called BOBGrid (Spatial Bag of BIC Grid) and BOBSlic (Spatial Bag of Slic BIC) to encode spatial information with a compact Bag-of-Visual Words representation. We compare them against WSA (Visual Word Spatial Arrangement) [Penatti et al., 2014] and BossaNova (Bag Of Statistical

Method	P@10	Significant	Confidence Interval (CI) = 95%
1) BOBGrid	**71.57**		CI = [60.35; 82.79]
2) DEOBBN	58.67	-	CI = [45.50; 71.84]
3) HSOBBN	46.95	√	CI = [33.88; 58.02]
4) OBOBBN	40.31	√	CI = [25.50; 55.12]
5) HSOBHMWSA	18.60	√	CI = [17.92; 19.28]
6) HSOBSMWSA	23.16	√	CI = [19.41; 26.91]
7) OBOBHMWSA	18.45	√	CI = [17.71; 19.19]
8) OBOBSMWSA	20.54	√	CI = [18.60; 22.48]

Table 5.3: Statistical test t-test, with 95% of confidence, between the BOBGrid approach and other descriptors. The difference is considered significant if it is marked with √.

Method	P@10	Significant	Confidence Interval (CI) = 95%
1) BOBSlic (n=10, c=50)	**71.64**		CI = [60.14; 83.14]
2) DEOBBN	58.67	-	CI = [45.50; 71.84]
3) HSOBBN	46.95	√	CI = [33.88; 58.02]
4) OBOBBN	40.31	√	CI = [25.50; 55.12]
5) HSOBHMWSA	18.60	√	CI = [17.92; 19.28]
6) HSOBSMWSA	23.16	√	CI = [19.41; 26.91]
7) OBOBHMWSA	18.45	√	CI = [17.71; 19.19]
8) OBOBSMWSA	20.54	√	CI = [18.60; 22.48]

Table 5.4: Statistical test t-test, with 95% of confidence, between the BOBSlic approach and other descriptors. The difference is considered significant if it is marked with √. n = Number of superpixels. c = Compactness.

Sampling Analysis) [Avila et al., 2013]. According to statistical analyzes, both BOBGrid and BOBSlic outperform our baseline of spatial bag of visual words WSA and the BossaNova algorithm in the WANG dataset. In addition, descriptors are more compact, which make them more suitable for mobile devices applications. They only have vectors of length 128, while the baselines have size 1024. For retrieval experiments, which are based on computing distances between vectors, vectors should be compact to avoid the curse of the dimensionality.

Experiments on WANG dataset that has images of scenes, objects in general and high intra-class variation were important to study spatial information enconding for image retrieval talks. The experiments were evaluated using several precision metrics (P@5, P@10, P@15 and MAP) and statistical analysis. The results presented indicate the importance of using image parts and segmentations on images to create more robust bag of words. We could observe that BOBSlic seems to be better than BOBGrid because it uses a segmentation approach to represent bag of words. We seek to use the default settings provided in the baselines.

Chapter 6

Conclusion and Future Work

Content-based image retrieval (CBIR) is a well-known field of research in information management in which a large number of methods have been proposed and investigated but in which still no satisfying general solutions exist. Nowadays, with the huge number of web-connected smart mobile devices, the amount of data available on the web tends to increase more and more. Due to the improvement of mobile devices and wireless network techonologies, applications based on visual information, such as visual localization and navigation, image based online shopping, image categorization and image retrieval, have been more and more popular. Mobile Visual Search (MVS) is concerned with the indexing and retrieval of information such as text, graphics, animation, sound, speech, image, video, and their possible combinations for using in mobile devices with wireless network connectivity. The challenges faced by mobile visual search can be divided into: 1) caused by the nature of the visual search problem itself and 2) caused by the mobile device limitations. MVS shares the same ideas with other information retrieval technologies: they all start with querying the information of interest and comparing the queried information with known information in a database. Some works in the literature have been modeling CBIR systems, which use mobile device as query interface with server-client architectures. The main drawback of these systems is the need for techniques able to transfer query image/videos from the user device to the server. In this work we use a client-server architecture where the process of low-cost image feature extraction run on mobile and the image retrieval process run on the server side.

In this thesis, we addressed two research issues in order to develop effective solutions for image retrieval on mobile devices. First, a comparative study of binary descriptors using mid-level representation and global descriptors (color, texture, and shape) in image retrieval context on mobile devices, as well as, image features compres-

sion techniques. We also analyze the impact of dense sampling and sparse sampling to compute descriptors using bags of words strategies. The second research issue refers to the problem of extracting spatial information on images to improve the quality of image representation on mobile devices, which could be crucial to distinguish types of objects and scenes. The traditional pooling methods usually discard the spatial configuration for visual words in the image. We used a spatial bag of visual words strategy and a bag of words with improvements on the pooling step as baselines: WSA (Visual Word Spatial Arrangement) and BOSSA (Bag Of Statistical Sampling Analysis), respectively. We also proposed two approaches called BOBGrid (Spatial Bag of BIC Grid) and BOBSlic (Spatial Bag of Slic BIC).

In this work, we deal with the *feature extraction triple trade-off problem* in mobile devices by evaluating low-cost feature representations. We concentrate our efforts in four main fronts: (1) binary low-level descriptor selection; (2) mid-level representation; (3) low-level global representation analysis and (4) feasibility analysis of data compression techniques. We achieved our goal comparing global descriptors and mid-level representation. Some approaches save energy consumption in mobile devices because they send more compact feature vector to be processed on the server side. We also conducted a series of experiments to evaluate aspects of effectiveness, efficiency and compactness of extracted features of images in order to perform content-based image retrieval on mobile devices. In this case, the user decides the best triple trade-off configuration regarding effectiveness, efficiency, and compactness of visual features using more or less resources on the mobile devices. We present several results (tables and graphs) to facilitate accurate analysis for image retrieval considering different datasets and image classes. We also present confusion matrices that assist in the identification of classes of images in which a certain type of algorithm has problems to retrieve images efficiently.

Experimental analysis showed that most suitable descriptors are BIC (Border/Interior Pixel Classification - a color descriptor), LCH (Local Color Histogram - a color descriptor), LAS (Local Activity Spectrum - a texture descriptor), DEOBSM (Bag of Words using Dense Sampling, ORB descriptor, Soft assignment and Maximum pooling), DEBFSM (Bag of Words using Dense Sampling, BRIEF descriptor, Soft assignment and Maximum pooling) and FTBKSM (Bag of Words using FAST Sampling, BRISK descriptor, Soft assignment and Maximum pooling). Considering the analysis of effectiveness, efficiency and compactness presented, we can draw our conclusions about the best global descriptors and bag of words representations to be used in the mobile image search scenario: 1) BIC (Border/Interior Pixel Classification) [Stehling et al., 2002] and 2) DEOBSM (Bag of Words using Dense Sampling, ORB descriptor,

Soft assignment and Maximum pooling), respectively. We did some preliminary tests in the smartphone LG Nexus 5 using BIC and DEOBSM. As a result, for images dimension of 300×500, the feature extraction of BIC takes about 300 milliseconds and DEOBSM takes around 500 milliseconds. We note that the BIC descriptor seems to outperform DEOBSM in almost cases. Paired statistical test in ten datasets showed that BIC can be considered better then DEOBSM. Therefore, for mobile visual retrieval, we may consider use BIC descriptor as the best option considering the triple trade-off problem regarding efficiency, effectiveness and compactness.

We performed several experiments that indicates that dense sampling is more accurate than interest point detection (sparse sampling) to compute bag-of-visual-word features. Sparse sampling look for points in images with properties such as repeatibility, which may create less ambiguity and are in discriminative regions. On the other hand, dense sampling, gives a better coverage of the entire object or scene and a constant amount of features per image area. In order to find the best sampling strategy to use in a mobile visual retrieval approach we use a dense sampling strategy and six keypoint detectors (FAST, GFTT, GFTTHarris, MSER, ORB detector, SURF detector) to create bag of words representation in order to find out the best configuration and use them in a mobile visual search application. The experiments were performed in two images datasets: WANG (1,000 images) and 15Scenes (4,485 images). We did several tests with sparse sampling, selecting the best combination of binary descriptor with a keypoint detector compared with the results of dense sampling. Paired statistical test (t-student distribution) with 95% of confidence showed that the best approach of dense sampling (DEOBSM = Bag of Words using Dense Sampling, ORB descriptor, Soft assignment and Maximum pooling) is better than sparse sampling using the same descriptor (FTOBSM = Bag of Words using FAST Algorithm Sampling, BRISK descriptor, Soft assignment and Maximum pooling) in the two datasets analyzed. Between the restriction of use dense sampling on mobile devices is that the images should not be so big, because if the image is big the dense sampling is slow. For this reason, is important resize images extremely big. Another option is to perform a more sparse dense sampling.

In this thesis, we use bag of words representation with the size (k) of 1024 as in [Ken Chatfield and Zisserman, 2011; Zhou et al., 2010b; Caetano et al., 2014a; Pessoa et al., 2015a], but paired statistical test in the Caltech101 dataset shows that use BoW with k = 1024 is better than BoW k = 256, but is not different than BoW with k = 512 and k = 2048, where K is the number of bins on the Bag of Words representations (BoW). Therefore, for mobile visual retrieval we may consider use bag of words with small size (for example, k = 512). For retrieval experiments, which are generally based

on computing distances between vectors, with the Euclidean or Manhattan distance, for example, vectors should be compact, or embedded in an index structure, to avoid the curse of the dimensionality. We also presented additional concepts related to visual dictionary model.

In Chapter 5, we proposed two approaches of extracting spatial information on images to improve the quality of image representation. These approaches are called BOBGrid (Spatial Bag of BIC Grid) and BOBSlic (Spatial Bag of Slic BIC). We compare them against WSA (Visual Word Spatial Arrangement) [Penatti et al., 2014] and BossaNova (Bag Of Statistical Sampling Analysis) [Avila et al., 2013]. In statistical analyzes, both BOBGrid and BOBSlic outperform our baseline of spatial bag of visual words WSA and the BossaNova algorithm in the WANG dataset. In addition, descriptors are more compact, which make them more suitable for mobile devices applications. Our approaches have vectors of length 128, while the baselines have size 1024. As aforementioned, for retrieval experiments, vectors should be compact to avoid the curse of the dimensionality. We could observe that BOBSlic seems to be better than BOBGrid because it uses a segmentation approach to represent bag of words.

In this thesis, we developed an Image Retrieval system to smartphone using Android Development that is not available, but a Web Interface is available on `goldenretriever.dcc.ufmg.br` (Attachment F, Figure F.1). The VPRRetriever (or GoldenRetriever) seeks to do a comparative study of different features and distance metrics in order to analyze the impact of these factors in the process of Content-Base Image Retrieval. Examples of image retrieval using BIC descriptor on the VPRRetriever System are shown on Attachment F.

6.1 Future Work

The contributions presented in this thesis focus primarily on provide suitable representation for the amount of local features extraction and the most suitable configurations in terms of effectiveness, efficiency, and compactness. The challenge here is how to extend the analysis and approaches proposed in this work. Thus, as future works, we propose the following possible research solutions:

1. *To perform more experimental analysis on mobile devices:* We are developing a prototype system for retrieving the content of general image by using mobile devices as query interface. We plan to develop suitable Android interfaces to support user on automatic search sections. The client module is a mobile application

implemented for Android platform[1].

2. *To evaluate algorithms of text processing and analyze a multimodal approach using text and image features together to improve the similarity search:* Recently severals image dataset also have tags (text information) associated with images. Thus, it is important to use image retrieval model (such as Vector model, Probabilistic model and Best Match model [Baeza-Yates and Ribeiro-Neto, 2011]) as well as techniques to indexing of documents (in our case, images with tags) for image retrieval. The idea is to use text information available on site and social networking accessed by mobile phones.

3. *To exploit more algorithms which use semantic or spatial information:* We plan to analyze others configurations of parameters in the case of BossaNova and other sparse sampling in the case of WSA and also use new approches of spatial information enconding, comparing them with our proposed approaches: BOBGrid and BOBSlic.

4. *To propose a method to select the best descriptors to use in an average rank aggregation approach:* Commonly, CBIR approaches compute distance measures considering only pairs of images, ignoring the rich information encoded in the relationships among images. Because of this, we plan to use rank aggregation approaches to combine results produced by different image descriptors, improving the final ranking.

5. *To develop deep learning algorithms on mobile devices in a client-server architecture:* With the evolution of smartphones, mobile phones start to have more processing resources such as GPU processing. Therefore, the idea is to use GPU processing to speed up feature extraction and the re-ranking computation. We also plan to use deep-learning techniques to improve the quality of image retrieval in a client-server architecture.

[1]Android platform - `http://developer.android.com/`

Appendix A

Distances: Euclidean Vs Manhattan

Pessoal et al [Pessoa et al., 2015a], presents a study on low-Cost representations (bag of words of five binary descriptors using manhattan distance) for image feature extraction on mobile devices, however, after statistical analyses using paired statistical test, we concluded that manhattan distance is better than euclidean distance with 95% of confidence. So, in the thesis we just use Manhattan distance (L1 distance). See Figure A.1.

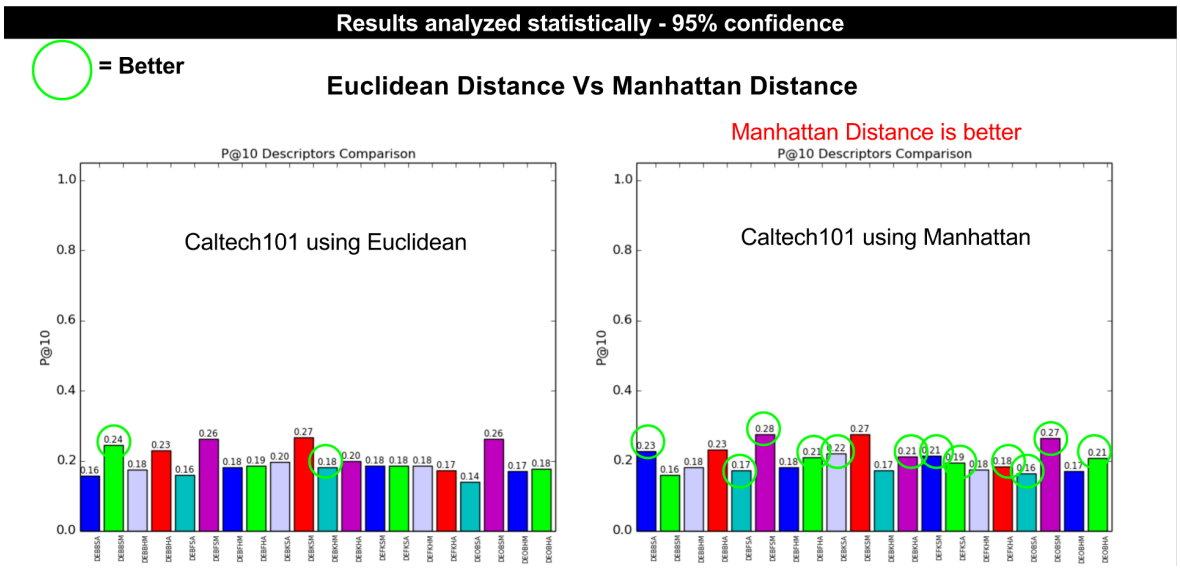


Figure A.1: Paired statistical test showed that Manhattan Distance is better than Euclidean Distance in almost all ten datasets analysed. We used a confidence of 95% and ANOVA (Analysis of Variance). DE = Dense sampling, BB = BinBoost, BF = BRIEF, BK = BRISK, FK = FREAK, OB = ORB, HA = Hard assignment with Average pooling, HM = Hard assignment with Maximum pooling, SA = Soft assignment with Average pooling, SM = Soft assignment with Maximum pooling.

Appendix B

Impact of the Bag of Words Representations Size

In this thesis we use bag of words representation with the size (k) of 1024 as in [Ken Chatfield and Zisserman, 2011; Zhou et al., 2010b; Caetano et al., 2014a; Pessoa et al., 2015a]. Paired statistical test in the Caltech101 dataset shows that use BoW with k = 1024 is better than BoW k = 256, but is not different than BoW with k = 512 and k = 2048, where k is the number of bins on the Bag of Words representations (BoW) (See Figure B.1). Therefore, for mobile visual retrieval we may consider use bag of words with small size (for example k = 512). For retrieval experiments, which are generally based on computing distances between vectors, with the Euclidean or Manhattan distance, for example, vectors should be compact, or embedded in an index structure, to avoid the curse of the dimensionality [Traina Jr et al., 2002; Hoàng et al., 2010; Kang et al., 2011; Weber et al., 1998].

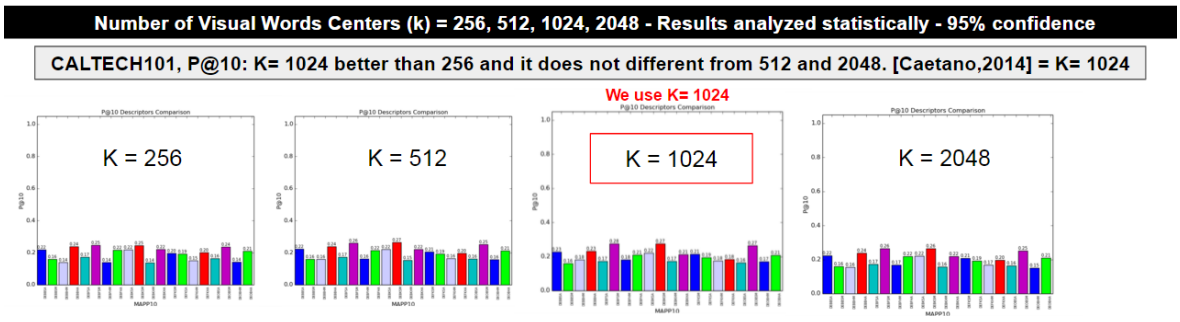


Figure B.1: We use bag of words representation with the size of 1024

Appendix C

MAP Overall - All Datasets

1. **15Scenes:** Figure C.1 (BoW Descriptors), Figure C.2 (Global Descriptors).
2. **OxBuild11:** Figure C.3 (BoW Descriptors), Figure C.4 (Global Descriptors).
3. **Paris:** Figure C.5 (BoW Descriptors), Figure C.6 (Global Descriptors).
4. **ZuBuD:** Figure C.7 (BoW Descriptors), Figure C.8 (Global Descriptors).
5. **SMVS692:** Figure C.9 (BoW Descriptors), Figure C.10 (Global Descriptors).
6. **WANG:** Figure C.11 (BoW Descriptors), Figure C.12 (Global Descriptors).
7. **Caltech101:** Figure C.13 (BoW Descriptors), Figure C.14 (Global Descriptors).
8. **Caltech256:** Figure C.15 (BoW Descriptors), Figure C.16 (Global Descriptors).
9. **VOC2007:** Figure C.17 (BoW Descriptors), Figure C.18 (Global Descriptors).
10. **UW:** Figure C.19 (BoW Descriptors), Figure C.20 (Global Descriptors).

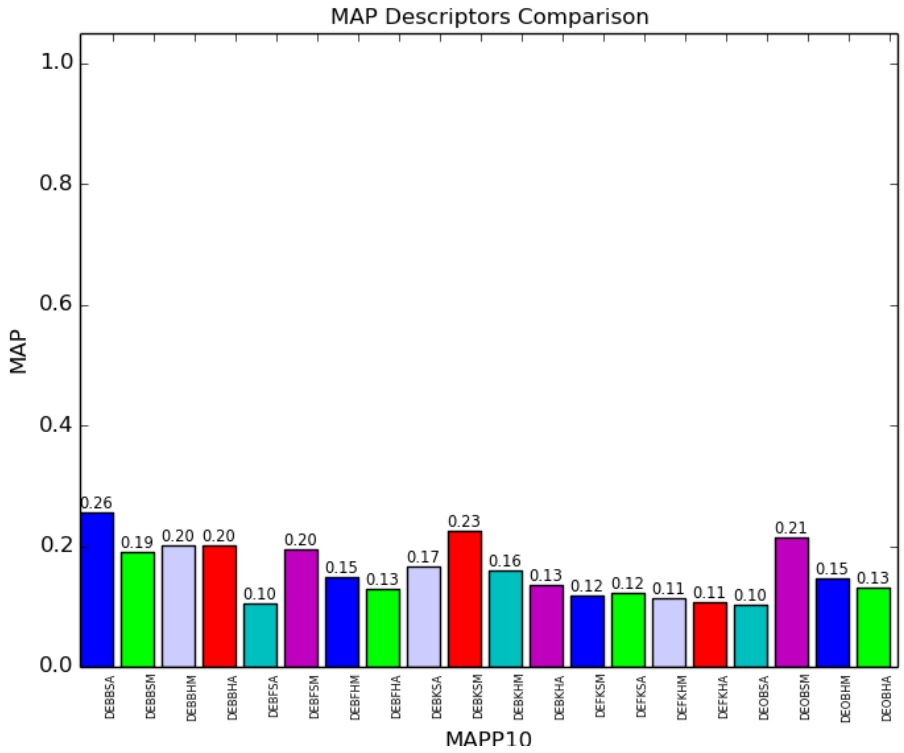


Figure C.1: Overall MAP to 15Scenes dataset (4,485 images) – BoW Descriptors.

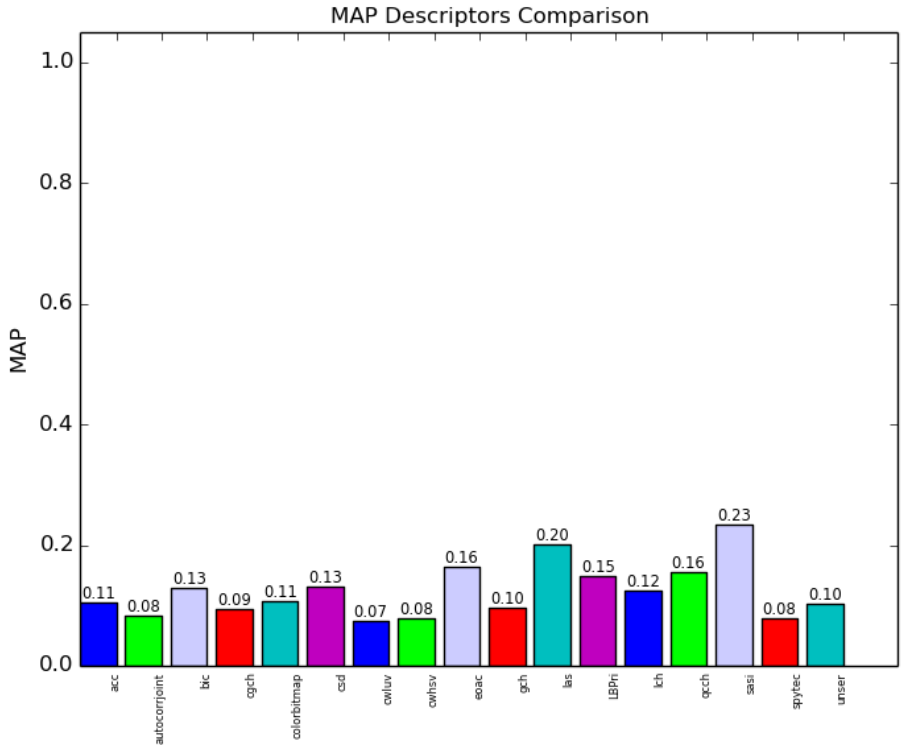


Figure C.2: Overall MAP to 15Scenes dataset (4,485 images) – Global Descriptors.

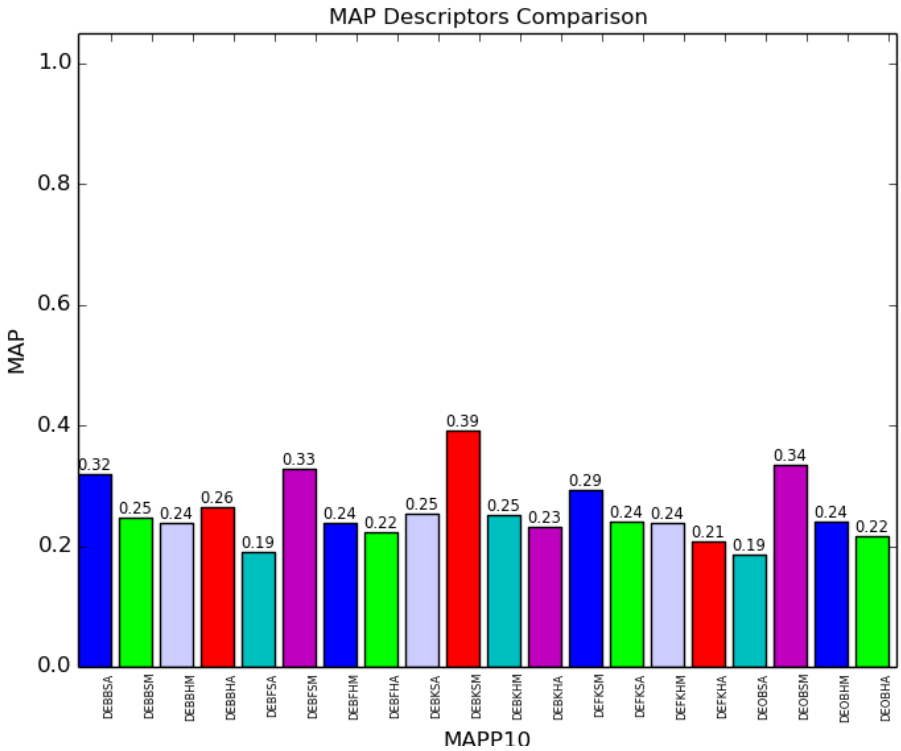


Figure C.3: Overall MAP to OxBuild11 dataset (567 images) – BoW Descriptors.

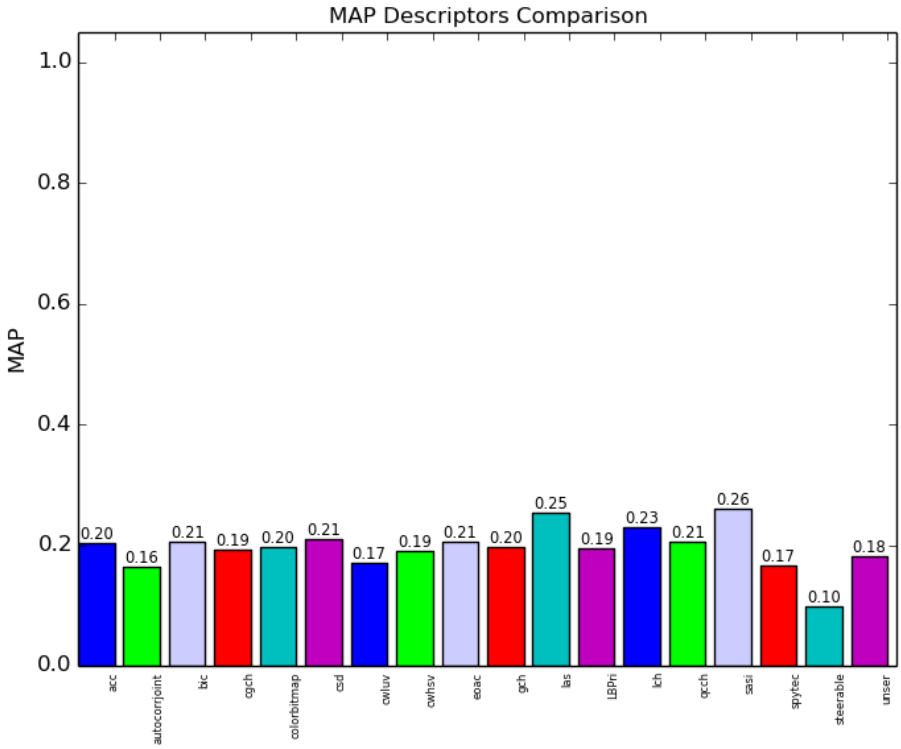


Figure C.4: Overall MAP to OxBuild11 dataset (567 images) – Global Descriptors.

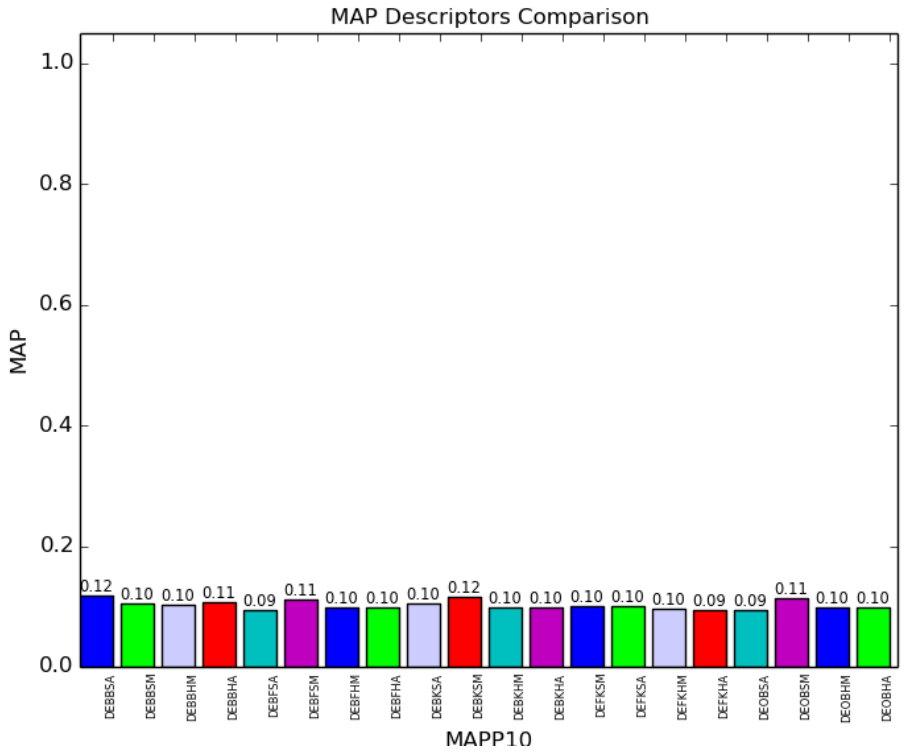


Figure C.5: Overall MAP to Paris dataset (6,392 images) – BoW Descriptors.

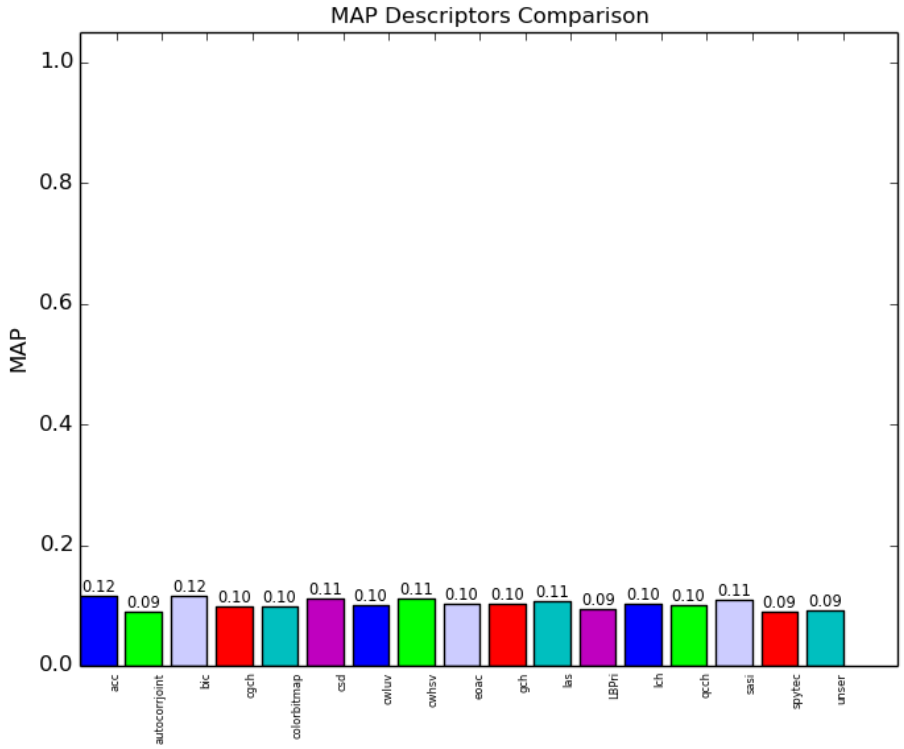


Figure C.6: Overall MAP to Paris dataset (6,392 images) – Global Descriptors.

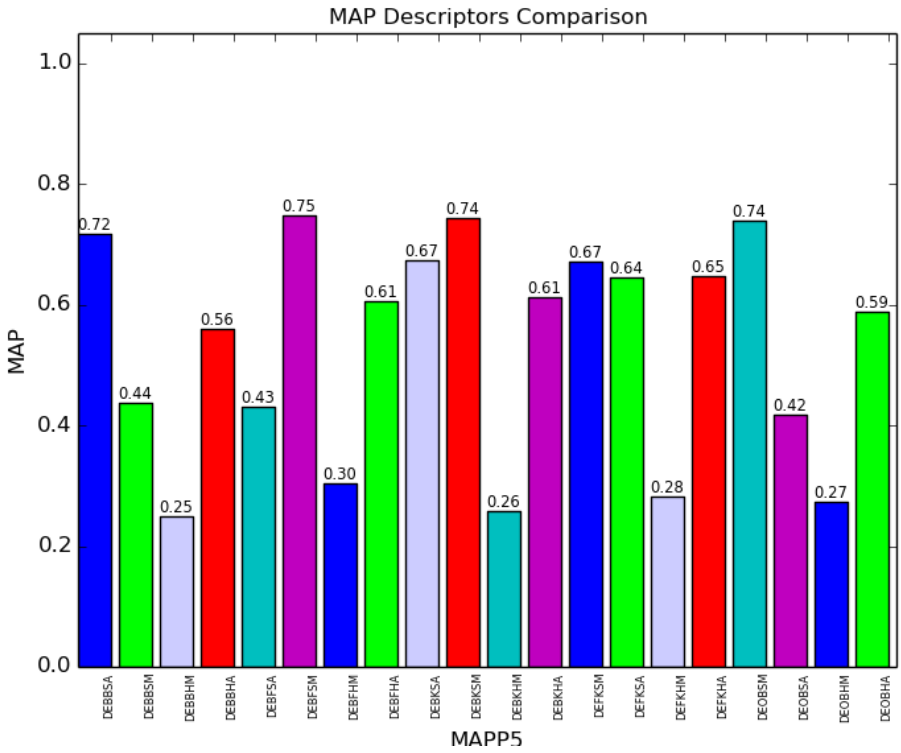


Figure C.7: Overall MAP to ZuBuD dataset (1,005 images) – BoW Descriptors.

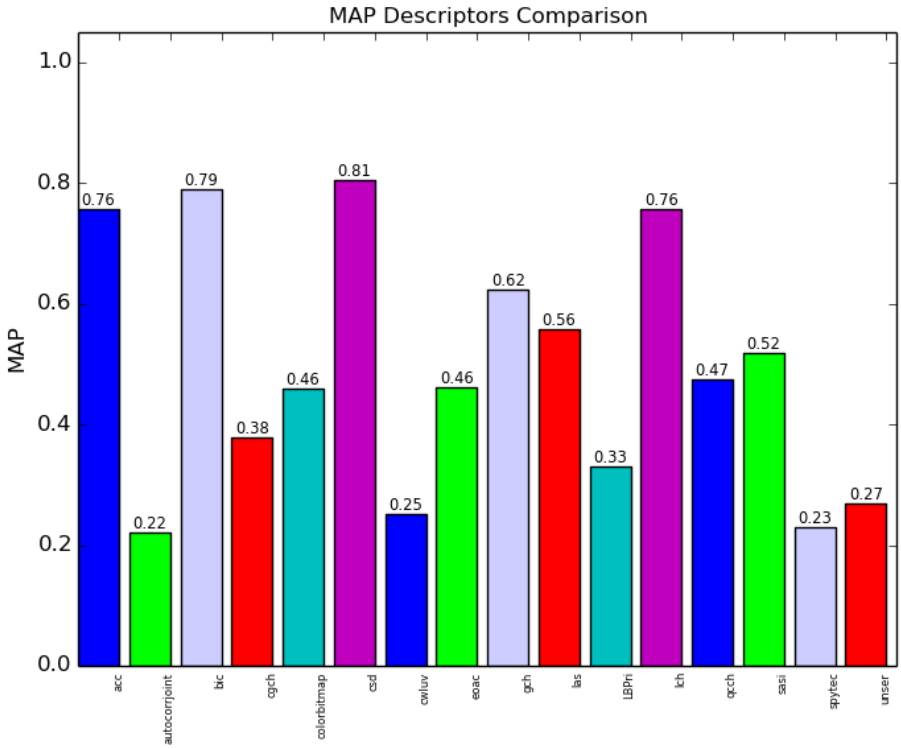


Figure C.8: Overall MAP to ZuBuD dataset (1,005 images) – Global Descriptors.

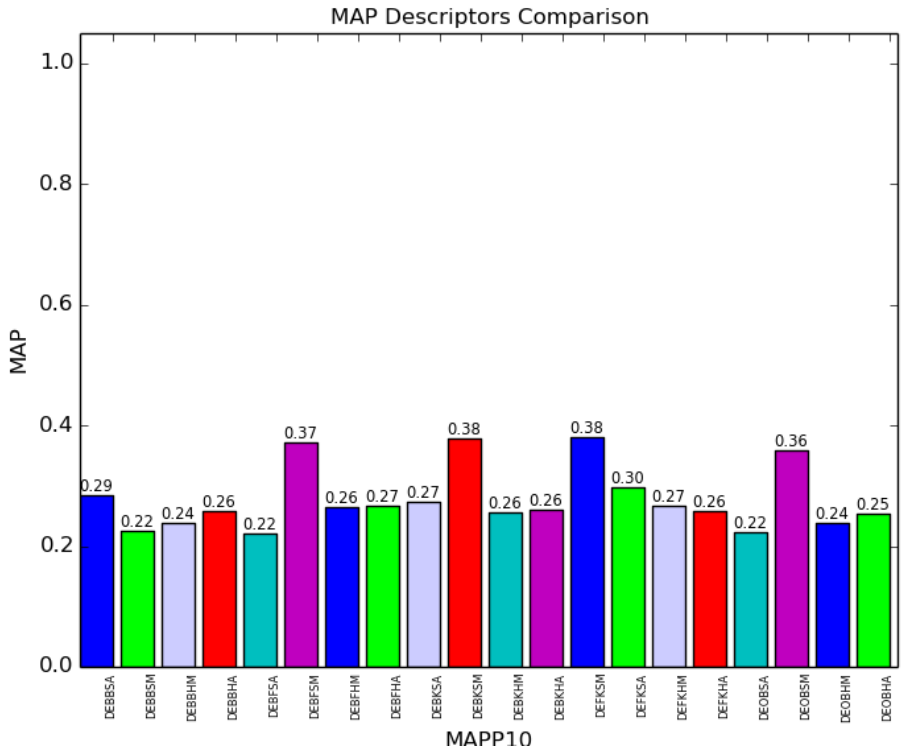


Figure C.9: Overall MAP to SMVS692 dataset (3,460 images) – BoW Descriptors.

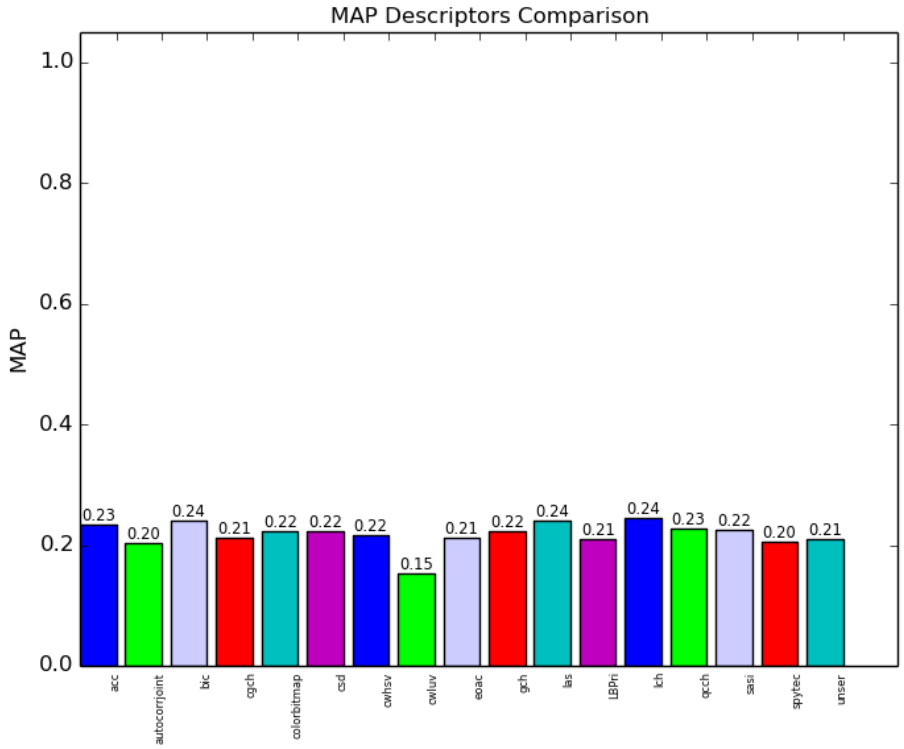


Figure C.10: Overall MAP to SMVS692 dataset (3,460 images) – Global Descriptors.

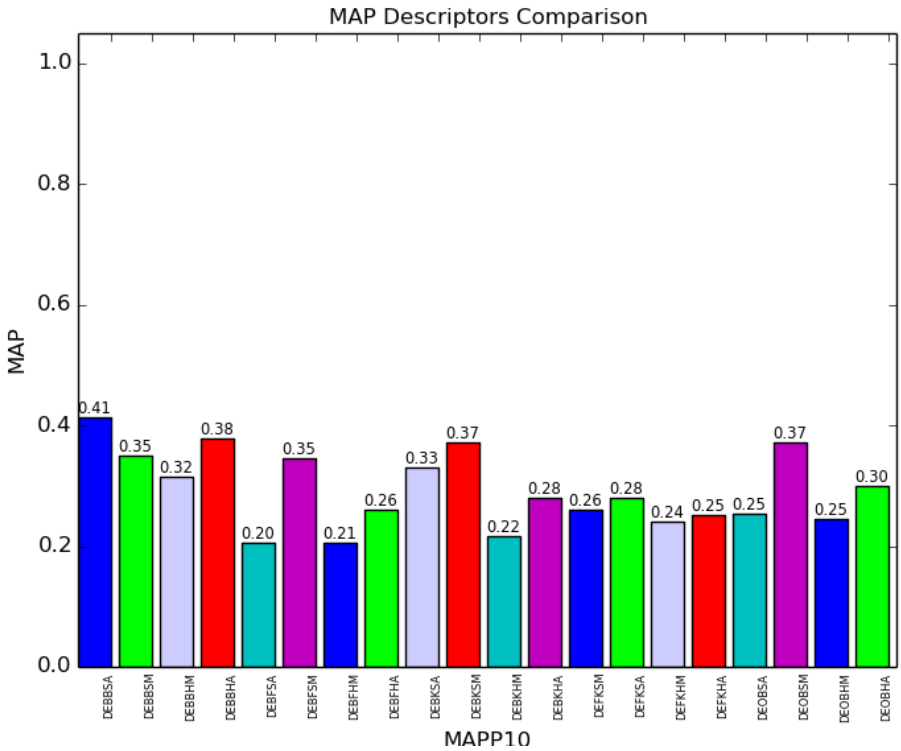


Figure C.11: Overall MAP to WANG dataset (1,000 images) – BoW Descriptors.

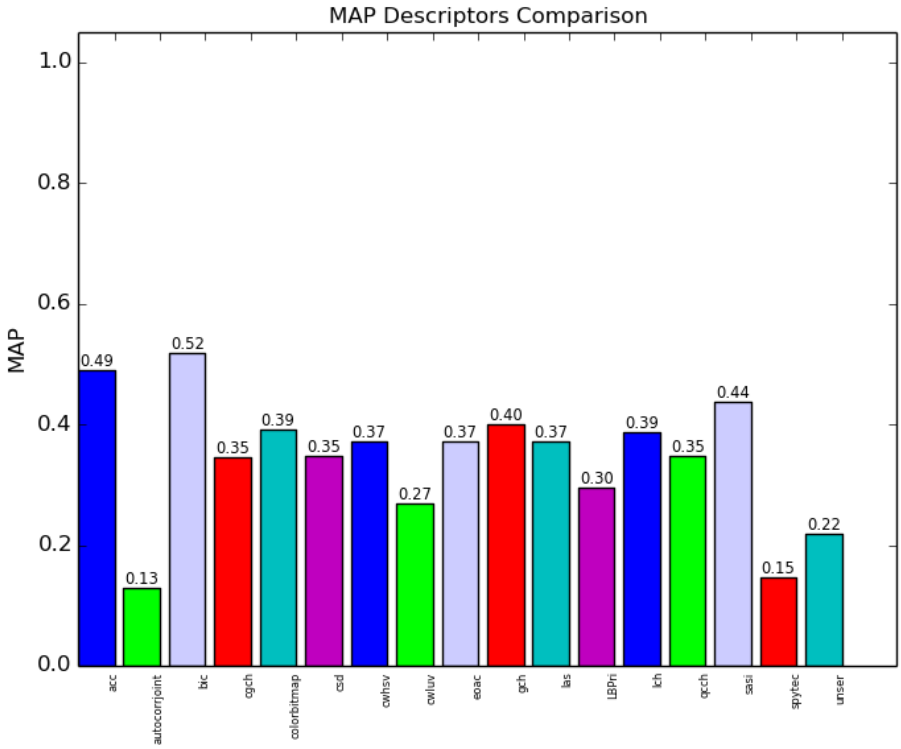


Figure C.12: Overall MAP to WANG dataset (1,000 images) – Global Descriptors.

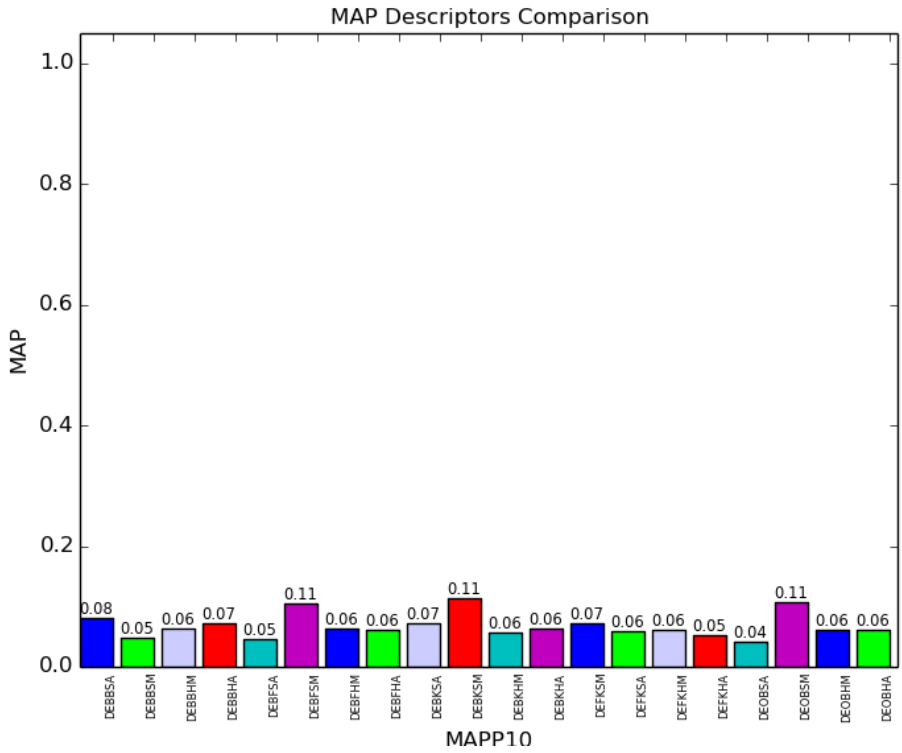


Figure C.13: Overall MAP to caltech101 dataset (9,144 images) – BoW Descriptors.

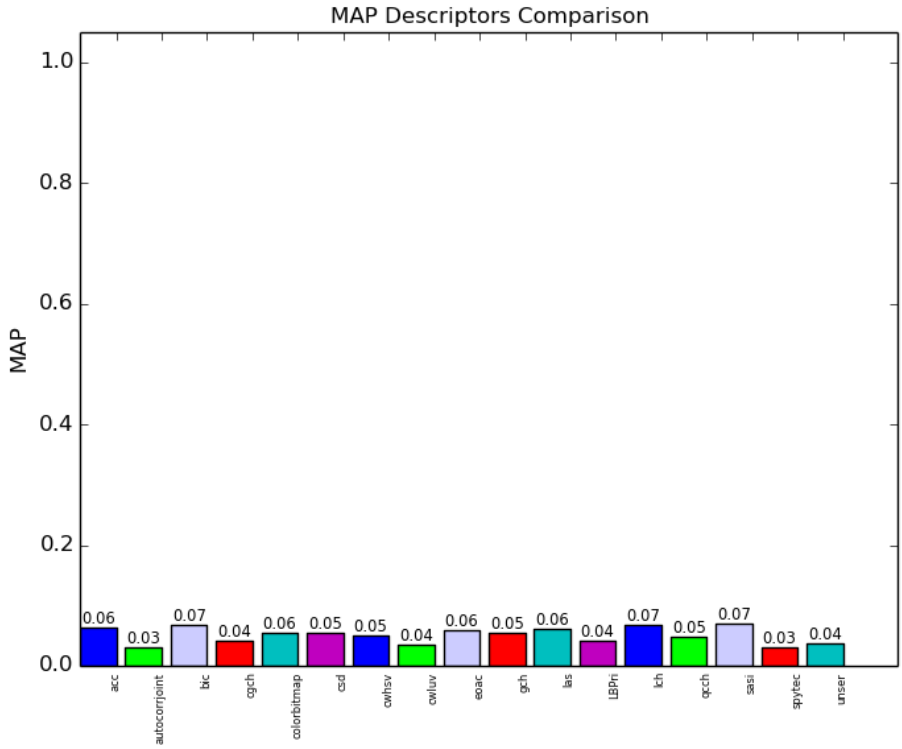


Figure C.14: Overall MAP to caltech101 dataset (9,144 images) – Global Descriptors.

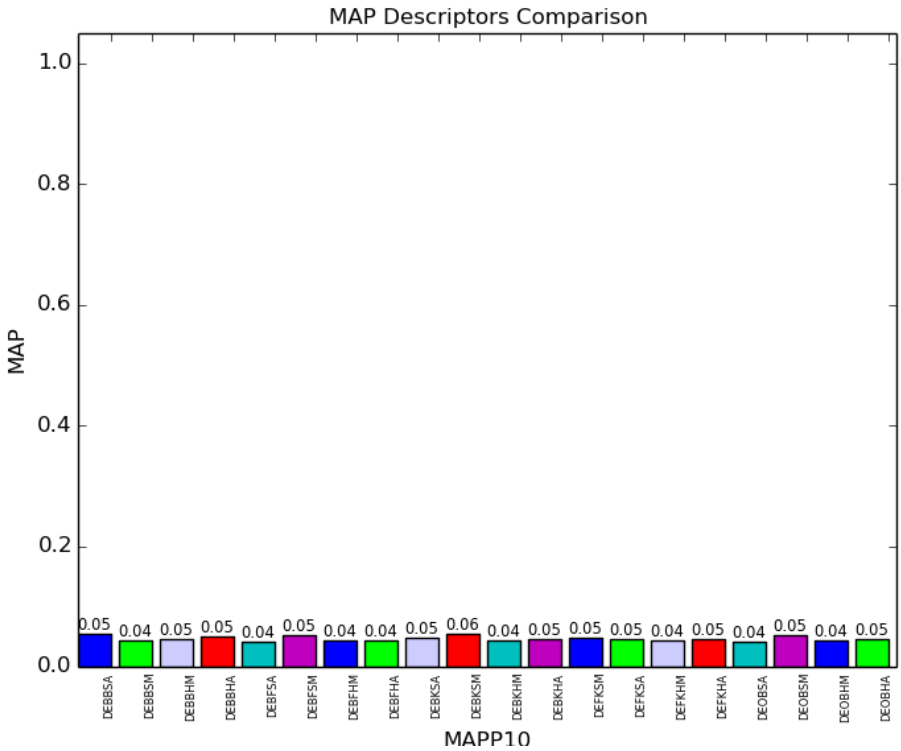


Figure C.15: Overall MAP to caltech256 dataset (30,607 images) – BoW Descriptors.

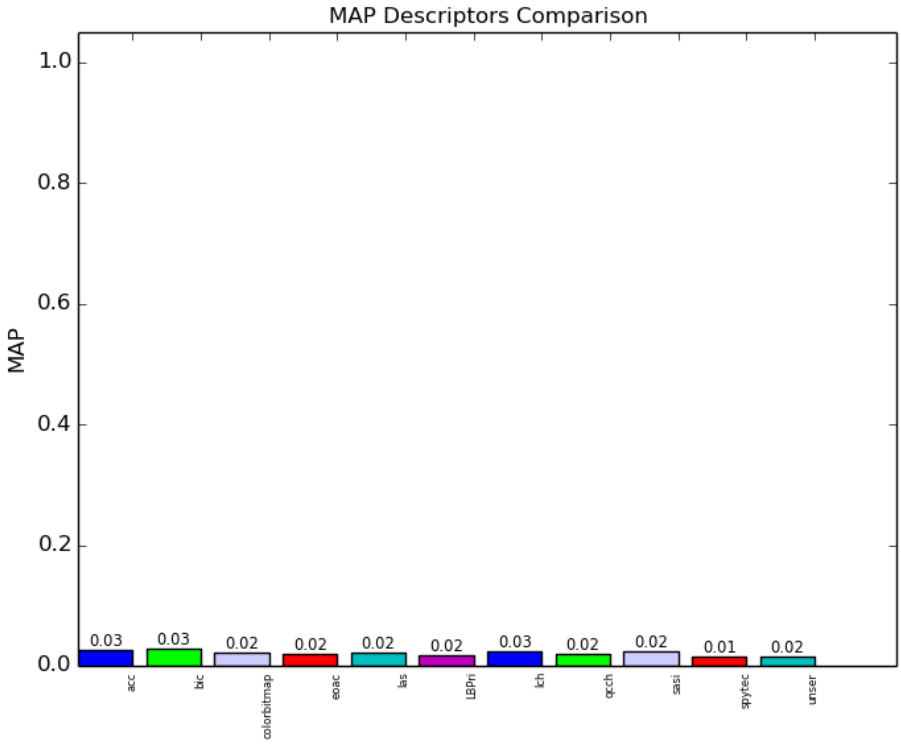


Figure C.16: Overall MAP to caltech256 (30,607 images) – 11 Best Global Descriptors.

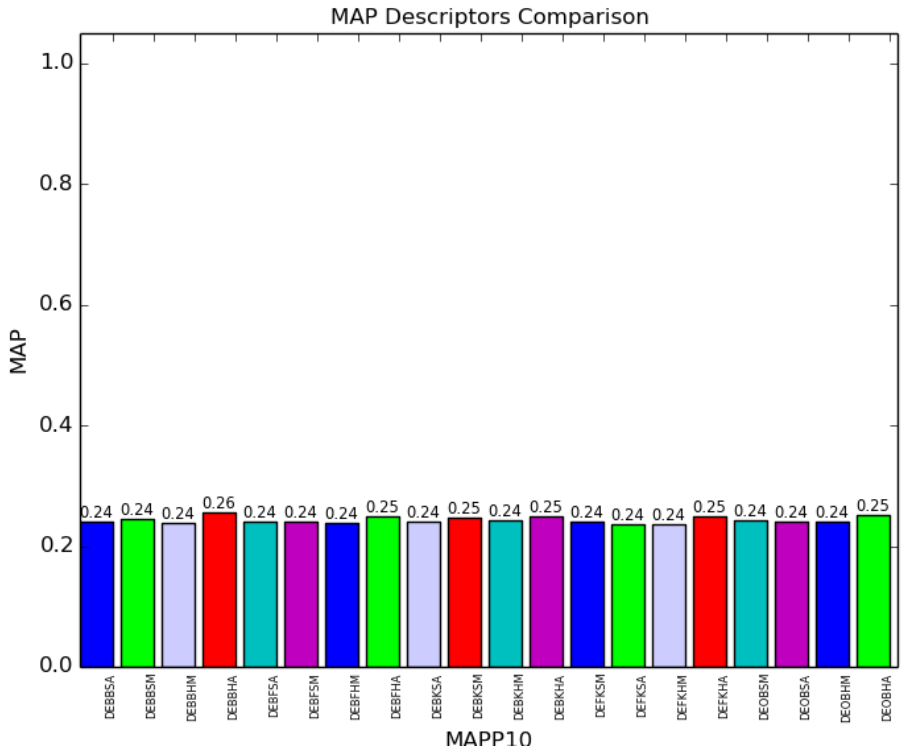


Figure C.17: Overall MAP to VOC2007 dataset (9,963 images) – BoW Descriptors.

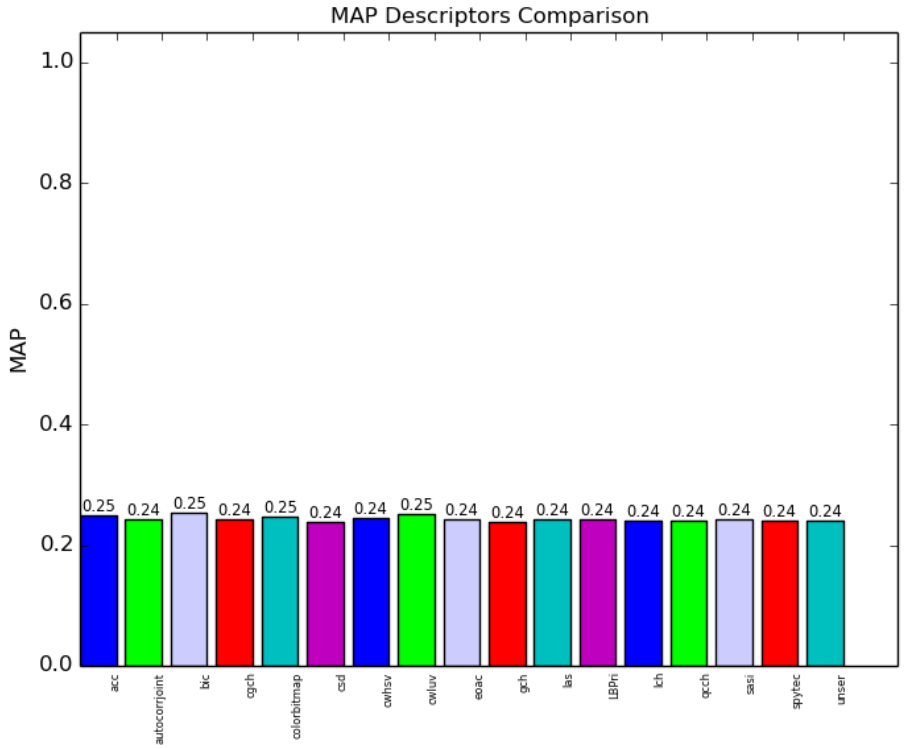


Figure C.18: Overall MAP to VOC2007 dataset (9,963 images) – Global Descriptors.

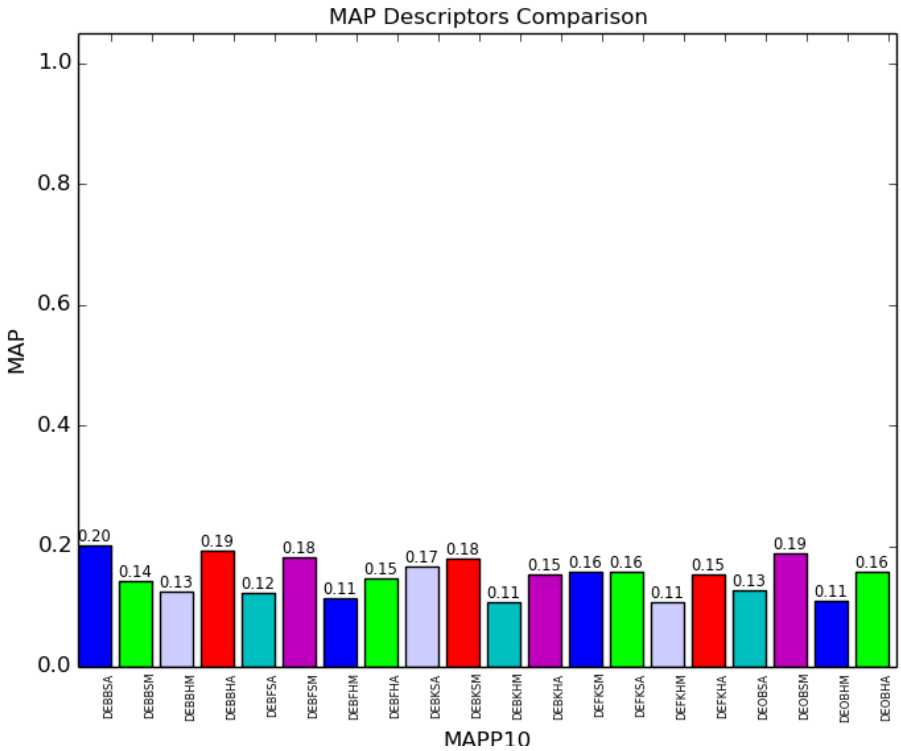


Figure C.19: Overall MAP to UW dataset (1,009 images) – BoW Descriptors.

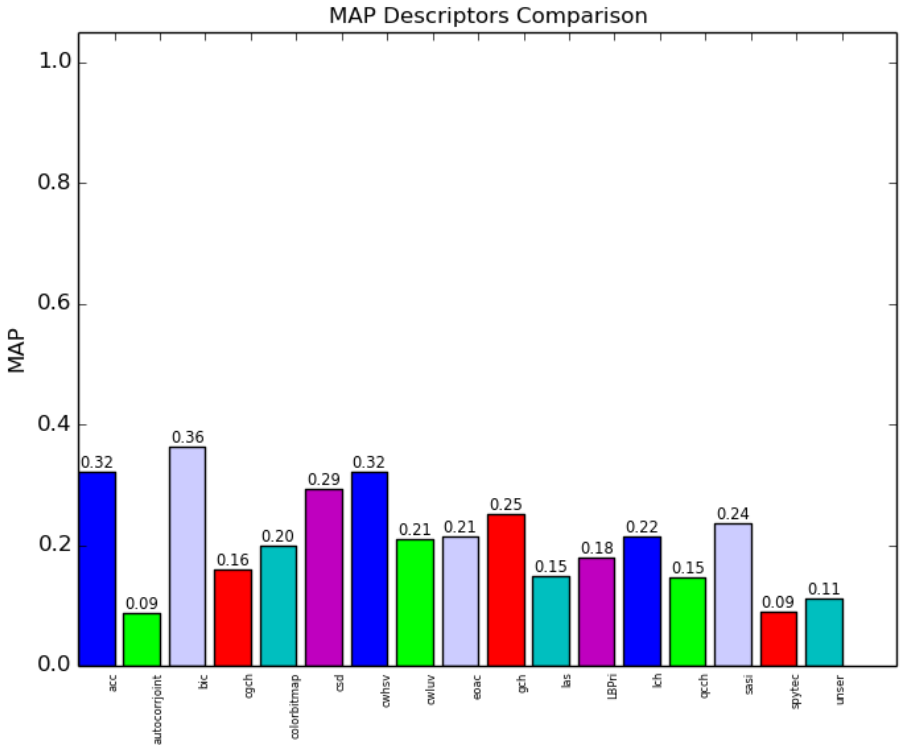


Figure C.20: Overall MAP to UW dataset (1,009 images) – Global Descriptors.

Appendix D

P@5/P@10 Overall - All Datasets

1. **15Scenes:** Figure D.1 (BoW Descriptors), Figure D.2 (Global Descriptors).
2. **OxBuild11:** Figure D.3 (BoW Descriptors), Figure D.4 (Global Descriptors).
3. **Paris:** Figure D.5 (BoW Descriptors), Figure D.6 (Global Descriptors).
4. **ZuBuD:** Figure D.7 (BoW Descriptors), Figure D.8 (Global Descriptors).
5. **SMVS692:** Figure D.9 (BoW Descriptors), Figure D.10 (Global Descriptors).
6. **WANG:** Figure D.11 (BoW Descriptors), Figure D.12 (Global Descriptors).
7. **Caltech101:** Figure D.13 (BoW Descriptors), Figure D.14 (Global Descriptors).
8. **Caltech256:** Figure D.15 (BoW Descriptors), Figure D.16 (Global Descriptors).
9. **VOC2007:** Figure D.17 (BoW Descriptors), Figure D.18 (Global Descriptors).
10. **UW:** Figure D.19 (BoW Descriptors), Figure D.20 (Global Descriptors).

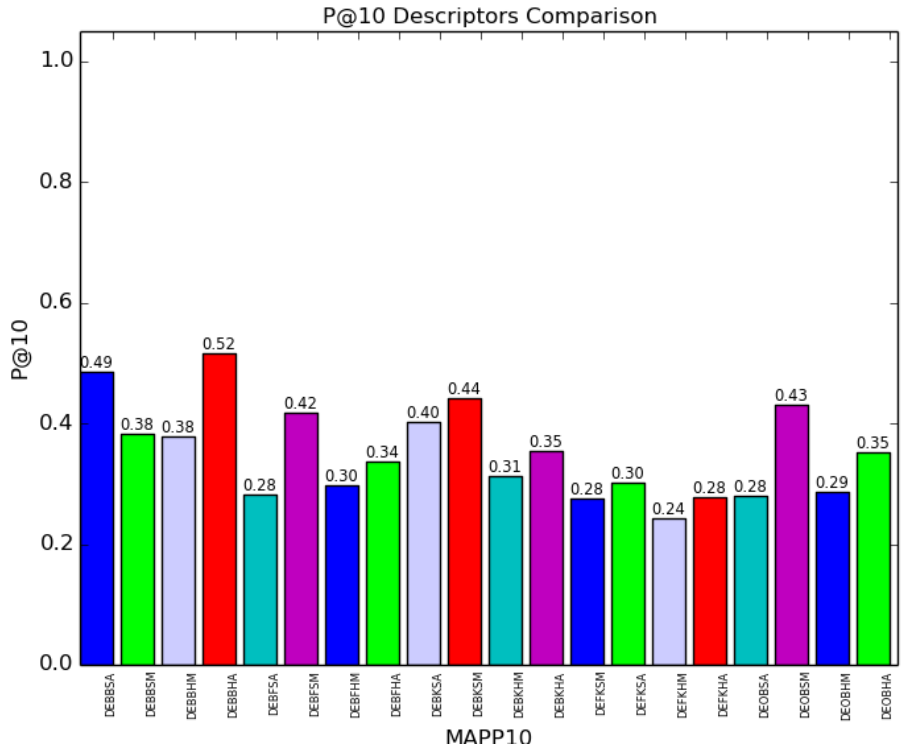


Figure D.1: Overall P@10 to 15Scenes dataset (4,485 images) – BoW Descriptors.

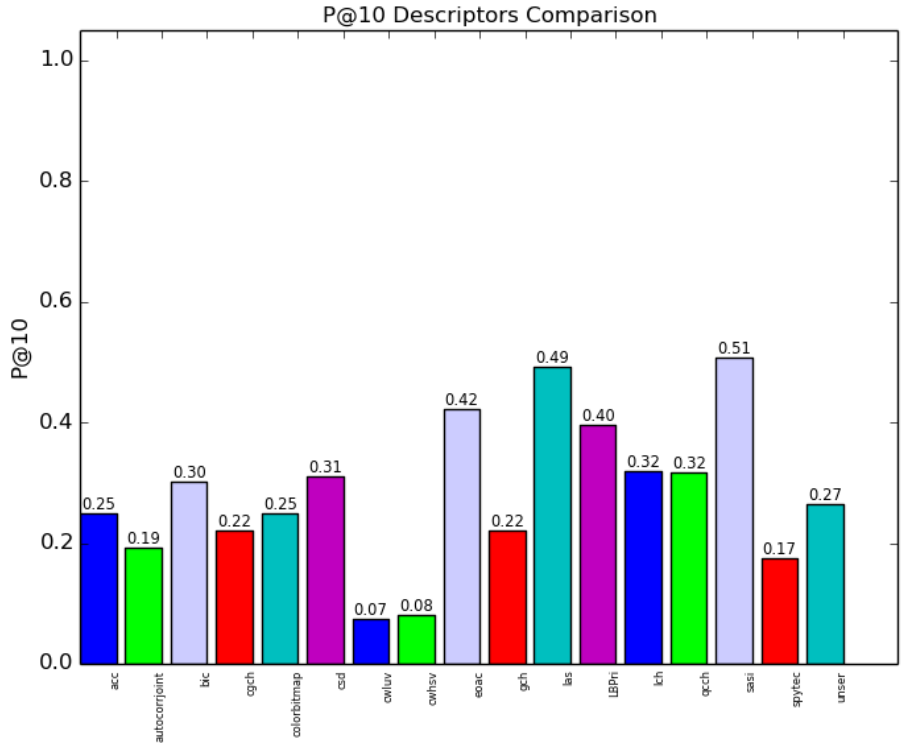


Figure D.2: Overall P@10 to 15Scenes dataset (4,485 images) – Global Descriptors.

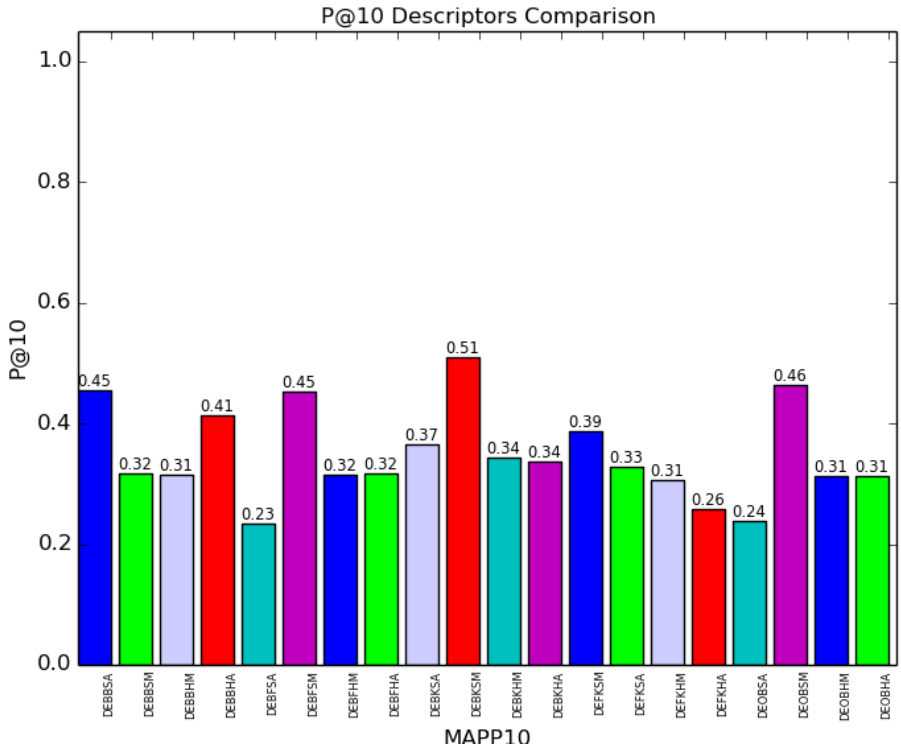


Figure D.3: Overall P@10 to OxBuild11 dataset (567 images) – BoW Descriptors.

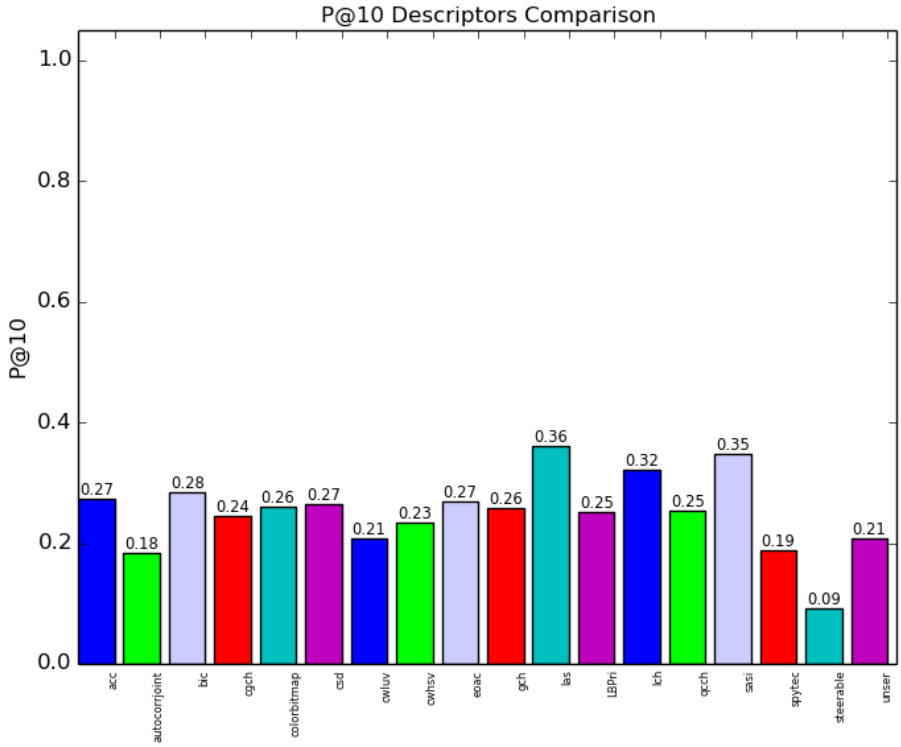


Figure D.4: Overall P@10 to OxBuild11 dataset (567 images) – Global Descriptors.

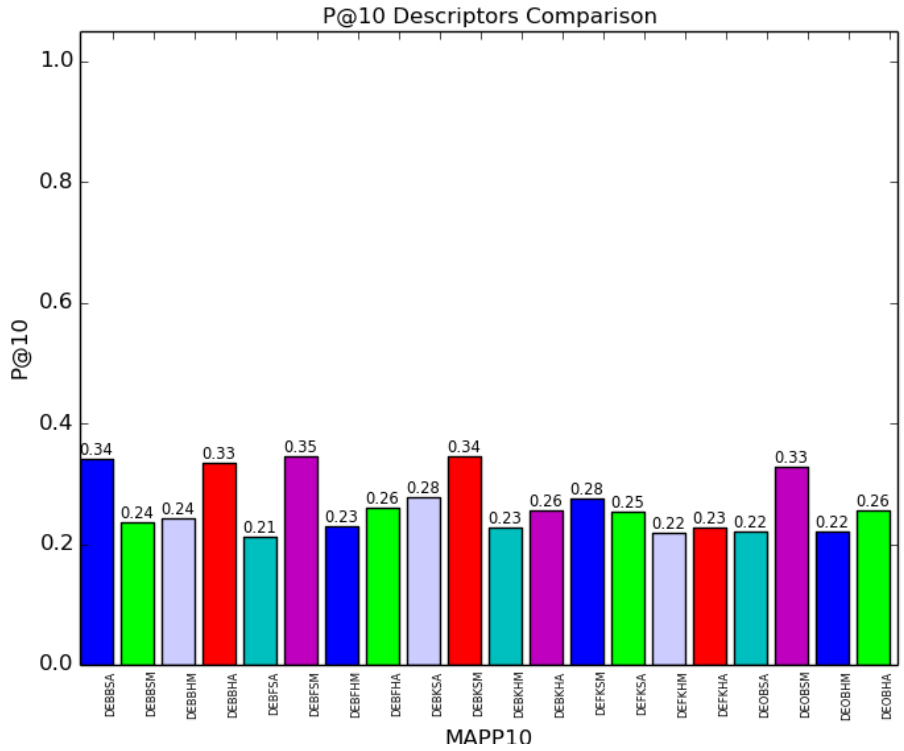


Figure D.5: Overall P@10 to Paris dataset (6,392 images) – BoW Descriptors.

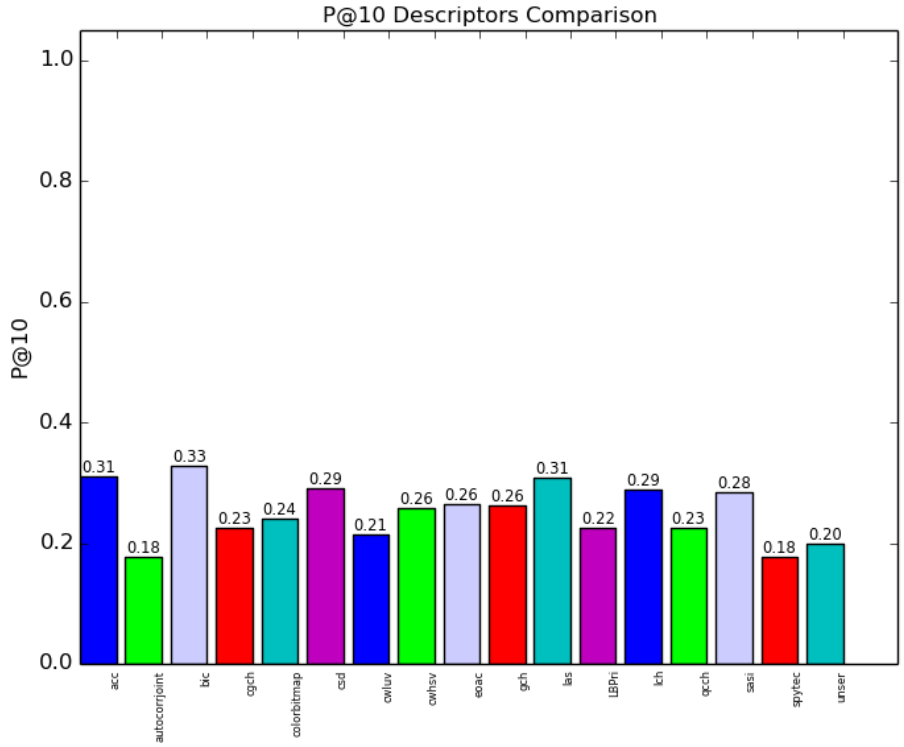


Figure D.6: Overall P@10 to Paris dataset (6,392 images) – Global Descriptors.

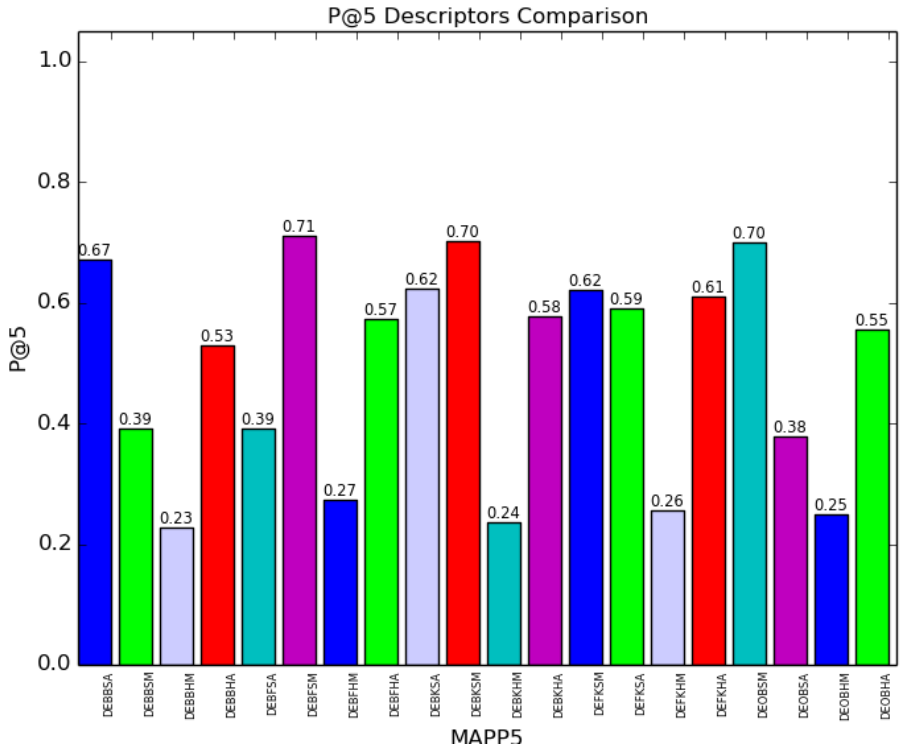


Figure D.7: Overall P@5 to ZuBuD dataset (1,005 images) – BoW Descriptors.

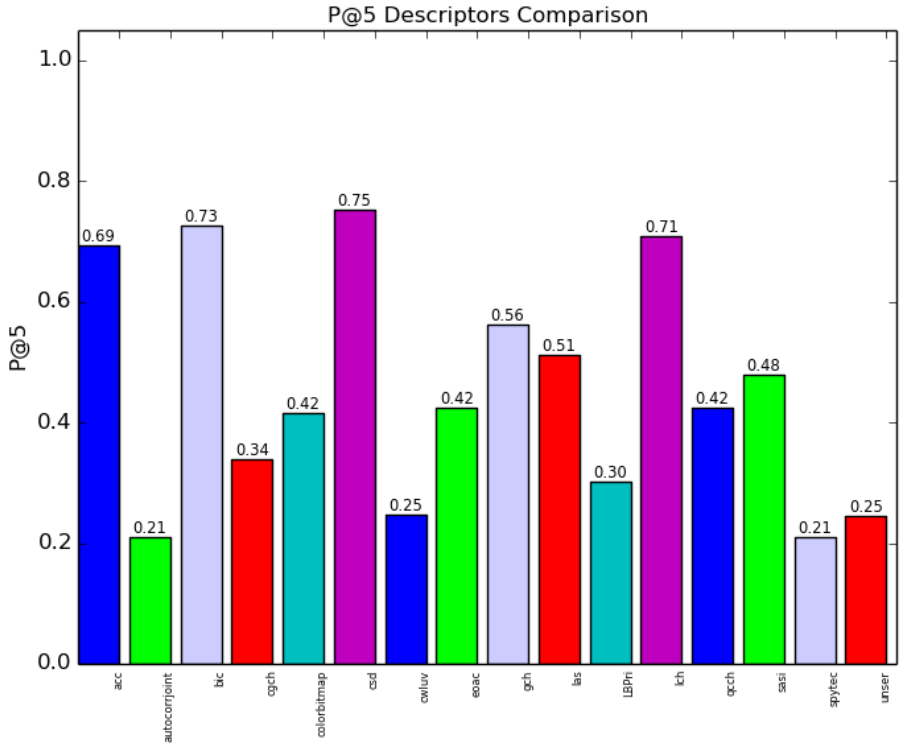


Figure D.8: Overall P@5 to ZuBuD dataset (1,005 images) – Global Descriptors.

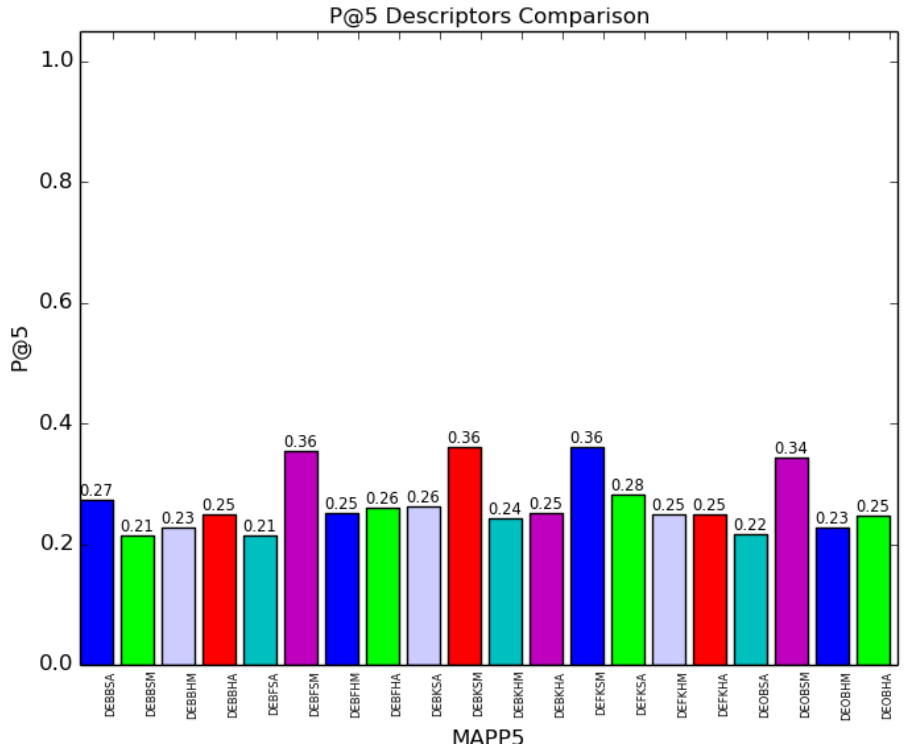


Figure D.9: Overall P@5 to SMVS692 dataset (3,460 images) – BoW Descriptors.

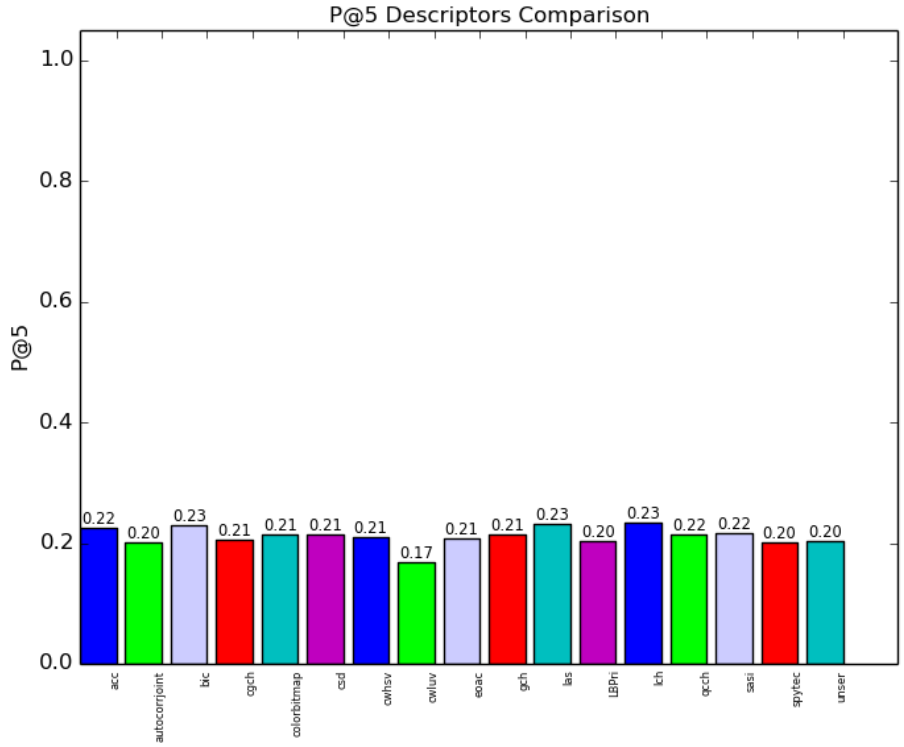


Figure D.10: Overall P@5 to SMVS692 dataset (3,460 images) – Global Descriptors.

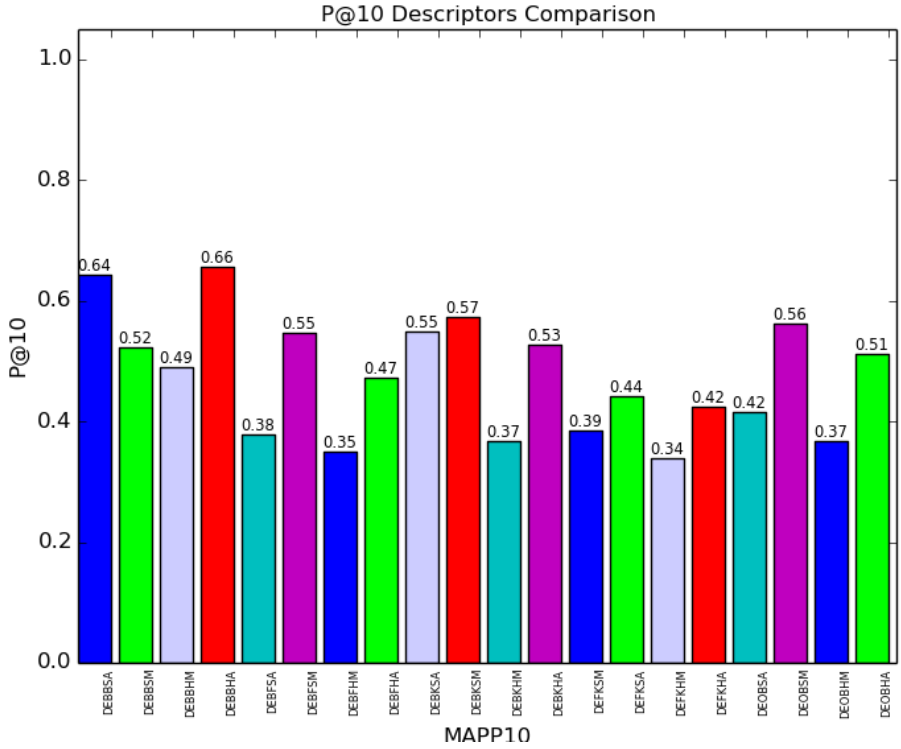


Figure D.11: Overall P@10 to WANG dataset (1,000 images) – BoW Descriptors.

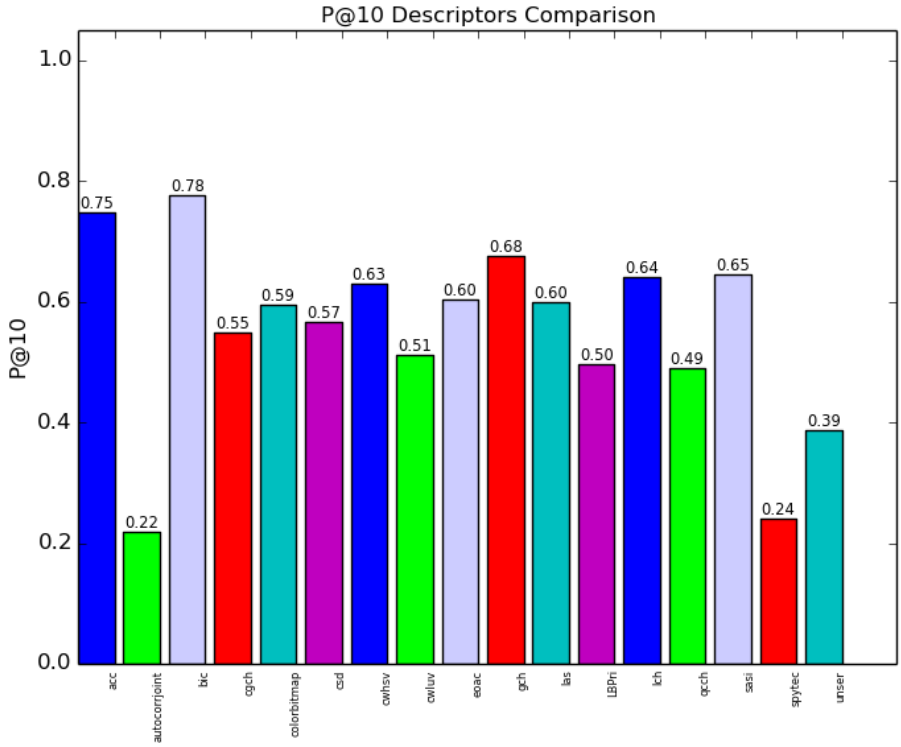


Figure D.12: Overall P@10 to WANG dataset (1,000 images) – Global Descriptors.

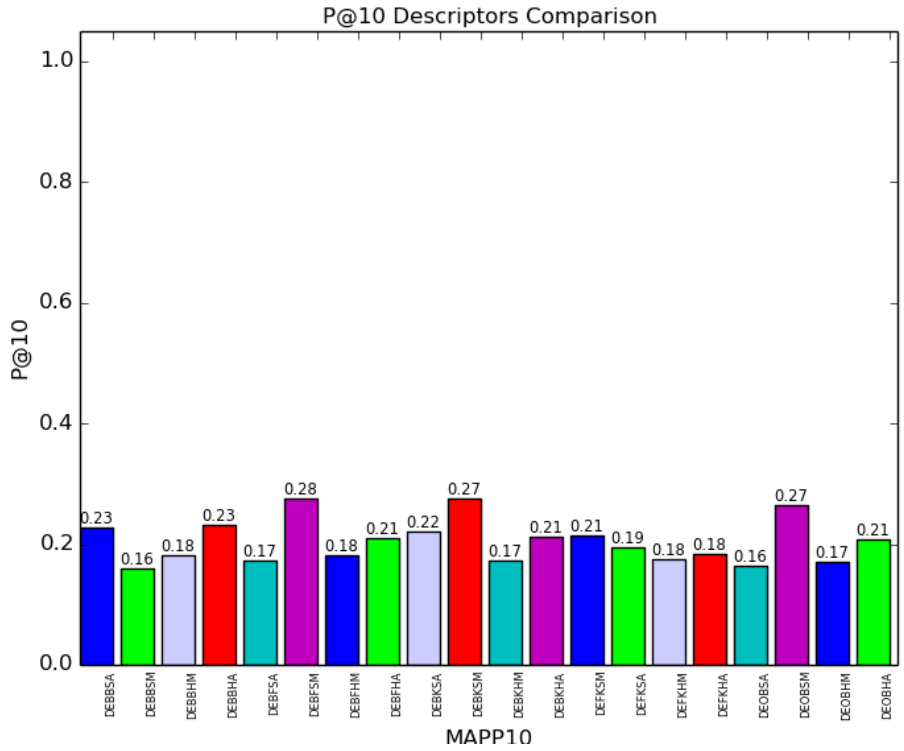


Figure D.13: Overall P@10 to caltech101 dataset (9,144 images) – BoW Descriptors.

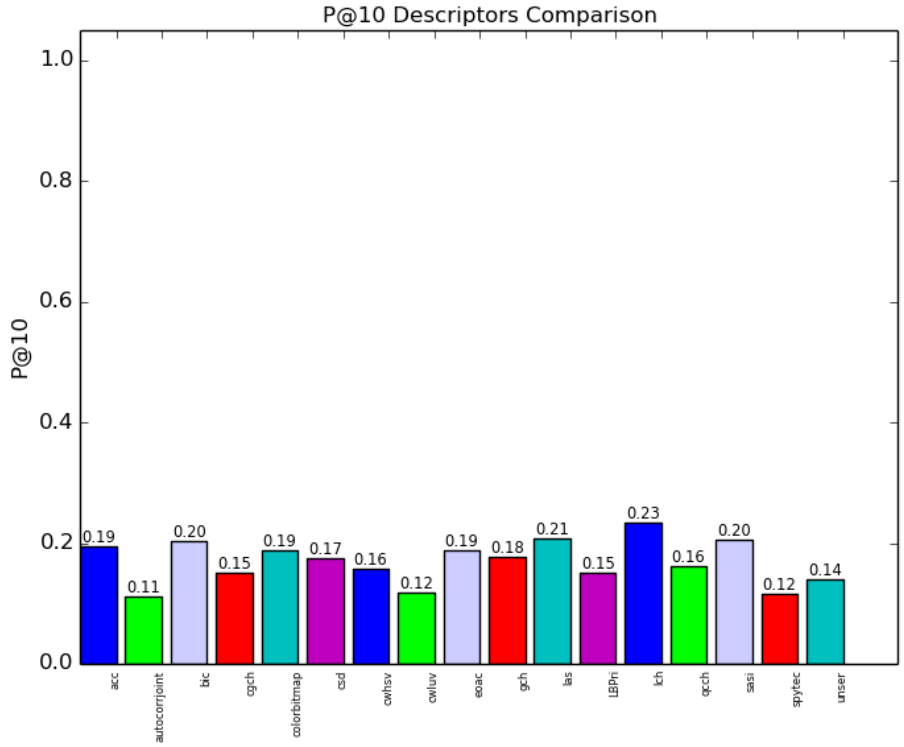


Figure D.14: Overall P@10 to caltech101 dataset (9,144 images) – Global Descriptors.

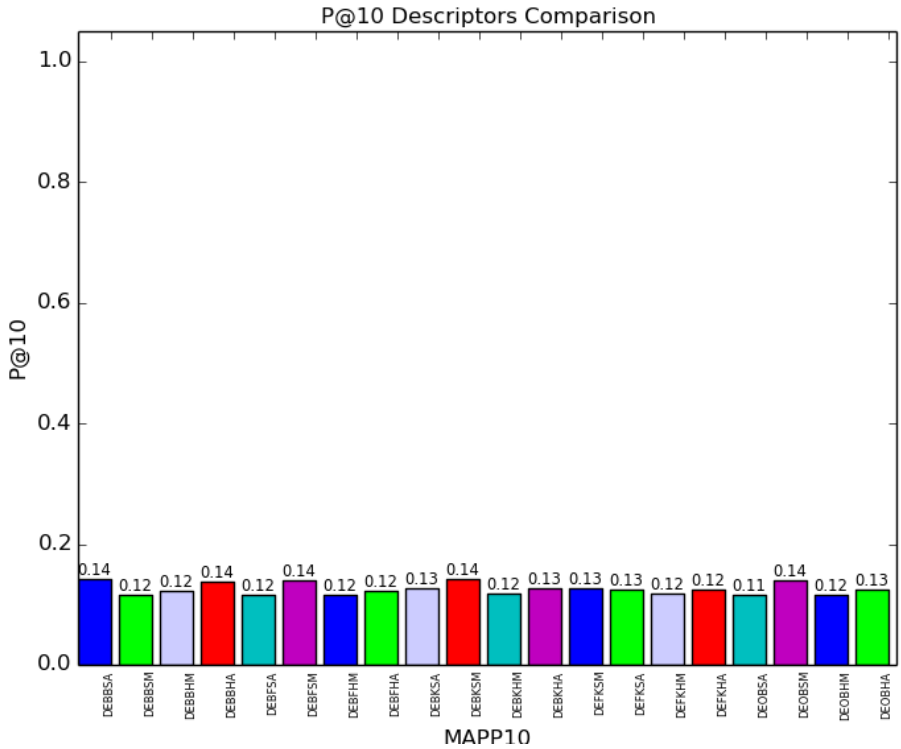


Figure D.15: Overall P@10 to caltech256 dataset (30,607 images) – BoW Descriptors.

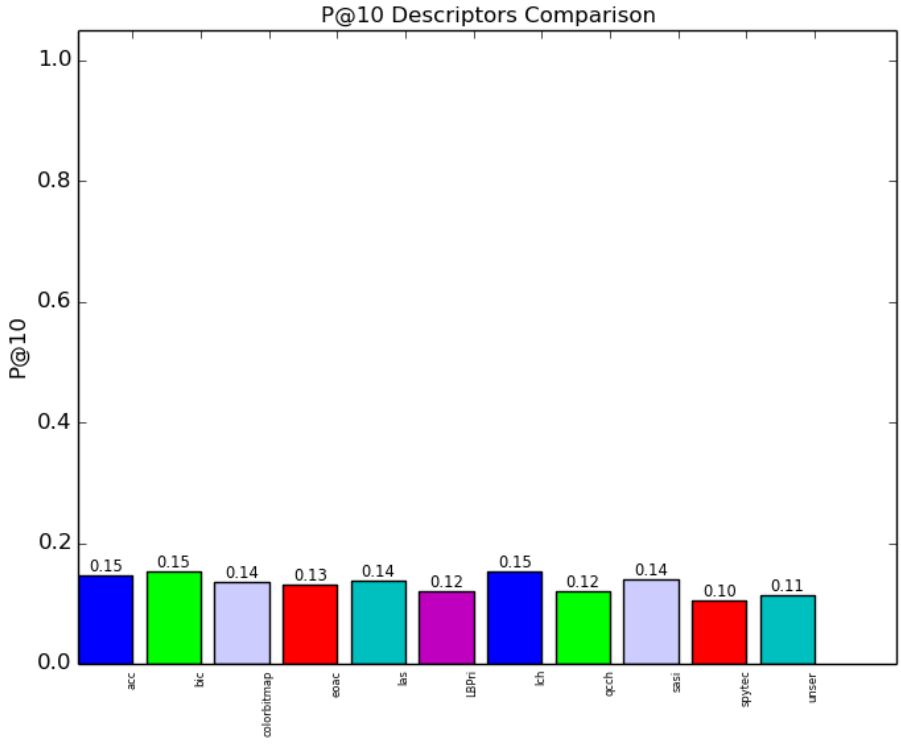


Figure D.16: Overall P@10 to caltech256 (30,607 images) – 11 Best Global Descriptors.

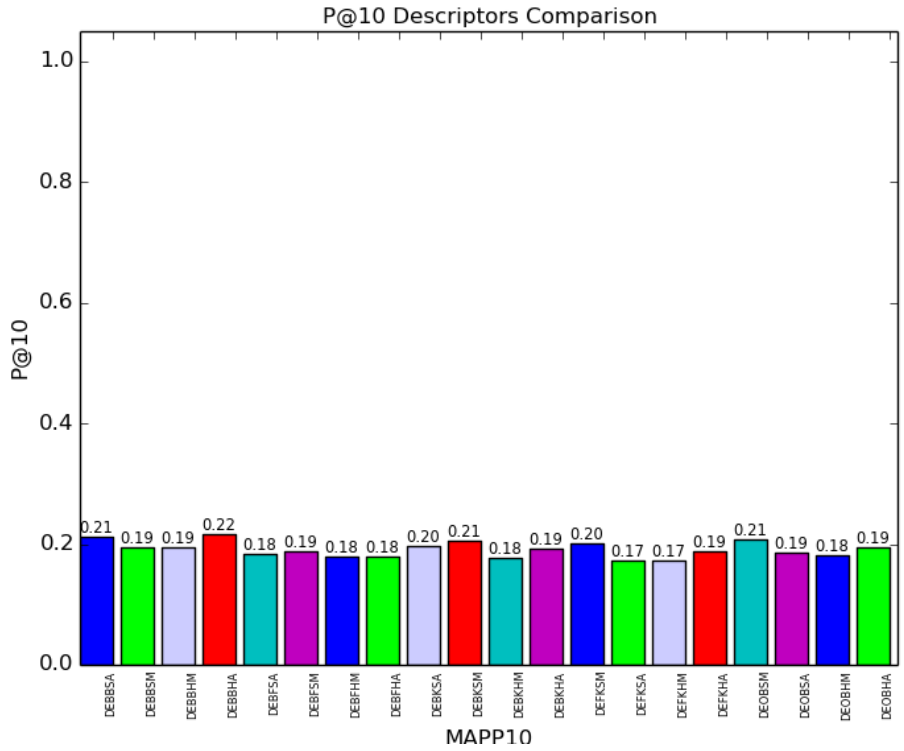


Figure D.17: Overall P@10 to VOC2007 dataset (9,963 images) – BoW Descriptors.

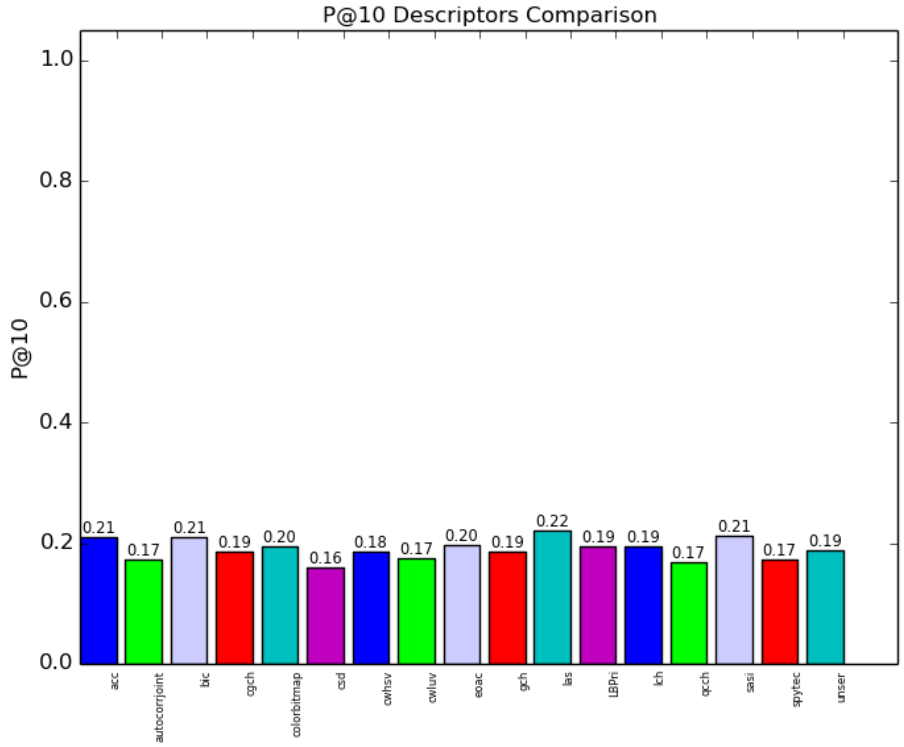


Figure D.18: Overall P@10 to VOC2007 dataset (9,963 images) – Global Descriptors.

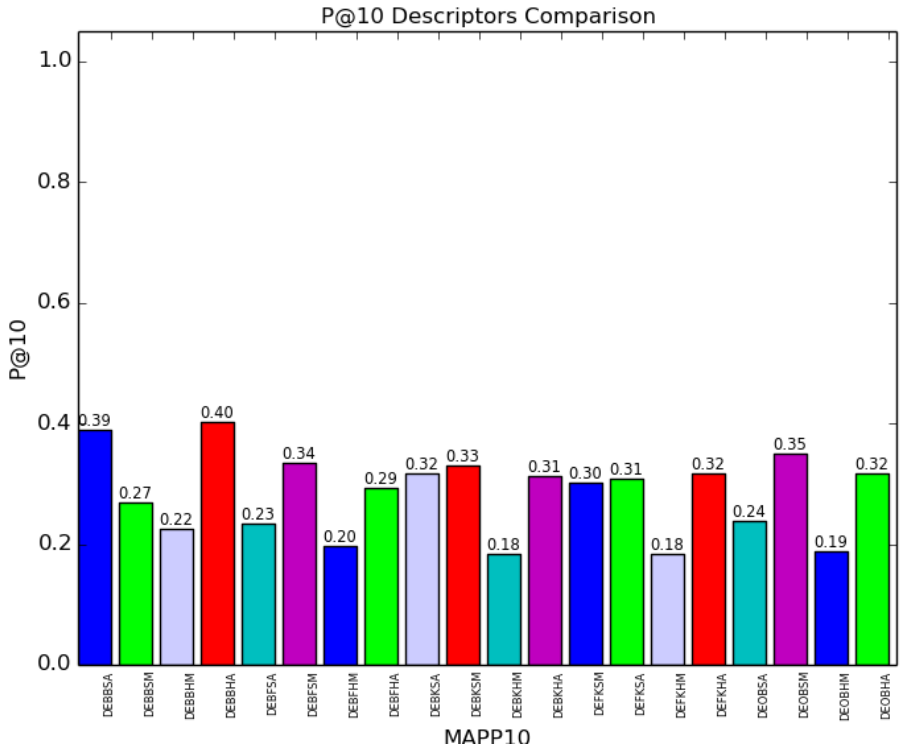


Figure D.19: Overall P@10 to UW dataset (1,009 images) – BoW Descriptors.

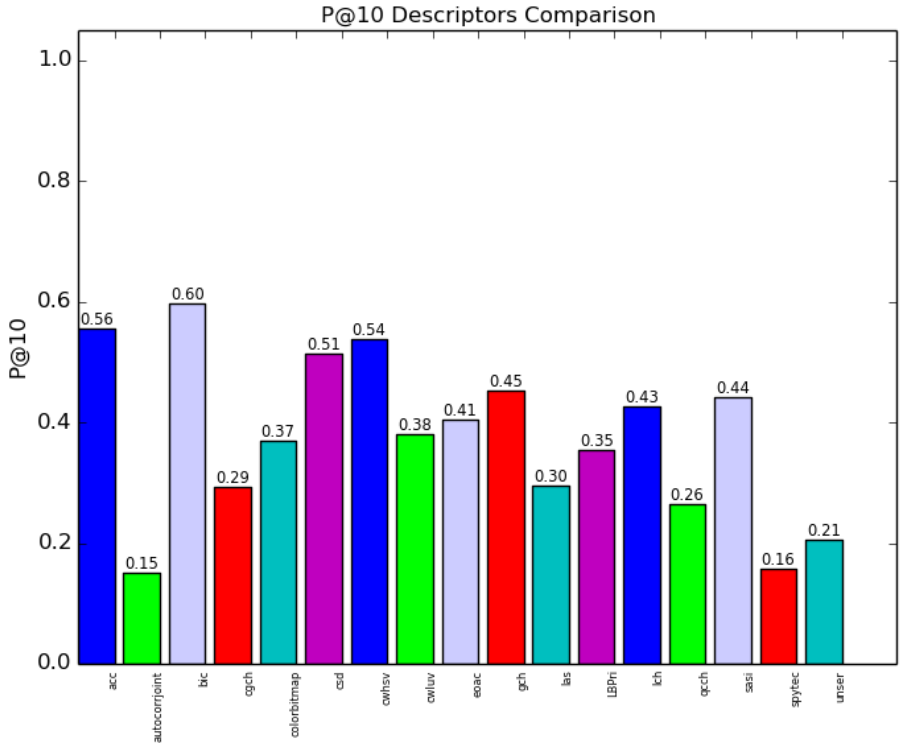


Figure D.20: Overall P@10 to UW dataset (1,009 images) – Global Descriptors.

Appendix E

Five Best Descriptors: P@5/P@10, MAP, Confusion Matrix per Class

1. **15Scenes:** Figure E.1 (BoW Descriptors), Figure E.2 (Global Descriptors).
2. **Paris:** Figure E.3 (BoW Descriptors), Figure E.4 (Global Descriptors).
3. **UW:** Figure E.5 (BoW Descriptors), Figure E.6 (Global Descriptors).
4. **VOC2007:** Figure E.7 (BoW Descriptors), Figure E.8 (Global Descriptors).
5. **WANG:** Figure E.9 (BoW Descriptors), Figure E.10 (Global Descriptors).

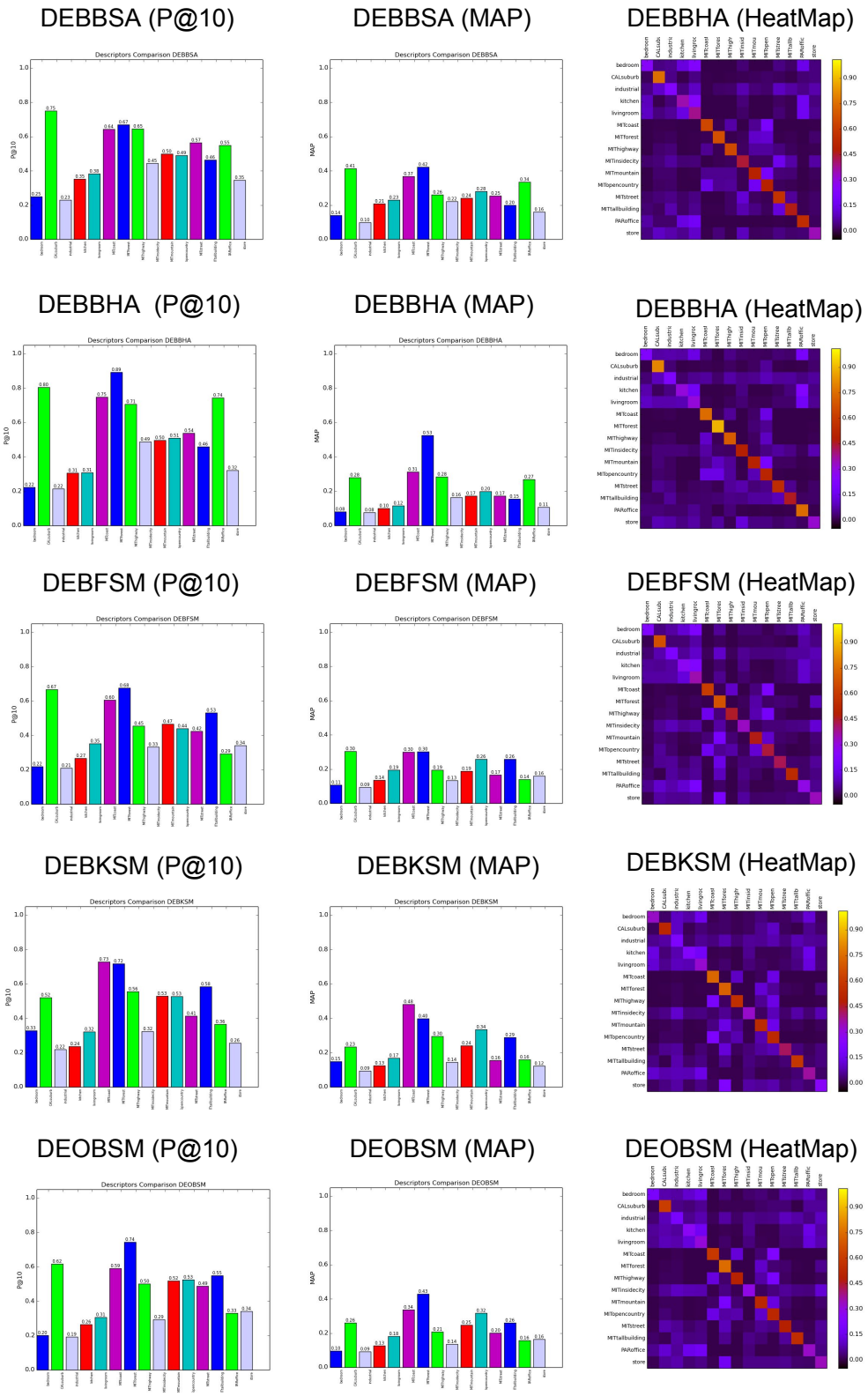


Figure E.1: Bag of Words Descriptors: P@10, MAP and HeatMap of the five best descriptors on 15Scenes Dataset.

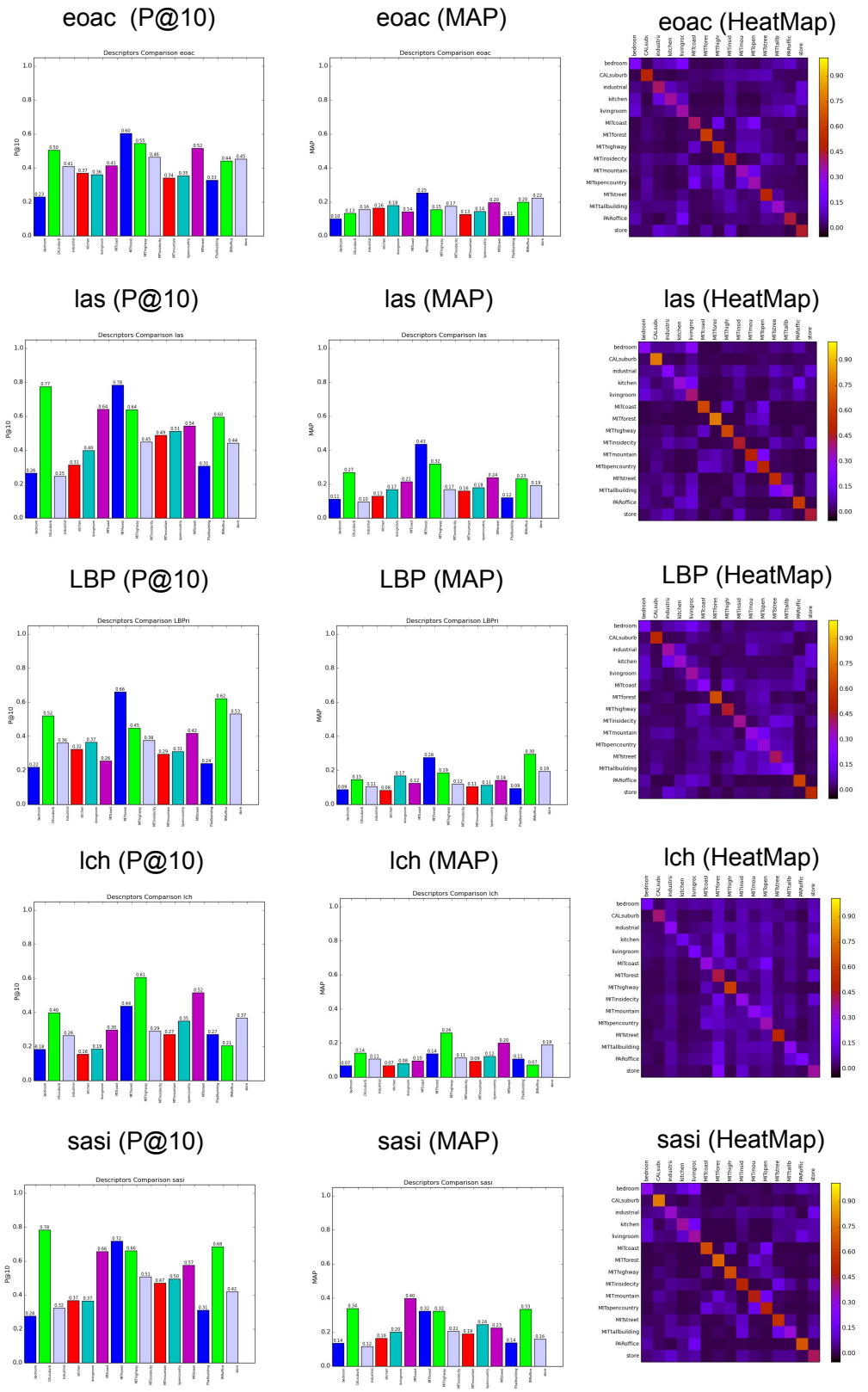


Figure E.2: Global Descriptors: P@10, MAP and HeatMap of the five best descriptors on 15Scenes Dataset.

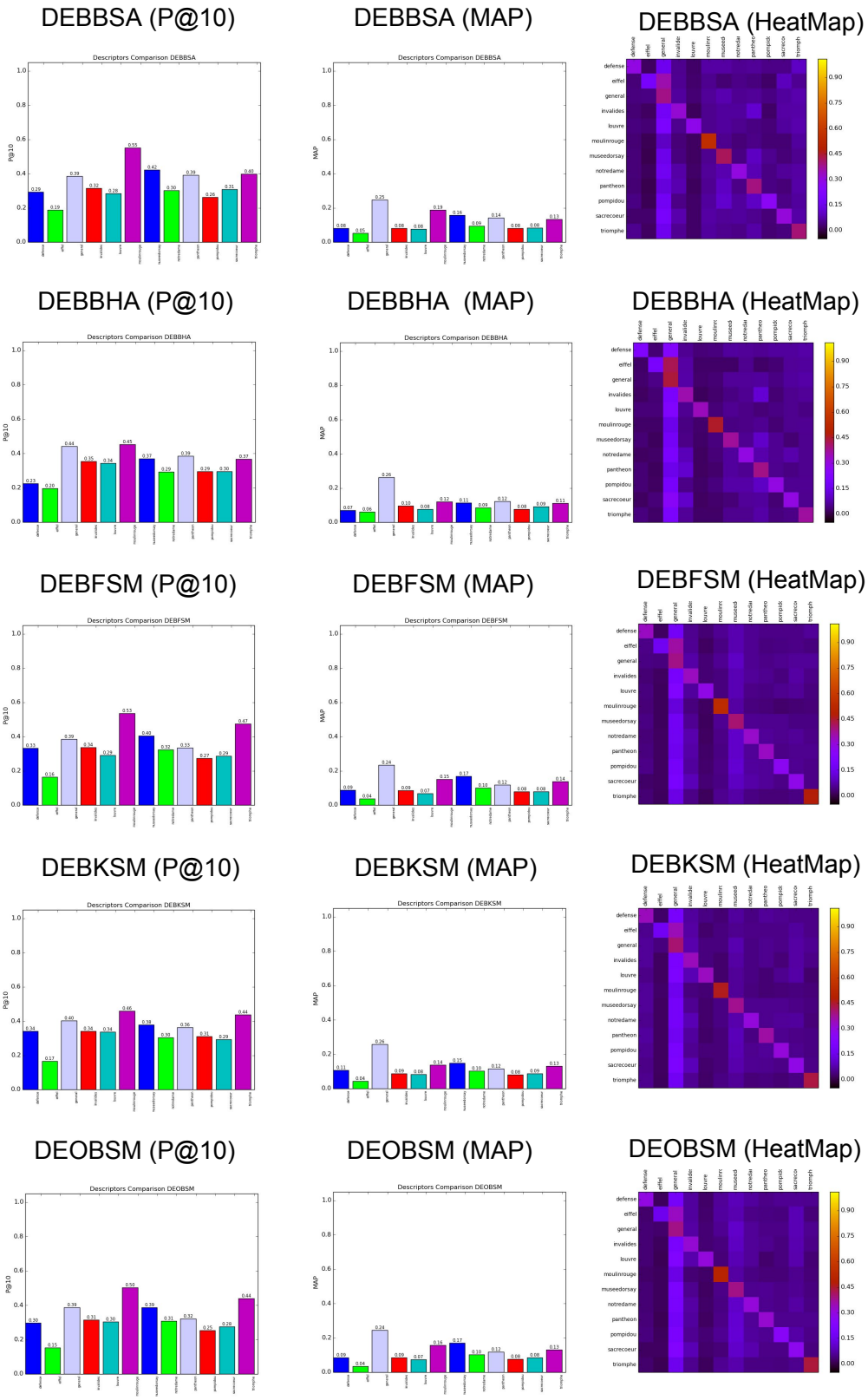


Figure E.3: Bag of Words Descriptors: P@10, MAP and HeatMap of the five best descriptors on Paris Dataset.

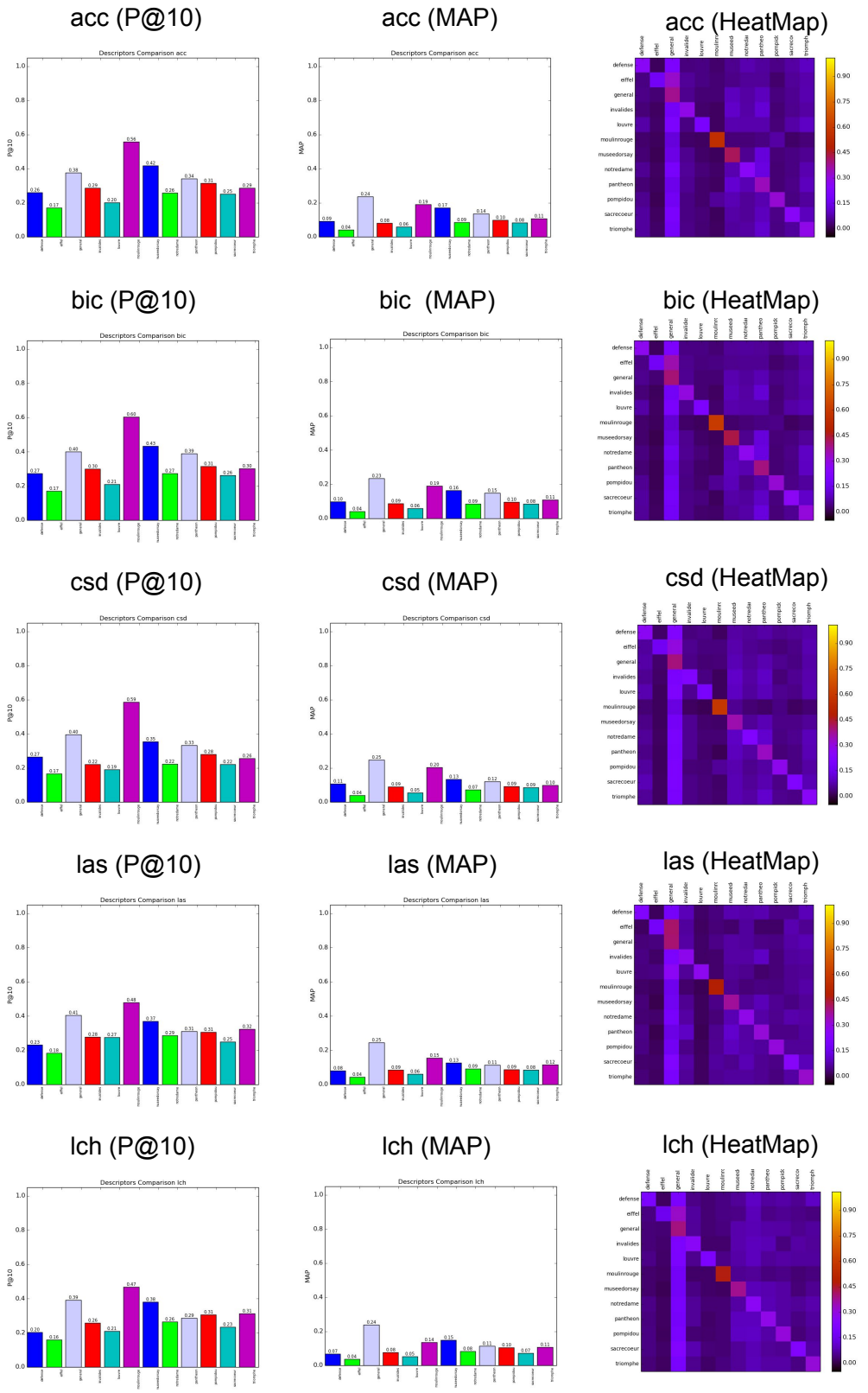


Figure E.4: Global Descriptors: P@10, MAP and HeatMap of the five best descriptors on Paris Dataset.

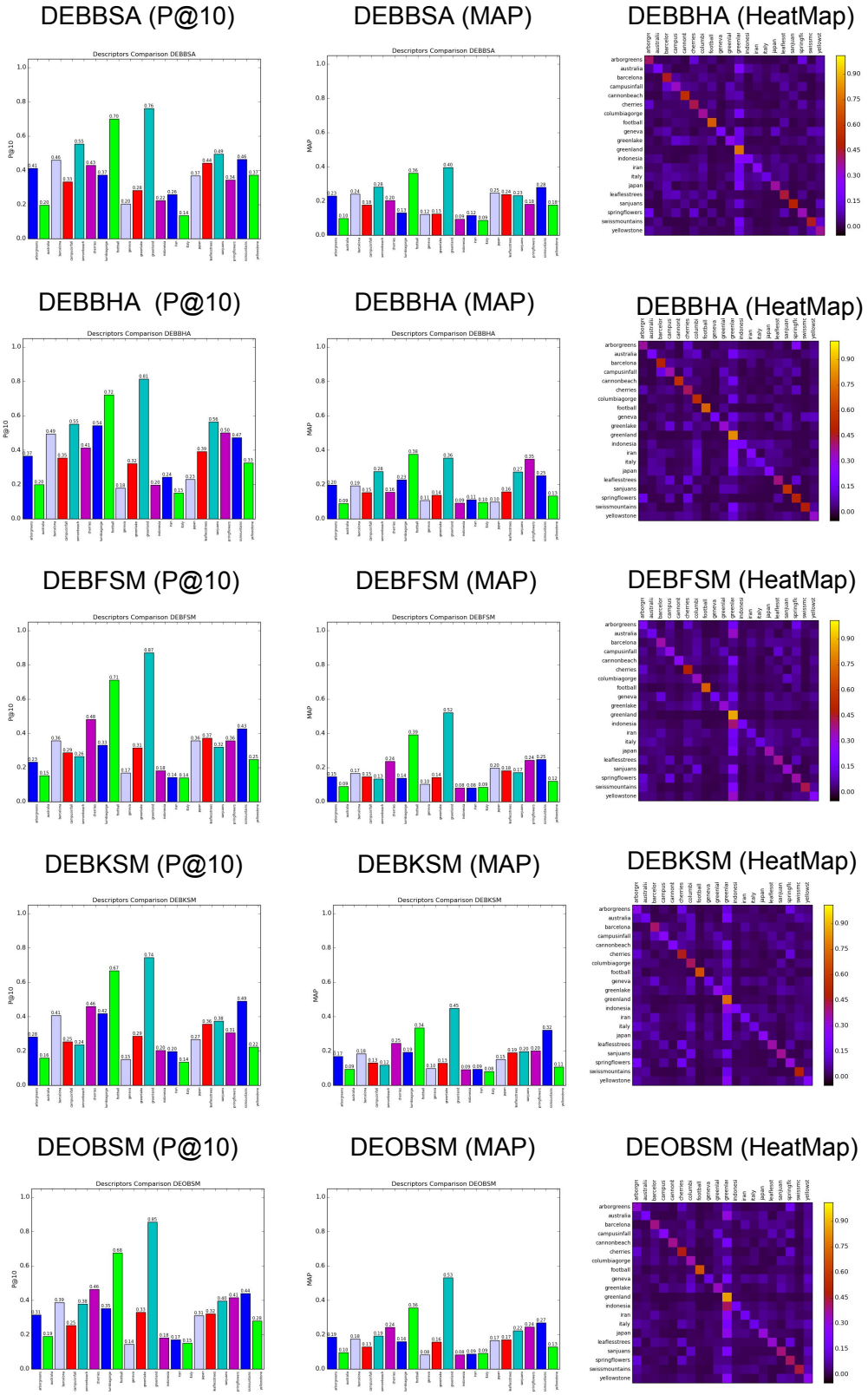


Figure E.5: Bag of Words Descriptors: P@10, MAP and HeatMap of the five best descriptors on UWdataset Dataset.

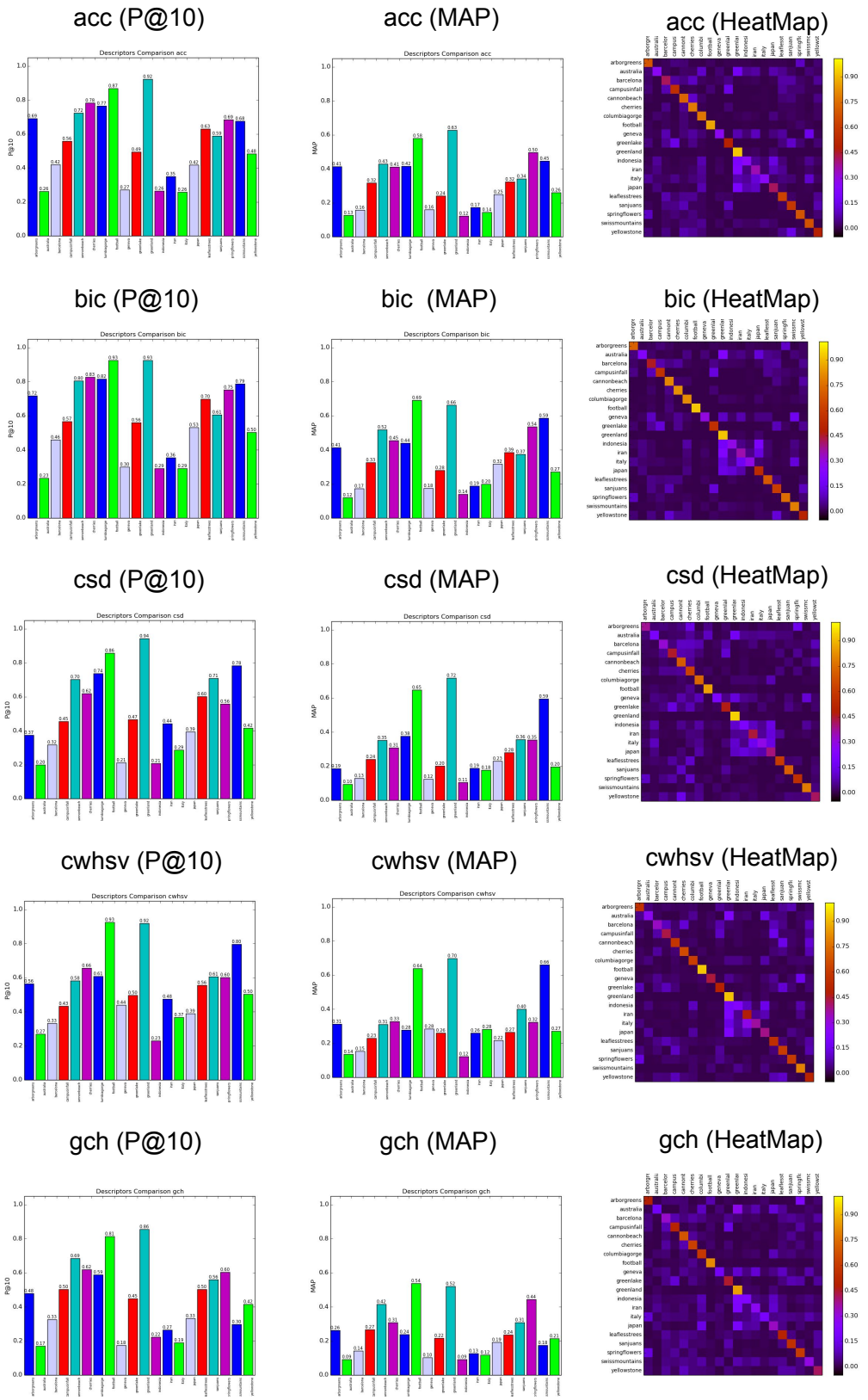


Figure E.6: Global Descriptors: P@10, MAP and HeatMap of the five best descriptors on UWdataset Dataset.

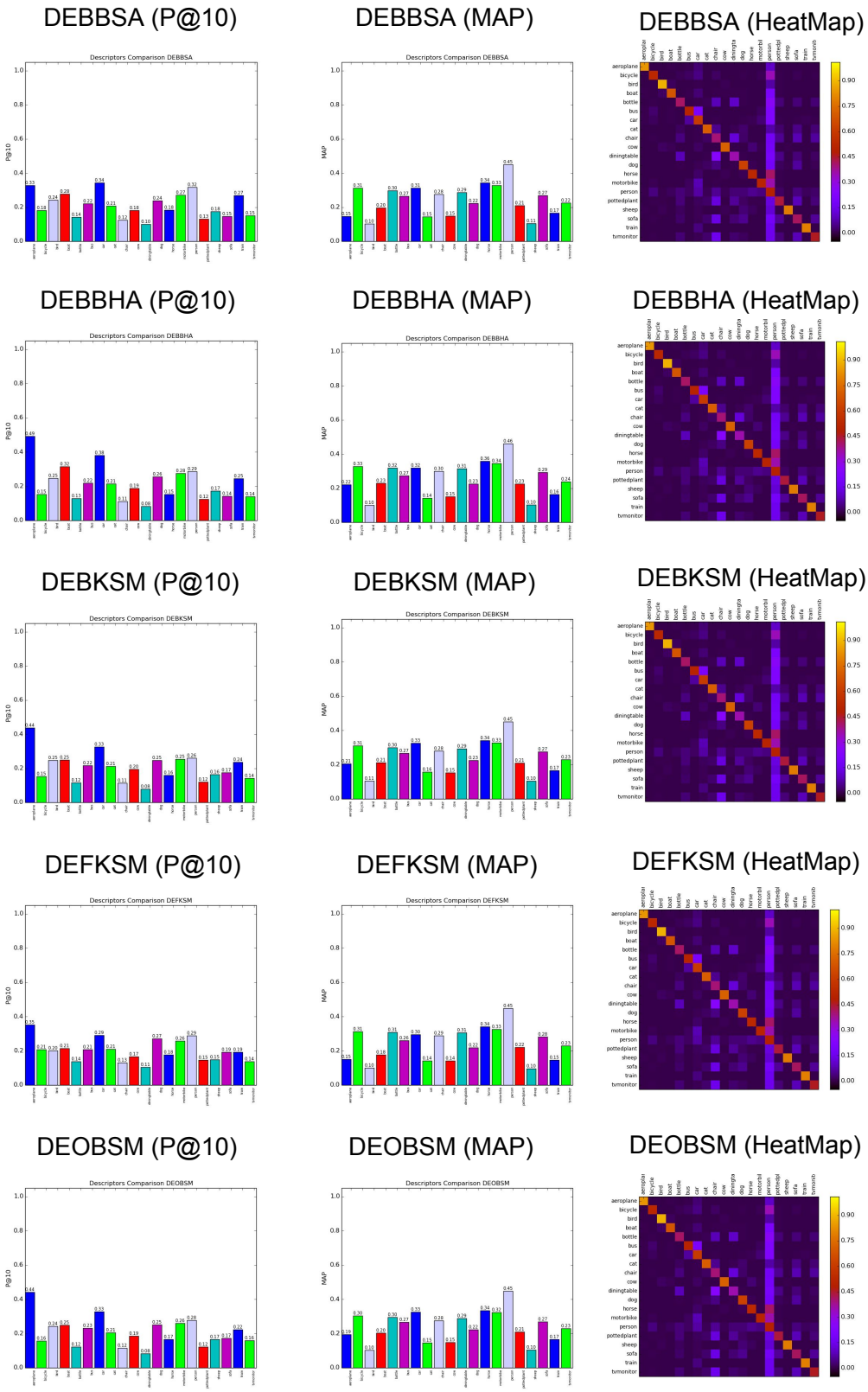


Figure E.7: Bag of Words Descriptors: P@10, MAP and HeatMap of the five best descriptors on VOC2007 Dataset.

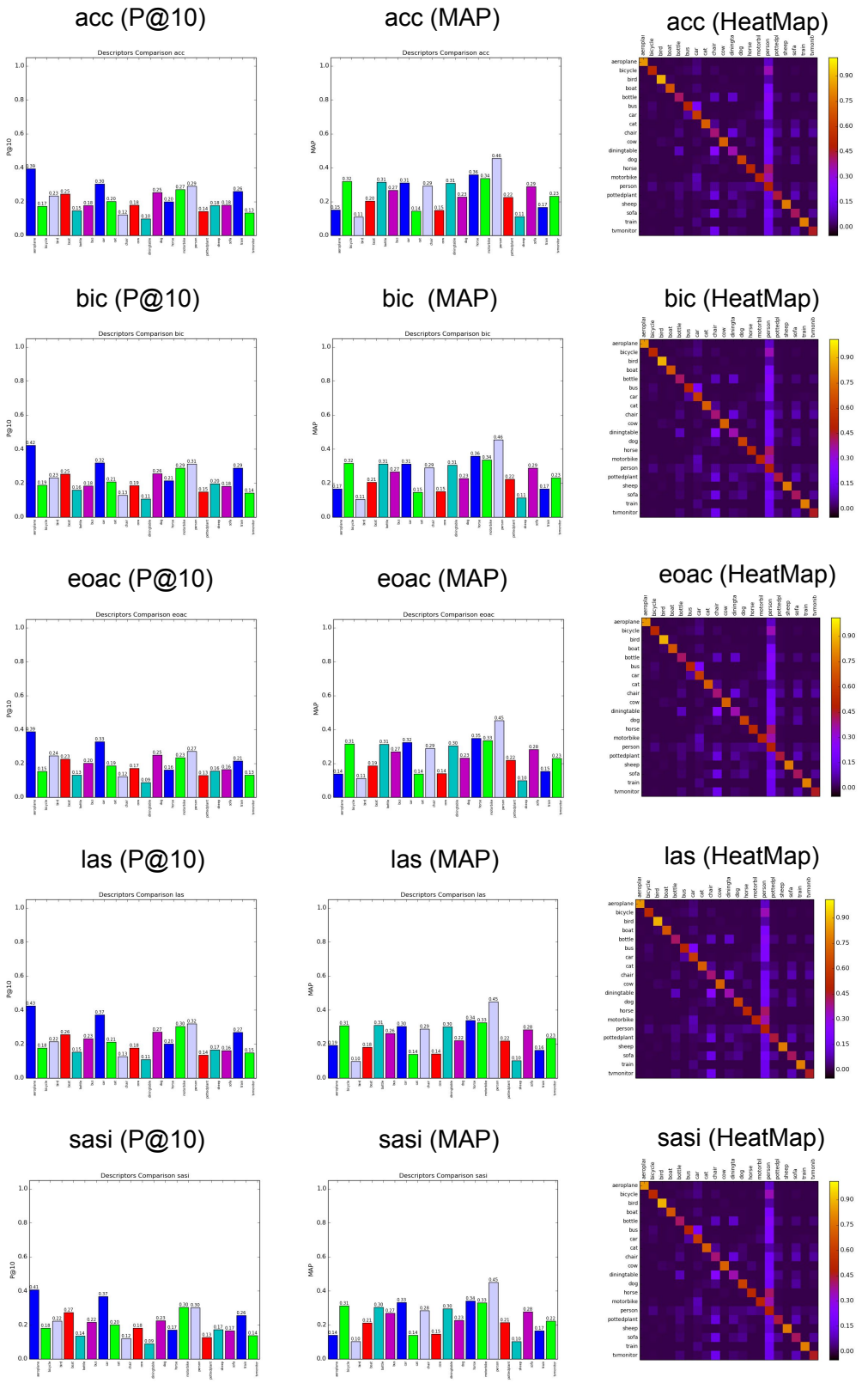


Figure E.8: Global Descriptors: P@10, MAP and HeatMap of the five best descriptors on VOC2007 Dataset.

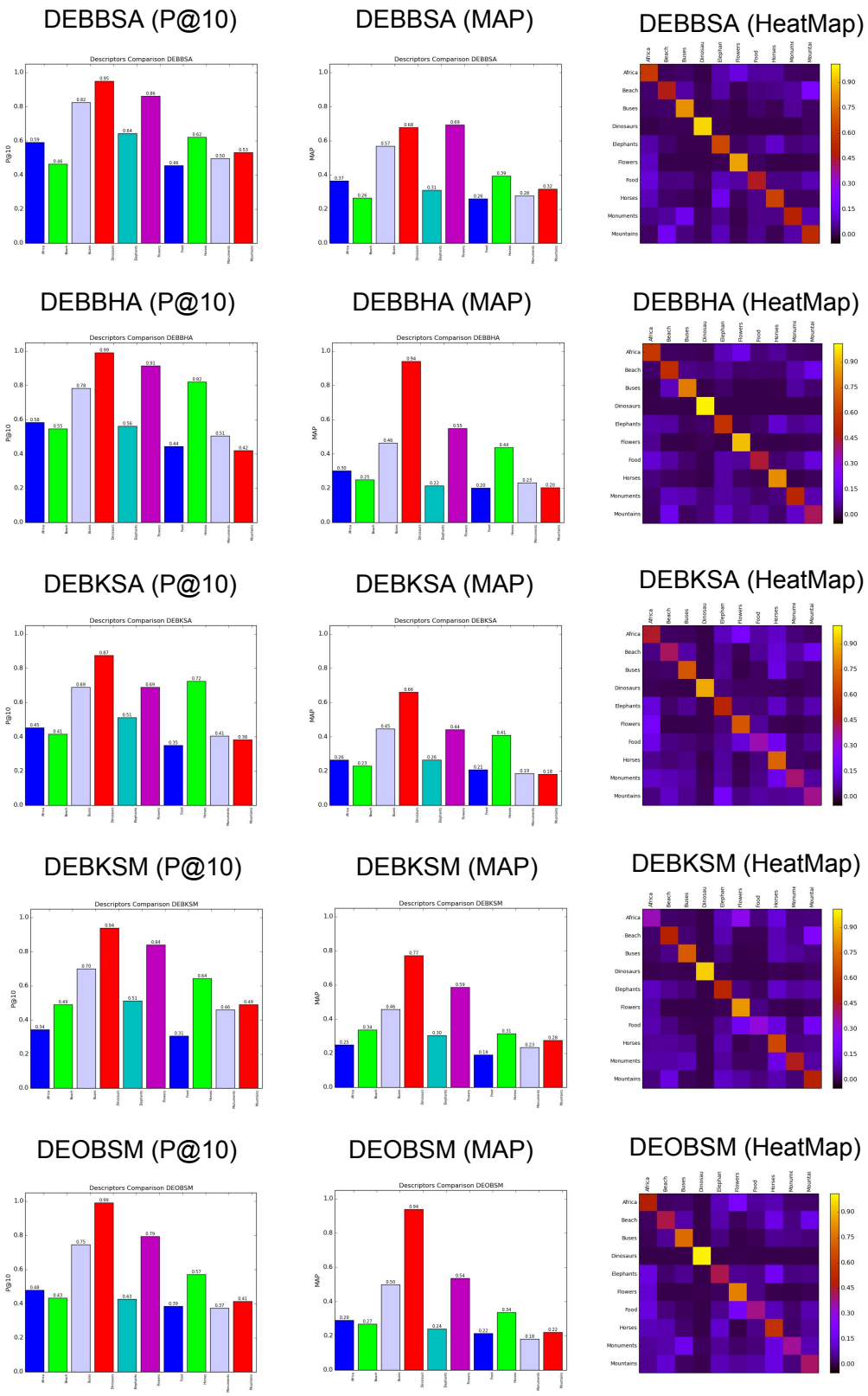


Figure E.9: Bag of Words Descriptors: P@10, MAP and HeatMap of the five best descriptors on WANG Dataset.

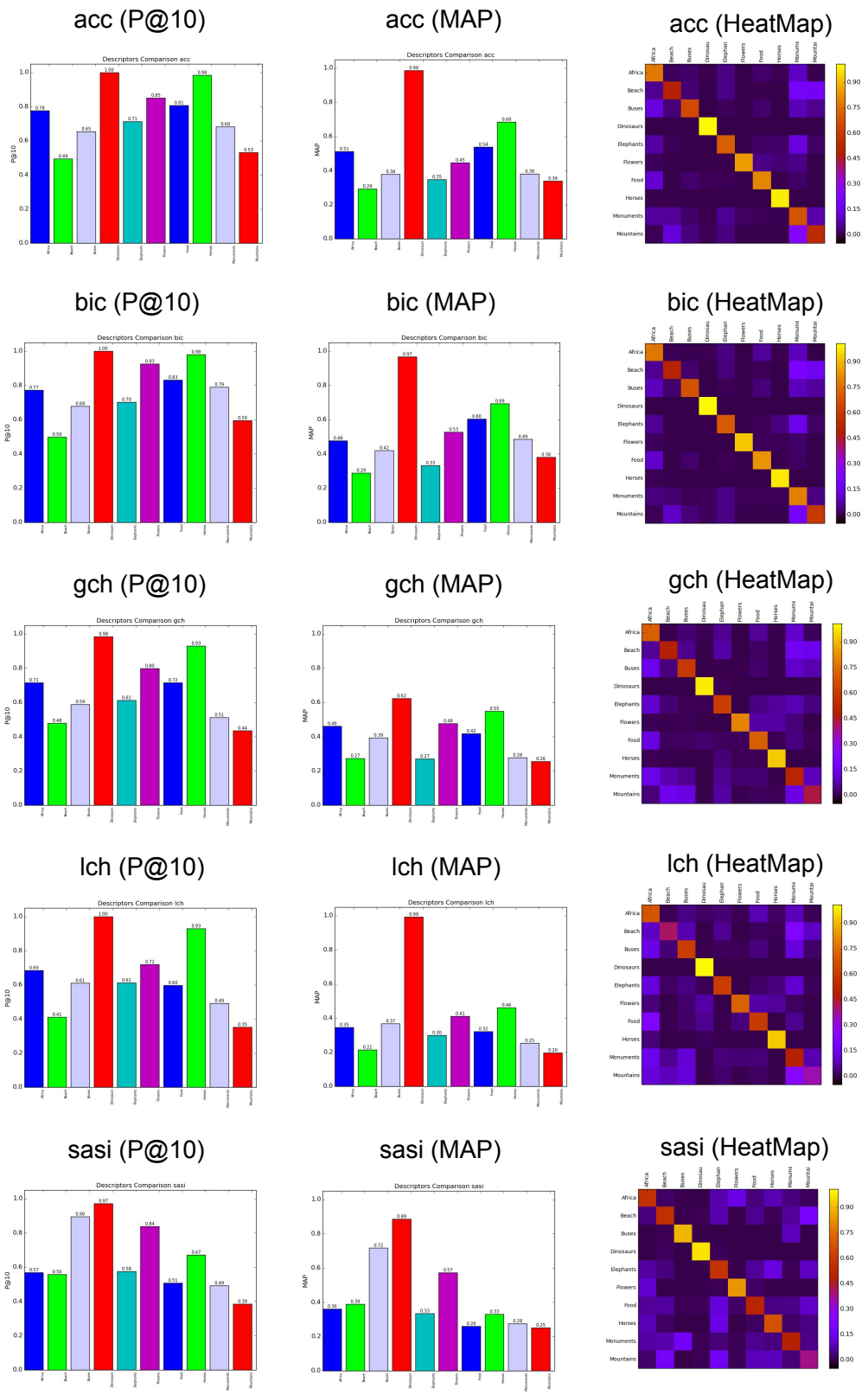


Figure E.10: Global Descriptors: P@10, MAP and HeatMap of the five best descriptors on WANG Dataset.

Appendix F

Image Retrieval Web Interface

In this thesis, we developed an Image Retrieval system to smartphone using Android Development that is not available, but a Web Interface is available on `goldenretriever.dcc.ufmg.br` (Figure F.1). The VPRRetriever (or GoldenRetriever) seeks to do a comparative study of different features and distance metrics in order to analyze the impact of these factors in the process of Content-Base Image Retrieval. Examples of image retrieval using BIC descriptor on the VPRRetriever System are shown in Figure F.2.

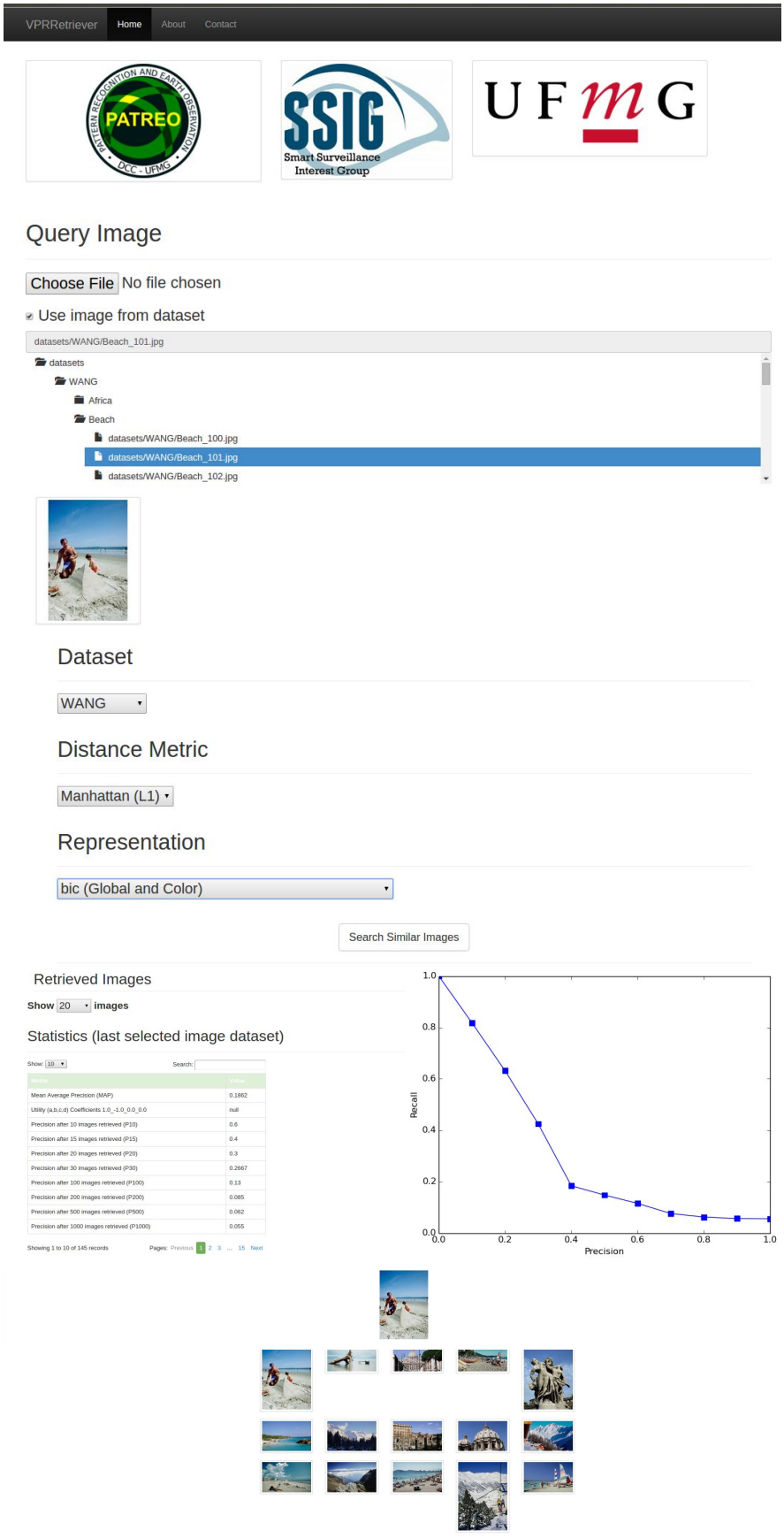


Figure F.1: VPR Retriever System: `goldenretriever.dcc.ufmg.br`

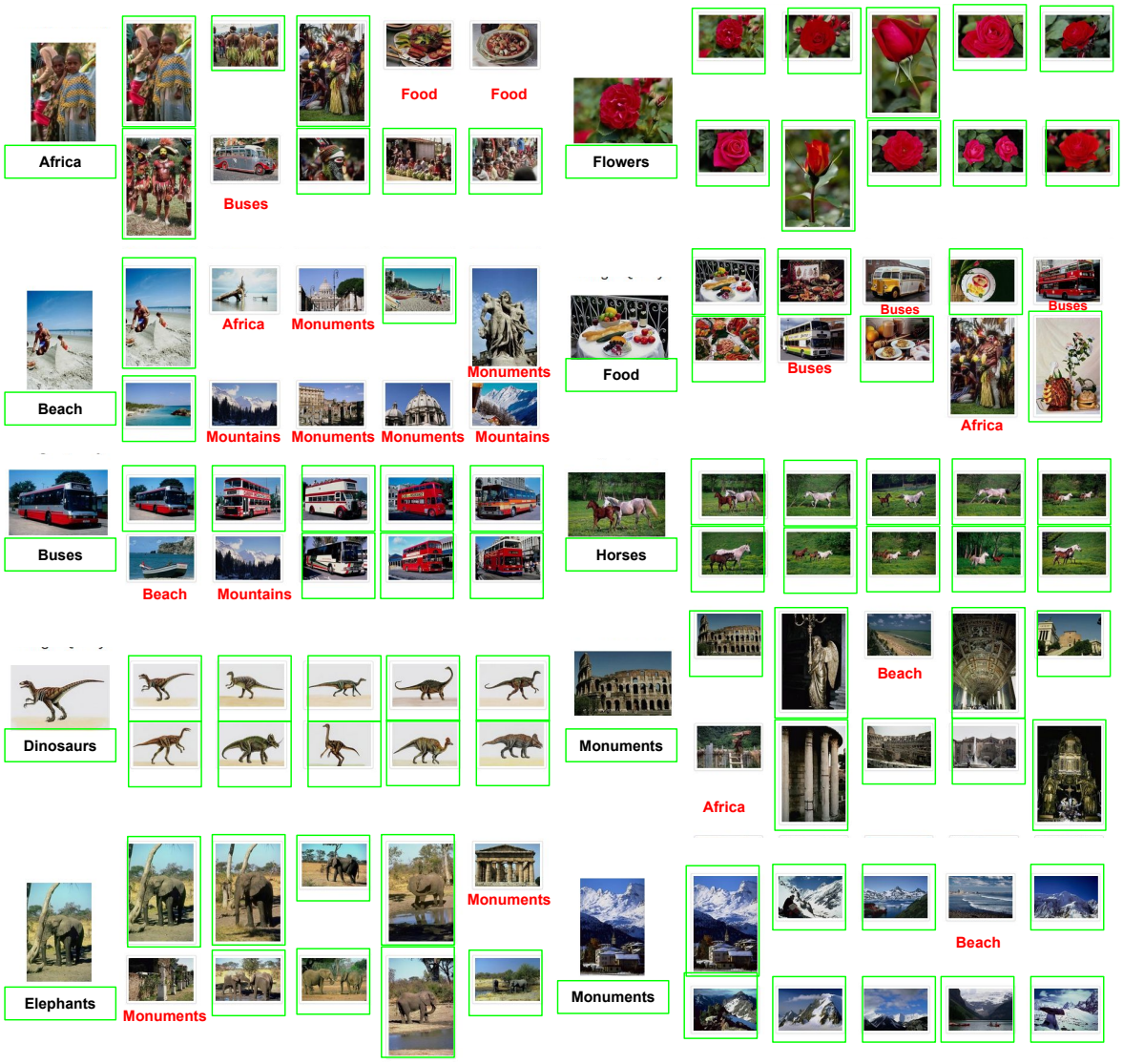


Figure F.2: Examples of images retrieval using BIC descriptor on the VPR Retriever System.

Bibliography

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Susstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274--2282. 25, 76

Ascenso, J. and Pereira, F. (2013). Lossless compression of binary image descriptors for visual sensor networks. In *18th International Conference on Digital Signal Processing (DSP), 2013*, pages 1–8. 3, 45, 46

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)*, 117(5):453--465. 16, 72, 73, 75, 78, 79, 80, 82, 86

Baeza-Yates, R. and Ribeiro-Neto, B. (2011). *Modern Information Retrieval: The Concepts and Technology behind Search (2nd Edition) (ACM Press Books)*. Addison-Wesley Professional, 2 edition. 87

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). volume 110, pages 346--359. Elsevier Science Inc., New York, NY, USA. 3, 6, 9, 19, 46

Beaudet, P. R. (1978). Rotationally invariant image operators. In *Proceedings of the 4th International Joint Conference on Pattern Recognition*, pages 579--583, Kyoto, Japan. 19

Boureau, Y.-L., Bach, F., LeCun, Y., and Ponce, J. (2010). Learning mid-level features for recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 2559–2566. 16

Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal of Software Tools*. 48

Bradski, G. and Kaehler, A. (2013). *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. O'Reilly Media, Inc., 2nd edition. 4

Caetano, C., Avila, S., Guimarães, S., and Araújo, A. d. A. (2014a). Representing local binary descriptors with bossanova for visual recognition. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pages 49--54. ACM. 48, 79, 80, 85, 91

Caetano, C., Avila, S., Guimaraes, S., and de A Araujo, A. (2014b). Pornography detection using bossanova video descriptor. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO), 2014*, pages 1681–1685. 48

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: Binary robust independent elementary features. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pages 778--792. Springer-Verlag, Berlin, Heidelberg. 10, 19, 48

Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. (2010). Spatial-bag-of-features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 3352--3359. IEEE. 72

Carkacioglu, A. (2001). Sasi: A new texture descriptor for content based image retrieval. In *International Conference on Image Processing, 2001*, volume 2, pages 137--140. IEEE. 15, 60

Çarkacıoglu, A. and Yarman-Vural, F. (2003). Sasi: A generic texture descriptor for image retrieval. *Pattern Recognition*, 36(11):2615--2633. 15, 60

Chandrasekhar, V., Reznik, Y., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., and Girod, B. (2010a). Quantization schemes for low bitrate compressed histogram of gradients descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010*, pages 33--40. IEEE. 46

Chandrasekhar, V., Reznik, Y., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., and Girod, B. (2010b). Study of quantization schemes for low bitrate chog descriptors. In *Proceedings of IEEE international workshop on mobile vision (IWMV)*. 3

Chandrasekhar, V., Takacs, G., Chen, D., Tsai, S., Grzeszczuk, R., and Girod, B. (2009). Chog: Compressed histogram of gradients a low bit-rate feature descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009*, pages 2504--2511. IEEE. 3, 45, 46

Chandrasekhar, V. R., Chen, D. M., Tsai, S. S., Cheung, N.-M., Chen, H., Takacs, G., Reznik, Y., Vedantham, R., Grzeszczuk, R., Bach, J., and Girod, B. (2011). The

stanford mobile visual search data set. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems*, pages 117--122, New York, NY, USA. ACM. xix, xxv, 35, 36, 37, 48

Chatzilari, E., Liaros, G., Nikolopoulos, S., and Kompatsiaris, Y. (2013). A comparative study on mobile visual recognition. In *Proceedings of the 9th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 442--457. Springer-Verlag, Berlin, Heidelberg. 1, 45, 46

Chen, D., Tsai, S., Chandrasekhar, V., Takacs, G., Vedantham, R., Grzeszczuk, R., and Girod, B. (2013). Residual enhanced visual vector as a compact signature for mobile visual search. *Signal Processing*, 93(8):2316--2327. 1, 5, 45, 46

Chen, D. M., Tsai, S. S., Chandrasekhar, V., Takacs, G., Vedantham, R., Grzeszczuk, R., and Girod, B. (2010). Inverted index compression for scalable image matching. In *Proceedings of the 2010 Data Compression Conference*, page 525, Washington, DC, USA. IEEE Computer Society. 3

Cisco (2015). Cisco visual networking index: Global mobile data traffic forecast update. Technical report. 1

Csurka, G., Dance, C., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, volume 1, pages 1--2. 3, 20

Deselaers, T., Keysers, D., and Ney, H. (2008). Features for image retrieval: An experimental comparison. *Information Retrieval*, 11(2):77--107. xx, xxv, 41, 42, 43, 48

Dos Santos, J. A., Ferreira, C. D., and da Silva Torres, R. (2008). A genetic programming approach for relevance feedback in region-based image retrieval systems. In *Proceedings of the 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*, volume 8, pages 155--162, Washington, DC, USA. IEEE Computer Society. 2

dos Santos, J. A., Penatti, O. A. B., and da Silva Torres, R. (2010). Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. In *Proceedings of the Fifth International Conference on Computer Vision Theory and Applications (VISAPP), 2010*, volume 2, pages 203--208, Angers, France. 1

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303--338. xx, xxv, 39, 41, 42, 48

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. xx, 38, 39, 48

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167--181. 25

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381--395. 3

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119--139. 11

Girod, B., Chandrasekhar, V., Chen, D. M., Cheung, N.-M., Grzeszczuk, R., Reznik, Y., Takacs, G., Tsai, S. S., and Vedantham, R. (2011). Mobile visual search. *Signal Processing Magazine, IEEE*, 28(4):61--76. 1, 2, 3, 5, 45

Gonzalez, R. C. and Woods, R. E. (2002). Digital image processing. 24

Griffin, G., Holub, A., and Perona, P. (2007). Caltech-256 object category dataset. *Caltech Technical Report*. xx, 38, 40, 48

Hoàng, N. V., Gouet-Brunet, V., Rukoz, M., and Manouvrier, M. (2010). Embedding spatial information into image content description for scene retrieval. *Pattern Recognition*, 43(9):3013--3024. 81, 91

Hong, Y.-J., Kumar, K., and Lu, Y.-H. (2009). Energy efficient content-based image retrieval for mobile systems. In *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, pages 1673--1676. IEEE. 5

Huang, C.-B. and Liu, Q. (2007). An orientation independent texture descriptor for image retrieval. In *International Conference on Communications, Circuits And Systems (ICCCAS), 2007*, pages 772--776. IEEE. 15

Huang, J., Kumar, S. R., Mitra, M., Zhu, W.-J., and Zabih, R. (1997). Image indexing using color correlograms. In *Conference on Computer Vision and Pattern Recognition (CVPR), 1997*, pages 762--768, Washington, DC, USA. IEEE, IEEE Computer Society. 12

Huffman, D. A. et al. (1952). A method for the construction of minimum redundancy codes. *Proceedings of the IRE*, 40(9):1098--1101. 23

Jain, A. K. and Farrokhnia, F. (1990). Unsupervised texture segmentation using gabor filters. In *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, volume 24, pages 1167--1186, New York, NY, USA. Elsevier Science Inc. 15

Jégou, H., Douze, M., and Schmid, C. (2010a). Improving bag-of-features for large scale image search. *International Journal of Computer Vision (IJCV)*, 87(3):316--336. 72

Jégou, H., Douze, M., Schmid, C., and Pérez, P. (2010b). Aggregating local descriptors into a compact image representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 3304--3311. IEEE. 16

Kang, H., Hebert, M., and Kanade, T. (2011). Image matching with distinctive visual vocabulary. In *IEEE Workshop on Applications of Computer Vision (WACV), 2011*, pages 402--409. IEEE. 81, 91

Ken Chatfield, Victor Lempitsky, A. V. and Zisserman, A. (2011). The devil is in the details: An evaluation of recent feature encoding methods. In *Proceedings of the British Machine Vision Conference*, pages 76.1--76.12. BMVA Press. 47, 48, 85, 91

Kumar, K. and Lu, Y.-H. (2010). Cloud computing for mobile users: Can offloading computation save energy? *Computer*, 43(4):51--56. 2

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2006*, volume 2, pages 2169--2178. IEEE. xix, xxv, 3, 32, 33, 48, 72

Lee, D.-H. and Kim, H.-J. (2001). A fast content-based indexing and retrieval technique by the shape information in large image database. *Journal of Systems and Software*, 56(2):165--182. 15

Leutenegger, S., Chli, M., and Siegwart, R. Y. (2011). Brisk: Binary robust invariant scalable keypoints. In *International Conference on Computer Vision (ICCV), 2011 Conference on*, pages 2548--2555. IEEE. 4, 10, 48

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV), 2004*, 60(2):91--110. 3, 6, 9

Lu, T.-C. and Chang, C.-C. (2007). Color image retrieval technique based on color features and image bitmap. *Information Processing and Management: an International Journal*, 43(2):461--472. 13

Mahmoudi, F., Shanbehzadeh, J., Eftekhari-Moghadam, A.-M., and Soltanian-Zadeh, H. (2003). Image retrieval based on shape similarity by edge orientation autocorrelogram. *Pattern recognition*, 36(8):1725--1736. 15

Manjunath, B. S., Ohm, J.-R., Vasudevan, V. V., and Yamada, A. (2001). Color and texture descriptors. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):703--715. 13

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide-baseline stereo from maximally stable extremal regions. pages 36.1–36.10. doi:10.5244/C.16.36. 19

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615--1630. 4

Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. (2005). A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1-2):43--72. 18, 78

Monteiro, P. and Ascenso, J. (2014). Coding mode decision algorithm for binary descriptor coding. In *Proceedings of the 22nd European Signal Processing Conference (EUSIPCO), 2014*, pages 541--545. IEEE. 4

Nallaperumal, K., Banu, M. S., and Christiyana, C. C. (2007). Content based image indexing and retrieval using color descriptor in wavelet domain. In *International Conference on Computational Intelligence and Multimedia Applications, 2007*, volume 3, pages 185--189. IEEE. 13

Nister, D. and Stewenius, H. (2006). Scalable recognition with a vocabulary tree. In *Conference on Computer Vision and Pattern Recognition (CVPR), 2006*, volume 2, pages 2161--2168. IEEE. 4

Ojala, T., Pietikäinen, M., and Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51--59. 14

Ojala, T., Pietikäinen, M., and Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971--987. 14

Ortiz, R. (2012). Freak: Fast retina keypoint. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 510--517, Washington, DC, USA. IEEE Computer Society. 4, 11, 48

Pedrini, H. and Schwartz, W. R. (2008). *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações*. Thomson Learning. 23, 24

Penatti, O. and da Silva Torres, R. (2008). Color descriptors for web image retrieval: A comparative study. In *XXI Brazilian Symposium on Computer Graphics and Image Processing, 2008*, pages 163--170. IEEE. 1, 60

Penatti, O. A., Valle, E., and Torres, R. d. S. (2011a). Encoding spatial arrangement of visual words. In San Martin, C. and Kim, S.-W., editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7042 of *Lecture Notes in Computer Science*, pages 240--247. Springer. 9, 72, 73, 74, 78

Penatti, O. A., Valle, E., and Torres, R. d. S. (2011b). Encoding spatial arrangement of visual words. In San Martin, C. and Kim, S.-W., editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7042 of *Lecture Notes in Computer Science*, pages 240--247. Springer. 21, 22

Penatti, O. A., Valle, E., and Torres, R. d. S. (2012). Comparative study of global color and texture descriptors for web image retrieval. *Journal of Visual Communication and Image Representation (JVCIR)*, 23(2):359--380. 47

Penatti, O. A. B., Silva, F. B., Valle, E., Gouet-Brunet, V., and da S. Torres, R. (2014). Visual word spatial arrangement for image retrieval and classification. *Pattern Recognition*, 47(2):705–720. 16, 20, 47, 48, 71, 81, 86

Perronnin, F., Sánchez, J., and Mensink, T. (2010). Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision (ECCV)*, pages 143--156. Springer-Verlag, Berlin, Heidelberg. 16

Pessoa, R. F., Schwartz, W. R., and dos Santos, J. A. (2015a). A study on low-cost representations for image feature extraction on mobile devices. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - 20th Iberoamerican Congress, CIARP 2015, Montevideo, Uruguay, November 9-12, 2015, Proceedings*, volume 9423 of *Lecture Notes in Computer Science*, pages 424--431. Springer International Publishing. xx, xxi, 7, 53, 54, 57, 59, 60, 85, 89, 91

Pessoa, R. F., Schwartz, W. R., and Santos, J. A. d. (2015b). An experimental comparison of feature extraction and distance metrics for image retrieval. In *XXVIII Conference on Graphics, Patterns and Images (SIBGRAPI). Salvador, Brazil, August 26-29, 2015*. 7

Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A. (2007). Object retrieval with large vocabularies and fast spatial matching. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007*, pages 1--8. IEEE. xix, xxv, 4, 32, 33, 34, 48

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008*, pages 1--8. IEEE. xix, xxv, 4, 34, 35, 48

Redondi, A., Baroffio, L., Ascenso, J., Cesana, M., and Tagliasacchi, M. (2013). Rate-accuracy optimization of binary descriptors. In *20th IEEE International Conference on Image Processing (ICIP), 2013*, pages 2910--2914. 4

Rosten, E. and Drummond, T. (2006). Machine learning for high-speed corner detection. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, pages 430--443. Springer-Verlag, Berlin, Heidelberg. 18

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). Orb: An efficient alternative to sift or surf. In *IEEE International Conference on Computer Vision (ICCV), 2011*, pages 2564--2571. IEEE. 4, 10, 19, 48, 78

Shao, T. S. H. and Gool, L. V. (2004). Zubud - zurich buildings database for image based recognition. Technical report 260, Swiss Federal Institute of Technology. xix, 34, 35, 48

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888--905. 24, 25

Siddiqi, K. (2009). Turbo pixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(12). 25

Silva de Moura, E., Navarro, G., Ziviani, N., and Baeza-Yates, R. (2000). Fast and flexible word searching on compressed text. *ACM Transactions on Information Systems*, 18(2):113--139. 23

Sivic, J. and Zisserman, A. (2003). Video google: A text retrieval approach to object matching in videos. In *Ninth IEEE International Conference on Computer Vision (ICCV), 2003*, volume 2, pages 1470--1477. IEEE. 16, 20

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 *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 102--109, New York, NY, USA. ACM, ACM. 12, 61, 63, 76, 84

Stricker, M. and Orengo, M. (1995). Similarity of color images. volume 2420, pages 381--392. 12

Swain, M. J. and Ballard, D. H. (1991). Color indexing. *International journal of computer vision*, 7(1):11--32. 13, 14, 60

Takacs, G., Chandrasekhar, V., Gelfand, N., Xiong, Y., Chen, W.-C., Bismpigiannis, T., Grzeszczuk, R., Pulli, K., and Girod, B. (2008). Outdoors augmented reality on mobile phone using loxel-based visual feature organization. In *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, pages 427--434, New York, NY, USA. ACM, ACM. 5

Tao, B. and Dickinson, B. W. (2000). Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation (JVCIR)*, 11(3):327--342. 14, 60

Tomasi, C. and Shi, J. (1994). Good features to track. *Conference on Computer Vision and Pattern Recognition (CVPR), 1994*, 600:593--593. 19, 78

Traina Jr, C., Traina, A., Faloutsos, C., and Seeger, B. (2002). Fast indexing and visualization of metric data sets using slim-trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(2):244--260. 81, 91

Trajković, M. and Hedley, M. (1998). Fast corner detection. *Image and Vision Computing*, 16(2):75--87. 18

Trzcinski, T., Christoudias, M., Fua, P., and Lepetit, V. (2013). Boosting binary keypoint descriptors. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2874--2881, Washington, DC, USA. IEEE, IEEE Computer Society. 11, 47, 48

Tsai, S. S., Chen, D., Chandrasekhar, V., Takacs, G., Cheung, N.-M., Vedantham, R., Grzeszczuk, R., and Girod, B. (2010). Mobile product recognition. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1587--1590, New York, NY, USA. ACM, ACM. 2, 3

Tsai, S. S., Chen, D., Takacs, G., Chandrasekhar, V., Singh, J. P., and Girod, B. (2009). Location coding for mobile image retrieval. In *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, pages 1--7, ICST, Brussels, Belgium, Belgium. ICST (Institute for Computer Sciences, Social - Informatics and Telecommunications Engineering). 3

Tuytelaars, T. (2010). Dense interest points. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, pages 2281--2288. IEEE. 16

Tuytelaars, T. and Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177--280. 50

Unser, M. (1986). Sum and difference histograms for texture classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):118--125. 15

Utenpattanant, A., Chitsobhuk, O., and Khawne, A. (2006). Color descriptor for image retrieval in wavelet domain. In *The 8th International Conference Advanced Communication Technology (ICACT), 2006*, volume 1, page 821. IEEE. 13

Valenzuela, R. E. G., Pedrini, H., and Schwartz, W. R. (2013). Dimensionality reduction through lda and bag-of-features applied to image retrieval. *Computational Vision and Medical Image Processing IV(VIPIMAGE), 2013*, page 31. 3

van de Sande, K., Gevers, T., and Snoek, C. (2010). Evaluating color descriptors for object and scene recognition. *TPAMI*, 32(9):1582–1596. 48

van Gemert, J. C., Veenman, C. J., Smeulders, A. W., and Geusebroek, J.-M. (2010). Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271--1283. 20

Vedaldi, A. and Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In Forsyth, D., Torr, P., and Zisserman, A., editors, *European Conference on Computer*

Vision (ECCV), 2008, volume 5305 of *Lecture Notes in Computer Science*, pages 705--718. Springer Berlin Heidelberg. 25

Veksler, O., Boykov, Y., and Mehrani, P. (2010). Superpixels and supervoxels in an energy optimization framework. In *Proceedings of the 11th European Conference on Computer Vision (ECCV), 2010*, pages 211--224. Springer-Verlag, Berlin, Heidelberg. 25

Viitaniemi, V. and Laaksonen, J. (2008). *Experiments on Selection of Codebooks for Local Image Feature Histograms*, volume 5188 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 79, 80

Wang, J. Z., Li, J., and Wiederhold, G. (2001). Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947--963. xix, xxv, xxvi, 38, 48, 80

Wang, L. and He, D.-C. (1990). Texture classification using texture spectrum. *Pattern Recognition*, 23(8):905--910. 14

Weber, R., Schek, H.-J., and Blott, S. (1998). A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases (VLDB)*, volume 98, pages 194--205, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 81, 91

Williams, A. and Yoon, P. (2007). Content-based image retrieval using joint correlograms. *Multimedia Tools Applications*, 34(2):239--248. 13

Zhou, W., Lu, Y., Li, H., Song, Y., and Tian, Q. (2010a). Spatial coding for large scale partial-duplicate web image search. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 511--520, New York, NY, USA. ACM. 72, 73

Zhou, W., Yang, M., Li, H., Wang, X., Lin, Y., and Tian, Q. (2014). Towards codebook-free: Scalable cascaded hashing for mobile image search. *IEEE Transactions on Multimedia*, 16(3):601--611. 45, 46

Zhou, X., Yu, K., Zhang, T., and Huang, T. S. (2010b). Image classification using super-vector coding of local image descriptors. In *European Conference on Computer Vision (ECCV), 2010*, pages 141--154. Springer-Verlag, Berlin, Heidelberg. 48, 85, 91

Zhuang, D., Zhang, D., Li, J., and Tian, Q. (2014). Binary feature from intensity quantization and weakly spatial contextual coding for image search. *Information Sciences*, 302:94--107. 4, 46

Zitnick, C. L. and Kang, S. B. (2007). Stereo for image-based rendering using image over-segmentation. *International Journal of Computer Vision (IJCV)*, 75(1):49--65. 25