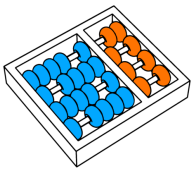
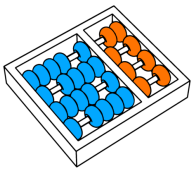


Junior John Fabian Arteaga

"Searching for People through Textual and Visual Attributes"

"*Busca de pessoas a partir de atributos visuais e textuais*"

CAMPINAS
2013



University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Junior John Fabian Arteaga

"Searching for People through Textual and Visual Attributes"

Supervisor:
Orientador(a): **Prof. Dr. Anderson de Rezende Rocha**

"Busca de pessoas a partir de atributos visuais e textuais"

MSc Dissertation presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a Mestre degree in Computer Science.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

This volume corresponds to the final version of the Dissertation defended by Junior John Fabian Arteaga, under the supervision of Prof. Dr. Anderson de Rezende Rocha.

Este exemplar corresponde à versão final da Dissertação defendida por Junior John Fabian Arteaga, sob orientação de Prof. Dr. Anderson de Rezende Rocha.

Anderson de Rezende Rocha

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

F112s

Fabián Arteaga, Junior John, 1987-
Searching for people through textual and visual attributes / Junior John Fabián Arteaga. – Campinas, SP : [s.n.], 2013.

Orientador: Anderson de Rezende Rocha.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Imagens - Recuperação. 2. Recuperação da informação. 3. Reconhecimento de padrões. 4. Análise de imagens. 5. Imagens digitais - Pesquisa. I. Rocha, Anderson de Rezende,1980-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Busca de pessoas a partir de atributos visuais e textuais
Palavras-chave em inglês:
Images - Retrieval
Information retrieval
Pattern recognition
Image analysis
Digital images - Research
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Anderson de Rezende Rocha [Orientador]
Hélio Pedrini
Daniel Carlos Guimarães Pedronette
Data de defesa: 30-09-2013
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 30 de setembro de 2013, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Daniel Carlos Guimarães Pedronette
UNESP / Rio Claro

Prof. Dr. Hélio Pedrini
IC / UNICAMP

Prof. Dr. Anderson de Rezende Rocha
IC / UNICAMP

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

Searching for People through Textual and Visual Attributes

Junior John Fabian Arteaga[1]

September 30, 2013

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Anderson de Rezende Rocha (Supervisor/*Orientador*)
- Prof. Dr. Hélio Pedrini
  Institute of Computing – UNICAMP
- Prof. Dr. Daniel Carlos Guimarães Pedronette
  Department of Statistics, Applied Mathematics and Computation – UNESP
- Prof. Dr. Ricardo Torres
  Institute of Computing – UNICAMP (Alternate Member)
- Prof. Dr. Fábio Augusto Menocci Cappabianco
  Institute of Science and Technology – UNIFESP (Alternate Member)

[1]Financial support: CNPq scholarship 07/2011 – 01/2012, CAPES scholarship 07/2012 – 07/2013

Abstract

Using personal traits for searching people is paramount in several application areas and has attracted an ever-growing attention from the scientific community over the past years. Some practical applications in the realm of digital forensics and surveillance include locating a suspect or finding missing people in a public space. In this work, we aim at assigning describable visual attributes (e.g., *white chubby male* wearing *glasses* and with *bangs*) as labels to images to describe their appearance and performing visual searches without relying on image annotations during testing. For that, we create mid-level image representations for face images based on visual dictionaries linking visual properties in the images to describable attributes. First, we propose one single-level and one multi-level approaches to solve simple queries (queries containing only one attribute). For both methods, the first step consists of obtaining image low-level features either using a sparse or a dense-sampling scheme. The characterization is followed by the visual dictionary creation step in which we assess both a random selection and a clustering algorithm for selecting the most important features collected in the first stage. Such features then feed 2-class classifiers for the describable visual attributes of interest which assign to each image a decision score used to obtain its ranking. As the multi-level image characterization involves combining the answers of different levels, we also propose some fusion methods in this regard. For more complex queries (2+ attributes), we use three state-of-the-art approaches for combining the rankings: product of probabilities, rank aggregation and rank position. We also extend upon the rank aggregation method in order to take advantage of complementary information produced by the different characterization schemes. We have considered fifteen attribute classifiers and, consequently, their direct counterparts theoretically allowing $2^{15} = 32,768$ different combined queries (the actual number is smaller since some attributes are contradictory or mutually exclusive). Experimental results show that the multilevel approach improves retrieval precision for most of the attributes in comparison with other methods. Finally, for combined attributes, the multilevel characterization approach along with the modified rank aggregation scheme boosts the precision performance when compared to other methods such as product of probabilities and rank position.

Resumo

Utilizar características pessoais para procurar pessoas é fundamental em diversas áreas de aplicação e nos últimos anos tem atraído uma atenção crescente por parte da comunidade científica com aplicações no campo da forense digital e vigilância tais como: localização de suspeitos ou de pessoas desaparecidas em espaços públicos. Neste trabalho, objetivamos utilizar atributos visuais descritíveis (por exemplo, *homens brancos com bochechas em destaque* usando *óculos* e com *franja*) como rótulos nas imagens para descrever sua aparência e, dessa forma, realizar buscas visuais por conteúdo sem depender de anotações nas imagens durante os testes. Para isso, criamos representações robustas para imagens de faces baseadas em dicionários visuais, vinculando as propriedades visuais das imagens aos atributos descritíveis. Primeiro, propomos duas abordagens de caracterização das imagens, uma de escala única e outra de múltiplas escalas para resolver consultas simples (somente um atributo). Em ambos os métodos, obtemos as características de baixo nível das imagens utilizando amostragens esparsas ou densas. Em seguida, selecionamos as características de maior repetibilidade para a criação de representações de médio nível baseadas em dicionários visuais. Posteriormente, treinamos classificadores binários para cada atributo visual os quais atribuem, para cada imagem, uma pontuação de decisão utilizada para obter sua classificação. Também propomos diferentes formas de fusão para o método de descrição de múltiplas escalas. Para consultas mais complexas (mais de dois atributos), avaliamos três abordagens presentes na literatura para combinar ordens (*rankings*): produto de probabilidades, *rank aggregation* e *rank position*. Além disso, propomos uma extensão do método de combinação baseado em *rank aggregation* para levar em conta informações complementares produzidas pelos diferentes métodos. Consideramos quinze classificadores de atributos e, consequentemente, seus negativos, permitindo, teoricamente, $2^{15} = 32\ 768$ diferentes consultas combinadas. Os experimentos mostram que a abordagem de descrição em múltiplas escalas melhora a precisão de recuperação para a maior parte dos atributos em comparação com outros métodos. Finalmente, para consultas mais complexas, a abordagem de descrição em múltiplas escalas em conjunto com versão estendida do *rank aggregation* melhoram a precisão em comparação com outros métodos de fusão como o produto de probabilidades e o *rank position*.

To Elmer, Valeria (my parents), Elio and Ever (my brothers), your unconditional love and support means everything for me.

To my girlfriend Julissa, thanks for demonstrating me that our love has no limits.

To the memory of my beloved ones that are no longer here, you will be always in my hearth.

Acknowledgements

The pathway that brought me here has not been easy, but I am sure that it is only the beginning of new challenges. I would first like to thank God who gave me the grace and privilege to reach the end of this stage of my life and successfully complete it despite the challenges presented.

A special gratitude goes to my family for all of the love, support and encouragement, specially to my mother, *Valeria*, because every day you showed me that the distance will never be a barrier for your unconditional love. Thanks also to my father, *Elmer*, because your words always made me strong to keep going and never give up. Thanks to my brothers, *Elio* and *Ever*, for all you support, I am very proud of you two. I want to express my gratitude to my girlfriend *Julissa*, for her great patience, understanding and support in all my decisions, I love you to the moon and back. All of you have made me stronger, better than I could ever imagined.

I would like to express my profound gratitude to my advisor, Professor *Anderson* for his patience, motivation, enthusiasm, and the immense knowledge that he shared with me. His guidance helped me at every time of the research. Without your support it would have been impossible for me to reach this goal. I will always be grateful to you. Thanks also to the RECOD colleagues for providing me a good atmosphere in our laboratory and for the useful discussions.

Thanks also to my roommates, *Ricardo* and *Roberto*, thank you very much for making the atmosphere of our room as friendly as possible. I never thought I would meet people like you. Thanks to all my friends, for the joyful gatherings and all their advices.

I am grateful to have had the privilege of attending the prestigious University of Campinas (UNICAMP). Finally, I would also like to thank CNPq and CAPES for the financial support.

"Research is the art of seeing what everyone else has seen, and doing what no-one else has done."

Anonymous

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A large set of applications takes facial attributes to identify people. An example is in criminal investigation, when the police are interested in locating a suspect. In such cases, typically eyewitnesses fill out a description form indicating personal traits associated with the target as seen at the time when an event has happened [9]. Based on such descriptions, action can be taken such as manually searching the entire image and video archive looking for a person with similar descriptions. However, this search process has the disadvantage of being time consuming and, often, inaccurate.

To date, most of the automated methods aiming at solving this problem rely on extracting image low-level features [2] and using such features to directly training classifiers with the objective of identifying or detecting the person of interest [30].

Aligned with this trend, in this work we propose to analyze the images with a unified mid-level representation for all associated textual description based on visual dictionaries and fusion techniques. Our approach builds visual dictionaries for representing each facial attribute important features, an approach inspired in the current computer vision and image processing literature.

During the years, textual metadata seemed to be the main currency of most online search engines. Notwithstanding, for the vast majority of images on the internet and in private collections, the attached annotations are often ambiguous, incorrect, or simply not present [30]. In this sense, it is paramount to design and deploy search methods with the goal of automatically labeling images with no need for any associated metadata.

Given a database of face images and a user query comprising a set of attributes representing the presence or absence of a visual feature (e.g., *blond hair woman* with *bangs* wearing *glasses*), our objective is to retrieve an image subset from the database satisfying each facial attribute contained in the query.

Some of the main challenges in this research include: defining a descriptor robust enough to generalize visual attributes on images, for this we use visual dictionaries. Other

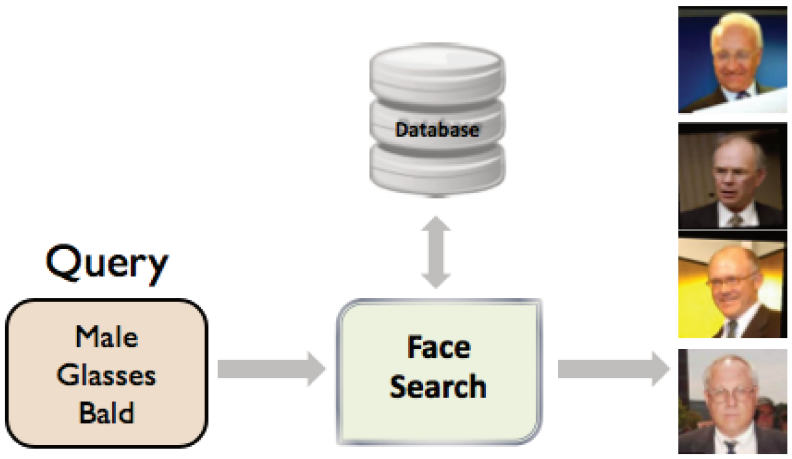


Figure 1.1: Face Search for a specific query $Q = \{male, glasses, bald\}$.

major challenge is at combining the ranking of different visual attributes for the final decision-making process. Figure 1.1 depicts an example of the proposed approach.

1.1 Stages of the Work

This section introduces the two major stages developed in this work. Figure 1.2 depicts the stages of the work proposed to solve the problem of searching for people through textual and visual attributes.

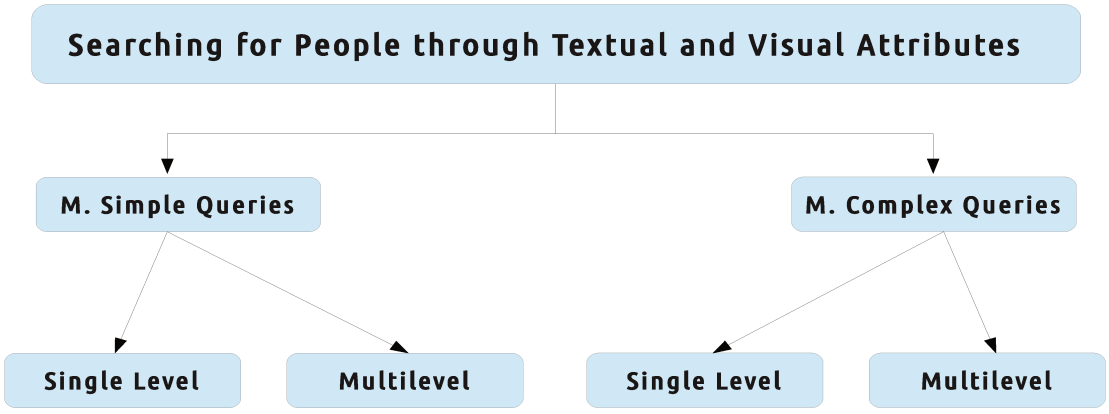


Figure 1.2: Stages of the Work.

1.1.1 Simple Queries

In Chapter 3, we present one single-level and one multilevel approaches to solving queries containing a single attribute. In the single level approach, we characterize the images in the original scale only, while in multilevel we characterize the images in six different scales (0.5×, 0.75×, 1×, 1.5×, 2× and 2.5×) and combine the different results in the end.

1.1.2 Complex Queries

Given a multiple-attribute query, we use some fusion methods to combine the outputs of the classifiers generating a reduced list of people complying with the describable attributes given by the user. The methods used in this work to solve complex queries are: Product of Probabilities, Rank Position and Rank Aggregation. In Chapter 4, we discuss in detail each of these methods. Given a query with multiple attributes, one of the main contributions of these methods is to generate a reduced list of people containing the attributes in the query as Figure 1.3 depicts.

Figure 1.3: Reducing the amount of images for a given query.

1.2 Contributions

The main contributions achieved in this work are:

- We propose a novel representation of low-level features for face description based on points of interest (PoIs) and in a common mid-level representation of such discriminative features using the concept of visual dictionaries. The idea is to design and deploy a characterization approach that captures nuances only in a low-level description (e.g., small variations in a local neighborhood) while also capturing higher-level properties (e.g., mid-level features shared by images of the same category).

- We evaluate sparse and dense feature characterization processes before building the visual dictionaries. The dense sampling-based method greatly improves the description power of our methods when compared to the approach we proposed in [12] based on sparse-based feature sampling which is also part of this dissertation.

- In the multilevel approach, we describe and classify the images in each of the scales in the same fashion as for the single level, then we discuss several methods for combining the different levels.

- With the visual dictionaries introduction and the dense-sampling approach, we achieve significant improvements on the results in comparison to the results obtained in the state of the art [30], our own prior work [12], and also a top-performer texture descriptor in the literature [5, 41].

- In order to solve complex queries, we evaluate three methods to combine the different ranked lists. The obtained results showed good performance in comparison to the state of the art [30, 47]. In some cases, for multiple-attribute queries, we obtained 100% of precision in the top positions.

1.3 Organization of the Text

We organized the remainder of this work into five chapters: Chapter 2 explains the related work. Chapters 3 and 4 present our methods for solving single and complex queries, respectively. Chapter 5 presents the experiments and the obtained results. Finally, Chapter 6 presents the conclusions of our work.

Chapter 2

Related Work

Our work here can be regarded as a form of Content-based Image Retrieval (CBIR), in which the content is limited to face images and the queries are describable attributes or keywords related to the face. In this chapter, we present the related work to our research in terms of objectives to be achieved. At the beginning, we detail the related researches to searching for people through visual attributes. Then, we present the state-of-the-art of the principal rank fusion techniques.

2.1 Searching for people through visual attributes

Several researchers have been conducted to improve the performance of people search engines. Thus, currently being developed robust features to represent visual attributes and advanced technologies to detect faces.

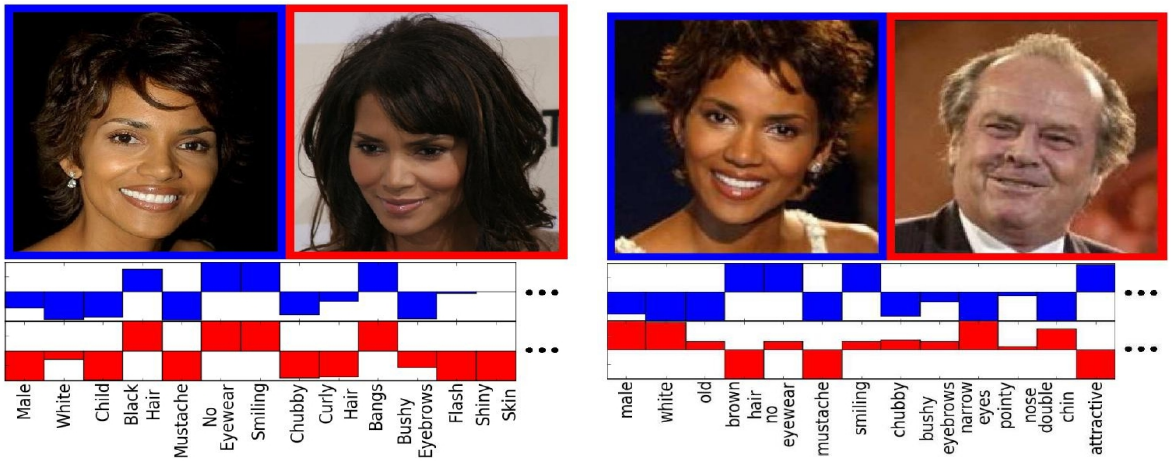
Several works have been done regarding face characterization. Early work on appearance-based face verification [50] looked at the distance between pairs of images in a low dimensional subspace obtained using Principal Components Analysis (PCA). Variations in pose, expression, and lighting cause significant difficulties in the face verification task. To solve these problems, sometimes alignment, especially in 3D are used. Unfortunately, in a real-world scenario, 3D alignment is difficult (expensive) [30]. The Fisherfaces work [3] showed that linear discriminant analysis could be used for simple attribute classification such as glasses/no glasses.

In computer vision, the use of attributes has recently been receiving much attention from a number of different groups. Prior work on visual attributes has focused mainly on ethnicity and on gender classification [7, 18]. Ferrari and Zisserman [16] were probably the first ones to propose visual attributes as text labels that can be automatically assigned to scenes, categories, or objects using machine learning techniques. Histograms of Oriented Gradient (HoG) is the most commonly used to characterize visual attributes [28, 29, 30].

Color descriptors have also been used to represent visual attributes. In [1], the authors discussed the main contributions of color to face recognition. Notwithstanding, the authors concluded that color does not provide diagnostic information for face recognition, this results were consistent with earlier reports [27]. However, color cues are not entirely disregarded, they contribute significantly under degraded conditions [1]. In this work, we do not evaluate color descriptors, because the images with which we conducted our experiments are in good conditions.

In [28], the authors created the first image search engine based entirely on faces. Using simple text queries such as "smiling men with blond hair and mustaches", users can search through over 3.1 million faces which have been automatically labeled on the basis of several facial attributes. The proposed approach was based on a combination of Support Vector Machines and Adaboost which exploits the strong structure of faces to select and train on the optimal set of features for each attribute.

The work of Kumar et al. [29] describes two methods for face verification. The first method - "attribute" classifiers - uses binary classifiers trained to recognize the presence or absence of describable aspects of visual appearance (e.g., gender, race, and age). The second method - "simile" classifiers - removes the manual labeling required for attribute classification and instead learns the similarity of faces, or regions of faces, to specific reference people. The authors evaluated their proposed methods with the Labeled Faces in the Wild (LFW) dataset. Figure 2.1 depicts the results obtained by different attribute classifiers for two images of the same person (a), and two images of the different persons (b).



(a) (b)

Figure 2.1: Responses of different visual attribute classifiers for image of (a) the same person, (b) different persons [29].

In [12], a previous work of our own, we proposed a sparse characterization method based on SURF [2] to extract low-level features and build visual dictionaries to represent visual attributes.

The works proposed by Kumar et al. [30] and Park et al. [37] regarding combination of textual and visual features are the most similar to ours. In [30], the authors explore direct image pixel intensities, edge magnitude, and edge orientation features with and without normalization for searching faces based on describable features. For dealing with multiple attribute queries, the authors use product of probabilities. Figure 2.2 shows the face verification pipeline proposed by the authors. In [37], the authors use soft biometric traits (scars, marks, and tattoos) for speeding up face matching and narrowing down face searching tasks. Additional research in describable visual attributes is also present in the computer vision community [28, 29, 32].

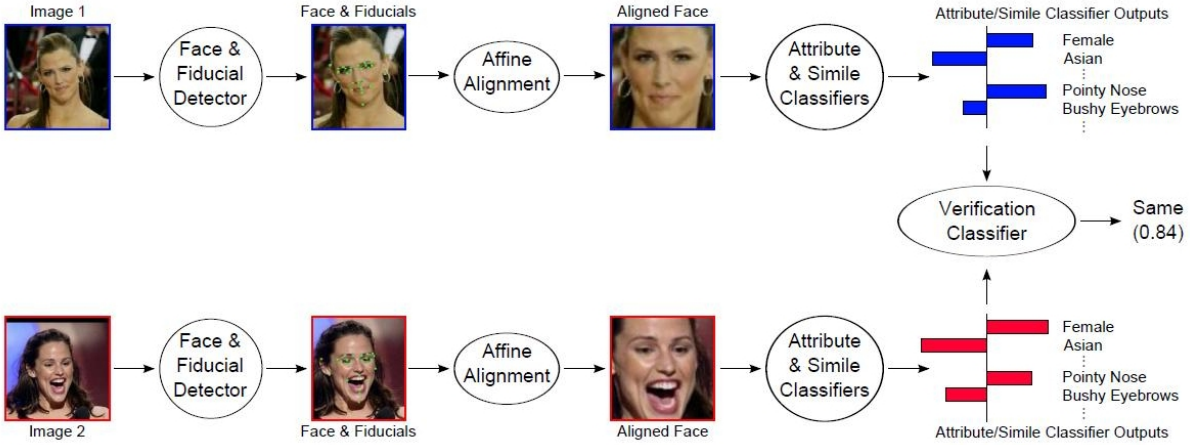


Figure 2.2: The face verification pipeline using attribute and simile classifiers proposed in [30].

Regarding the underlying objective, our method is similar to [30] and [37]. However, the three methods have important differences. For instance, our method relies on a mid-level image representation based on visual dictionaries which serve as a projection space for the low-level features extracted from the faces. This projection space is more discriminative in some situations as we shall discuss in the next sections. In addition, here we explore the power of dense sampling characterization techniques as well as different fusion methods.

2.2 Rank fusion techniques

For combining different decision scores towards a unified decision-making scheme, researchers have proposed various approaches. In [44], the author proposed the Borda count method and the Condorcet algorithm for combining documents. In Borda count method, the highest ranked individual (in an n-way vote) gets n votes and each subsequent gets one vote less (so the number two gets $n-1$ and the number three gets $n-2$ and so on). Then, for each alternative, all the votes are added up and the alternative with the highest number of votes wins the election. In the Condorcet election method, voters rank the candidates in the order of preference. The vote counting procedure then takes into account each preference of each voter for one candidate over another. The Condorcet voting algorithm is a majoritarian method that specifies the winner as the candidate, which beats each of the other candidates in a pair wise comparison.

Bayesian networks have also been explored for intelligent decision [47] as well as logical operations (e.g., AND, OR), majority voting, summing decision scores [9], and behavior knowledge space [31].

Recently, Scheirer et al. [48] have introduced techniques based on the statistical Extreme Value Theory for constructing normalized "multi-attribute spaces" from raw classifier decision scores. The authors map each decision score (in its own domain) onto a probability that the given attribute is present in the image. Following [48], in this work we also normalize classifier decision scores and map them onto probabilities.

In addition to product of probabilities, in this work we used rank position [36] and rank aggregation to combine different decision scores.

2.2.1 Rank position (reciprocal rank) method

In this approach, to merge the images into a unified list only the rank positions of retrieved images are used. Retrieval systems determine the rank positions. When a duplicated image is found, the inverse of its rankings are summed up, since the images returned by more than one retrieval system might be more likely to be relevant. The following equation shows the computation of the rank score of document i using the position information of this image in all of the systems ($j = 1...n$).

$$r(d_i) = \frac{1}{\sum_j 1/position(d_{ij})},$$

In this approach, first Rank Position score of each document to be combined is evaluated, then using these rank position scores, documents are sorted in non-decreasing order.

Example. Suppose that we have four different retrieval systems A, B, C, and D with a document collection composed of documents a, b, c, d, e, f, and g. Let us assume that for a given query their top four results are ranked as follows:

$A = (a, b, c, d)$
$B = (a, d, b, e)$
$C = (c, a, f, e)$
$D = (b, g, e, f)$

Now, we compute the rank position of each document in the document list, and the rank scores of the documents are as follows:

$r(a) = 1/(1 + 1 + 1/2) = 0.4$
$r(b) = 1/(1/2 + 1/3 + 1) = 0.54$, and so on

The final ranked list of documents is $a > b > c > d > e > f > g$, i.e., a is the document with the highest rank, i.e., it is the top most document; b is the second document, etc.

2.2.2 Rank aggregation

Rank aggregation is a traditional approach which has been employed with this objective in many applications [33]. Basically, rank aggregation approaches aim at combining different rankings in order to obtain a more accurate one. More precisely, rank aggregation can be seen as the task of finding a permutation that minimizes the Kendall-tau distance to the input rankings. The Kendall-tau distance is defined as the sum over all input rankings of the number of pairs of elements that are in a different order in the input ranking than in the output ranking [46].

Recently, a rank aggregation method was proposed aiming at considering the relationships among images being ranked in content-based image retrieval tasks. The RL-Sim Algorithm [40] considers the similarity among ranked lists for analyzing the relationships among objects.

In this work, we use the traditional rank aggregation as a voting system. First, we evaluate which documents are in the first position in each one of the results. If a document is in the first position in the most of the results, then, this document will appear in the first position in our final ranked list. Next, we remove the document of our results and we repeat the process until remove all documents.

Example. Suppose that we have three different retrieval systems A, B and C with a document collection composed of documents $a, b, c,$, and d. Let us assume that for a given query their results are ranked as follows:

$A = (a, b, c, d)$
$B = (a, c, b, d)$
$C = (b, c, a, d)$

In this case, the document a is in the first position in the most of our results, so the document a is removed of our results and will appear in the first position in our final list.

$A = (b, c, d)$
$B = (c, b, d)$
$C = (b, c, d)$
$FinalList = (a)$

Then, we repeat the same process until we remove all the documents in our results:

$A = (c, d)$
$B = (c, d)$
$C = (c, d)$
$FinalList = (a, b)$

$A = (d)$
$B = (d)$
$C = (d)$
$FinalList = (a, b, c)$

$A = ()$
$B = ()$
$C = ()$
$FinalList = (a, b, c, d)$

Chapter 3

Methods for Simple Queries

In order to solve the problem of searching people through textual and visual attributes, the first step of our approach is to solve the problem for simple queries (e.g., queries composed by only one attribute). Then, we can use different methods for combining the results obtained for these queries and solve more complex searches. In this chapter, we explain in detail our proposed methodology to solve this first step. We propose two methodologies for solving simple queries: one single-level and one multilevel. Some of the methods developed herein resulted in the publication [12].

3.1 Single Level

In this section, we explain how to solve simple queries using a single-level characterization approach. Our proposal is based on three stages: image characterization, visual word dictionaries and image classification.

3.1.1 Image Characterization

Assuming the facial images are, at least, roughly aligned, we use an algorithm for extraction of points of interest to represent their visual content and to characterize their surrounding regions. We extract "low-level" features related to the attribute of interest from different face regions. The features we use provide the representation of visual content of a given image through a set of Points of Interest (PoIs) in the image. In this sense, in [12], we used a sparse-sampling characterization approach based on the Speeded Up Robust Features (SURF) algorithm [2]. Then, in [11], we extended upon our prior work and used a dense-sampling characterization method for finding more discriminative points for each attribute of interest.

More specifically, we extract representative interest points in an image region using either a sparse- or dense-sampling approach.

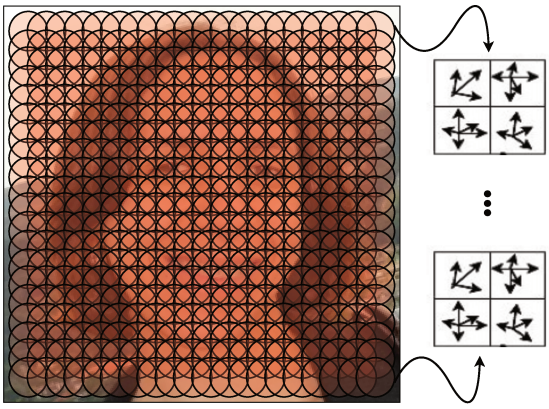
Sparse-sampling approach For the sparse-sampling approach, we extract points of interest representing the visual content of an image through the same algorithm presented in [12]. Here, the objective is to find scale-invariant interest points such that we have a representation robust to some possible image transformations (e.g., rotations, scale, and partial occlusions). In the specific context of faces, normally this is not a serious problem since it is possible to roughly align faces based on eye distances. For finding such interest points, we use the well-known SURF algorithm [2], whose four main steps are:

1. **Feature Point Detection:** In this stage, SURF uses a Hessian detector approximation and integral images [51] to speed up the involved operations.
2. **Feature Point Localization:** SURF uses the determinant of the Hessian for both location and scale. To localize the interest points in the image across different scales, the method performs nonmaximum suppression in a $3 \times 3 \times 3$ neighborhood. The determinant's maxima of the Hessian matrix are then interpolated in scale and image space.
3. **Orientation Assignment:** In order to be invariant to rotation, SURF calculates the Haar-wavelet responses for both x and y directions within a circular neighborhood of radius $6s$ around the interest point, with $s = \sigma$ the scale at which the interest point was detected. The dominant orientation is estimated by calculating the sum of all responses within a sliding orientation window covering an angle of $\frac{\pi}{3}$ and the interest point gets the orientation of the longest calculated vector [2].
4. **PoI Characterization:** For the extraction of the descriptor, SURF constructs a square region centered around the interest point and oriented along the orientation selected in the previous stage. The region is split up regularly into smaller 4×4 square sub-regions and, for each sub-region, the method computes a Haar wavelet responses at 5×5 regularly-spaced sample points. Finally, the wavelet responses are summed up over each subregion and form a first set of entries to the feature vector [2].

Dense-sampling approach For the dense-sampling approach, for each image, we extract the center point in a regular lattice of size 3×3 in the whole image with superposition. Then we use SURF's last two steps [2] (PoI Assignment and Characterization) to describe

these points. This approach greatly improves the description power of our method when compared to the method we proposed in [12] based on sparse-based feature sampling.

As we deal with partially-aligned image faces, we can achieve better representation by constraining the feature extraction to regions of interest most likely related to an attribute of interest. In this context, Figure 3.1 depicts the dense characterization approach, the PoI description using SURF's last two steps and the regions of interest used for the feature selection on affine-aligned face images.



(a) Dense-Sampling and PoI Characterization

(b) Regions of Interest for Distinct Describable Attributes

Figure 3.1: (a) The Dense-Sampling approach to finding the points of interest as well as its center point description using gradient orientation assignment. (b) Regions of interest for each describable attribute. R_1: glasses. R_2: male, asian, black, senior, chubby, white and youth. R_3: bald, bangs, black hair, blond hair, gray hair. R_4: mustache. R_5: beard. Note that the faces are aligned by the eye as the first step.

3.1.2 Visual Word Dictionaries

After extracting points of interest in the image, we compute a mid-level image representation using visual dictionaries for preserving the distinctiveness power of the descriptors while increasing their generalization [8]. Basically, we select the most representative points of interest according to the describable attribute of interest by means of either random selection or clustering. The final set of selected points of interest represents a projection space onto which the points of interest found in any image are projected creating its representative feature vector. Given a set of 'words' from the visual dictionary, we find the feature vector representing each image of the collection analyzing and assigning each of its PoIs to the closest visual word in the dictionary, a process sometimes called projection or hard assignment coding [4].

As previously mentioned, surely low-level features are not enough to fully represent images of faces. When searching for a specific target, this discriminative power is extremely important. However, when searching for more complex categories, this high discriminability is a problem since the ability to generalize becomes uppermost. As these solutions are often designed for exact matching [34, 2], they do not translate directly into good results for image classification. In this sense, we can use the concept of visual vocabularies [8, 15, 25, 10] to increase the descriptor generalization.

When creating a visual vocabulary, each set of points of interest becomes a 'visual word' of a 'dictionary'. Searching for *"senior people"*, for instance, in a database of images with faces, consists of selecting and creating a database of training examples comprising training positive examples (i.e., faces of senior people) and negative images (i.e., faces of non-senior people). The points of interest are calculated within the region of interest for the attribute 'senior' (R_2) as Figure 3.1 depicts.

After filtering the points of interest, we create a visual dictionary representing distinctive features of images for each specific describable attribute either using random selection of points of interest or using a clustering algorithm.

Here is another crucial detail of the methods we discuss in this work when compared to traditional visual dictionaries approaches in the literature. To create the visual dictionary, we set $\frac{k}{2}$ words to represent the presence of the describable attribute (e.g., senior people) and $\frac{k}{2}$ for the absence of such attribute (e.g., non-senior people). Normally, in the literature, a dictionary is created without class information (bag of words). Throughout a series of experiments and also from previous experience of our group in other classification problems [24, 42, 45], we have observed that, at least for binary visual classification problems, the class-aware dictionary creation process is more effective.

3.1.3 Image Classification

For the final classification procedure, we use 2-class machine learning classifiers such as Support Vector Machines (SVMs) for each describable attribute of interest. We train each 2-class classifier with the signatures (feature vectors) of the training images containing positive (images containing a specific attribute) and negative (images without the attribute) examples.

After training, all of the images, except the ones contained in the training set, are classified yielding a decision score. Search results are ranked by confidence, so that the most relevant images are shown first. We use the computed distance to the classifier margin decision boundary as a measure of confidence similar to [30]. We then sort the images in a decreasing order of decision score.
The main steps of the proposed method in a single level are depicted in Figure 3.2.

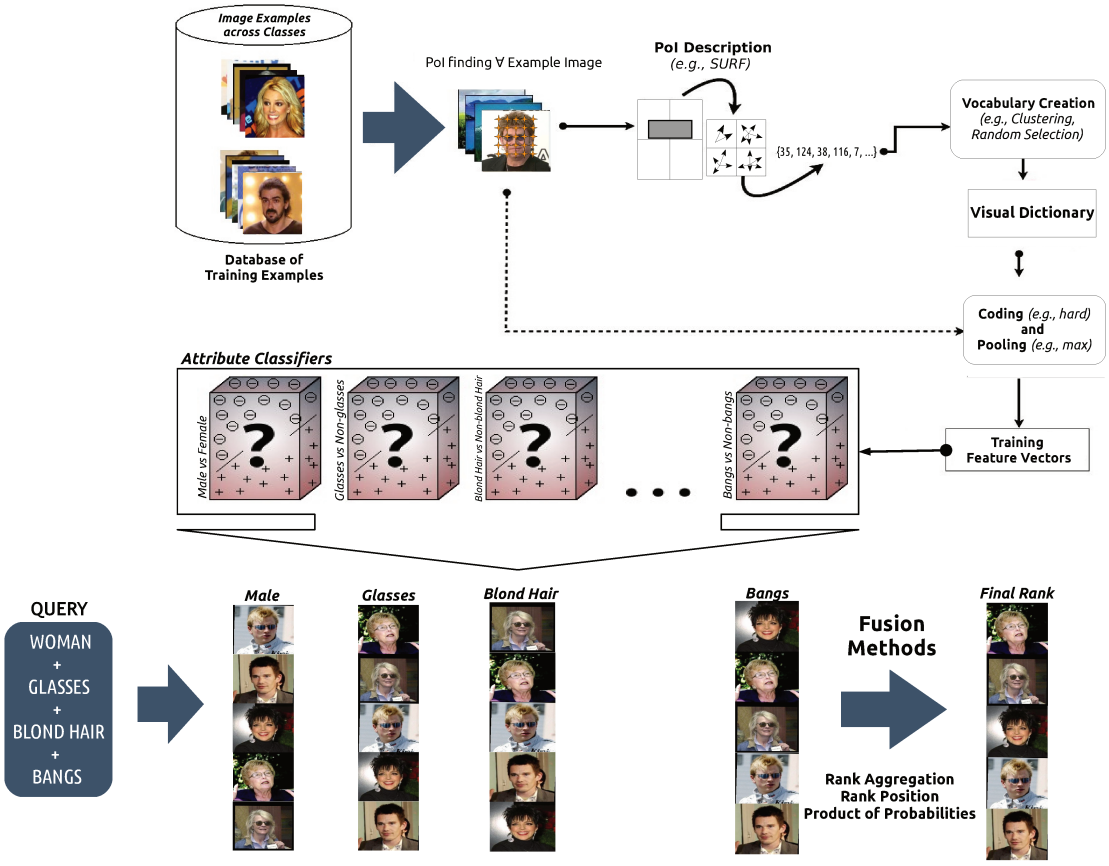


Figure 3.2: The proposed approach aims at searching for people using textual and visual attributes in a single level.

3.2 Multilevel Approach

In order to improve the characterization of the images and consequently the performance of our attribute classifiers, we also propose a multilevel image characterization approach. In this section, we explain this novel scheme to solve simple queries. Initially, our proposal is similar to the single-level approach and also uses three steps: image characterization, visual word dictionaries and image classification. However, as we now have different answers for each level/scale, we need to combine them toward a final answer. For that, we propose some combination methods taking into consideration the resulting lists produced by the attribute classifiers across different scales.

3.2.1 Image Characterization

To characterize the images in a multilevel approach, we first need to extract the regions according to each attribute as depicted in Figure 3.1. Then, we scale the regions in different scales. In this work, we evaluate six scales: 0.5×, 0.75×, 1×, 1.5×, 2× and 2.5× and represent them respectively as *Level 1*, *Level 2*, *Level 3*, *Level 4*, *Level 5* and *Level 6*. Finally, for each region at each level, we describe the images using a sparse-sampling approach in the same fashion that in a single level. We do not evaluate the dense-sampling in the multilevel approach because it is too expensive. Figure 3.3 shows the image characterization process in the multilevel approach.

3.2.2 Visual Dictionaries

Having created the descriptions for the images in the six scales, we build the feature vectors for each image using the best dictionaries previously generated in a single level (i.e., for each attribute, we simply use the dictionary created with points calculated with the single-level characterization method). Figure 3.3 depicts how we use the best dictionaries anteriorly created to generate a feature vector for each image at each level.

3.2.3 Image Classification

To classify the images in the different levels and generate a score for each image, we use the best SVM models created in a single level (i.e., for each attribute, we simply use the SVM classification model created with feature vectors created through the single-level characterization method). Then, for each region, according to each attribute, we obtain six different scores from each attribute classifier. Figure 3.3 shows the image classification process in the multilevel approach.

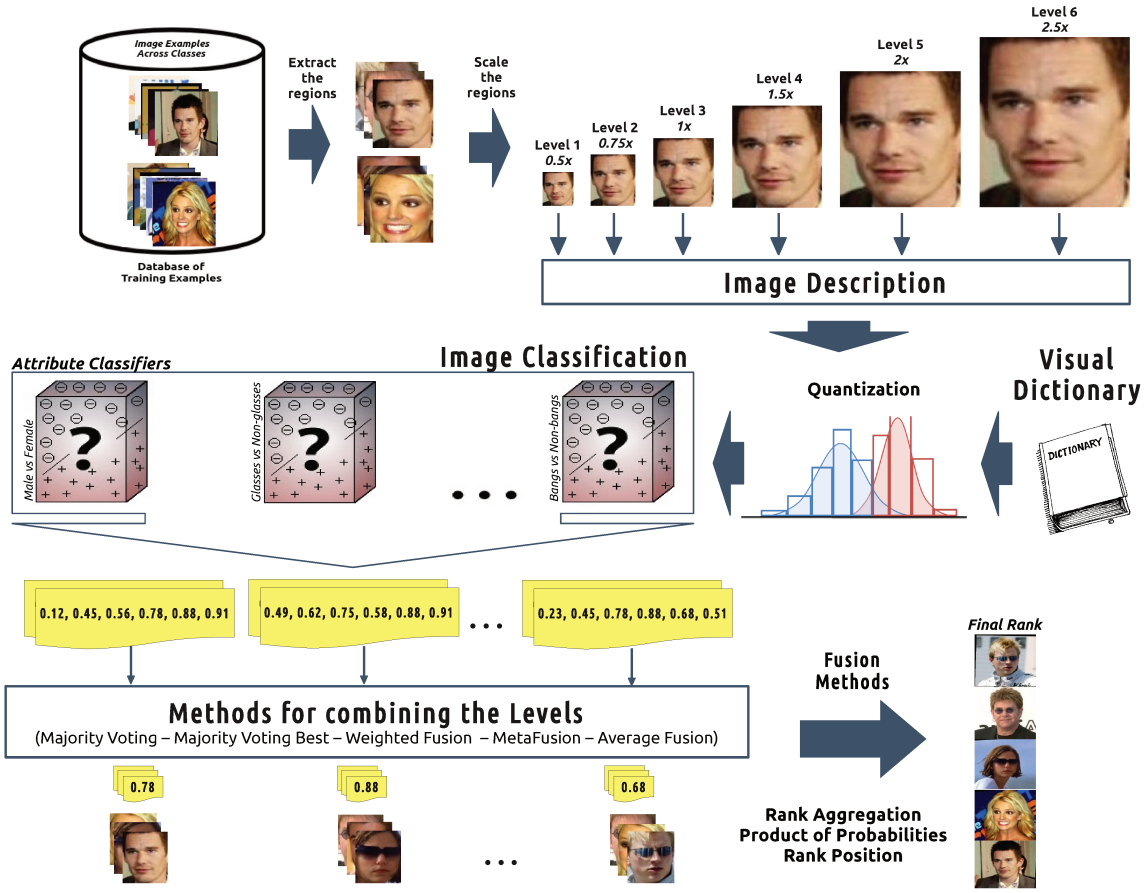


Figure 3.3: The proposed approach aims at searching for people using textual and visual attributes in a multilevel approach.

3.2.4 Methods for combining the levels

To combine the different scores obtained in each one of the six levels, we propose five methods: Majority Voting, Majority Voting Best, Weighted Fusion, MetaFusion and Average Fusion. For a better understanding of these methods we present the following example: consider the picture depicted in Figure 3.4, and the *query = "male"*. For this example, we solve the query using the multilevel approach as depicted in Figure 3.3. Then, using our previously trained classifiers for the attribute "male", suppose we obtain the scores in Table 3.1.

Majority Voting (MV): This method works as a democratic system. First, we evaluate how many scores are greater or equal to 0.5 (which represents that the image belongs to the positive class), and how many scores are smaller than 0.5 (which represents that

Figure 3.4: Ben Affleck.jpg - image obtained from PubFig dataset [29].

Table 3.1: Scores obtained by our attribute classifier for each one of the six levels.

Image	**Scores**					
	Level 1	**Level 2**	**Level 3**	**Level 4**	**Level 5**	**Level 6**
Ben Affleck.jpg	0.01	0.56	0.59	0.89	0.94	0.97

the image belongs to the negative class). To obtain the final score using this method, we simply average the scores of the winning class. For instance, if the number of scores belonging to the positive class is greater or equal to the number of scores belonging to the negative class, we average the scores belonging to the positive class. For the image in Figure 3.4, most of the scores in Table 3.1 belong to the positive class (Level 2, Level 3, Level 4, Level 5 and Level 6), then we averaged these scores. The final score is 0.79.

Majority Voting Best (MVB): Majority Voting Best is a variant of the previous method. The main difference is that before counting the votes, we remove the three closest values to 0.5 (decisions for which the classifiers are mostly in doubt). After that, we perform the majority voting in the remaining three values. For the example in Table 3.1, we remove the Level 2, Level 3 and Level 4, then we use the MV method only in the Level 1, Level 5 and Level 6. By using this method, sometimes we remove votes with high confidence, but even so, our experiments shown a good performance in comparison with the other methods. The final score for this image using the MVB method is 0.64.

Weighted Fusion (WF): In this method, we weighted the scores using the confidences obtained by each attribute classifier. First, we normalize the confidences by dividing each

one of these by the sum of all confidences. Then, we multiply the normalized confidences by the scores obtained in each level. Finally, we get the final score by adding all the weighted scores. In our experiments, the confidences used for the attribute "male" were as follows. Level 1: 0.744, Level 2: 0.806, Level 3: 0.834, Level 4: 0.804, Level 5: 0.818 and Level 6: 0.82. For the example above, the final score using the WF method is 0.67. The classifier confidences are learned during training using a separated validation set.

MetaFusion (MF): In this method, we trained an SVM classifier using the six scores obtained in each level as a feature vector. The classifiers are learned during training using a separated validation set. Then, we evaluate the images using the classifiers previously trained obtaining the final score. For our example, the final score is 0.73.

Average Fusion (AF): This method is the simplest of all. The final score is obtained by averaging the scores of the different levels. In our example, the final score is 0.66. Note that AF is a special case of WF in which all the weights are set to 1.

Finally, Table 3.2 shows the initial scores in the different levels and the final (combined) scores obtained using the proposed methods for the example above.

Table 3.2: Initial and final scores obtained using the proposed methods.

	Initial Scores						Final Scores				
Image	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	MV	MVB	WF	MF	AF
Ben Affleck.jpg	0.01	0.56	0.59	0.89	0.94	0.97	**0.79**	**0.64**	**0.67**	**0.73**	**0.66**

Chapter 4

Methods for Complex Queries

In order to solve complex queries (e.g., queries with more than one attribute), we combine the classification confidence of different attribute classifiers such that the final ranking refers to images in decreasing order of relevance regarding the query terms.

For solving a query such as "give me faces depicting a *white woman* with *blond hair* wearing *glasses* and with *bangs*", we fuse the scores given by each attribute classifier (a rank for *white people*, a rank for *non-male*, a rank for *blond hair*, a rank for *glasses* and a rank for *bangs*) to produce a ranking based on the combination of the attributes.

We consider three combination methods herein: product of probabilities, rank position and rank aggregation computed over the scores of each individual attribute classifier. In this chapter, we explain at length each of these methods. Most of the methods discussed in this chapter were published in [12] and then improved upon and submitted to a journal [11].

4.1 Product of Probabilities

Given a query $Q = \{a_p, ..., a_q\}$, this method finds, for each image I_i in the database, its scores in each rank $r_p, ..., r_q$, and multiply the values [30]. The I_i's resulting scores are then sorted in decreasing order. To prevent high confidence for one attribute from dominating the search results, it is necessary to convert the confidences to probabilities. We transform each score s_i in a new score s'_i ensuring that the difference between s'_i and s'_{i+1} is equal to the difference between s'_{i+1} and s'_{i+2}. Figure 4.1 depicts an example of rank fusion using product of probabilities before normalization aforementioned.

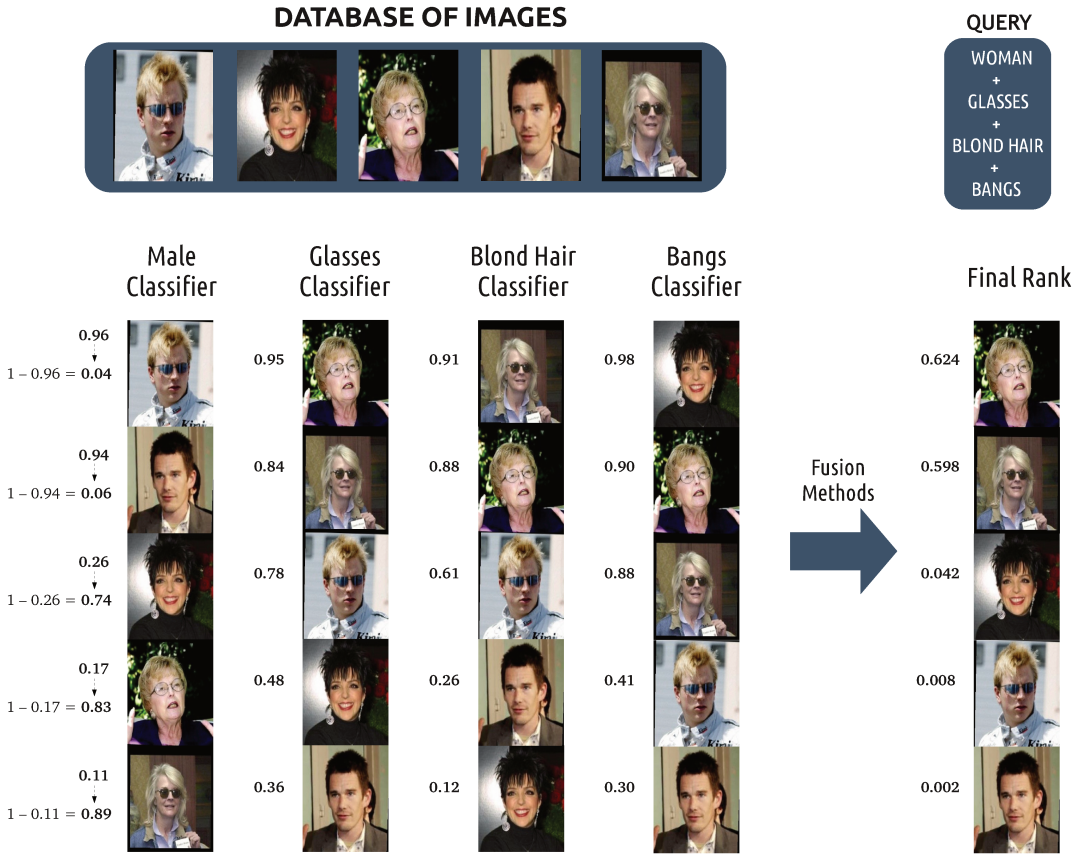


Figure 4.1: Combining scores obtained by different classifiers for a given query using product of probabilities. In this case, to evaluate the attribute woman, we use the complementary probabilities of the male classifier outputs.

4.2 Rank Position

Given a query $Q = \{a_p, ..., a_q\}$, this method finds, for each image I_i in the database, its positions in each rank $r_p, ..., r_q$, and use the following equation to obtain a final rank

$$r(d_i) = \frac{1}{\sum_j 1/position(d_{ij})},$$

where $r(d_i)$ is the final score for each image I_i, and $position(d_{ij})$ is the position obtained in the rank r_j for the image I_i. In this approach, the score of each image I_i is calculated by the above equation and then the images are increasingly ordered depending on the scores, obtaining a final rank.

4.3 Rank Aggregation

In this section, we explain the traditional rank aggregation algorithm which has been used in our previous work [12] and we describe an extension to the RL-Sim Algorithm [40] for combining ranked lists computed by different classifiers.

4.3.1 Traditional Rank Aggregation

Rank aggregation takes m different rankings of n candidates (possibly given by different voters) and aggregate them in a single ranking. Kemeny [26] proposed an aggregation mechanism that produces the global ranking that minimizes the number of inverted pairs with the input rankings. The algorithm produces a *Footrule-optimal aggregation* that minimizes the sum of the differences of the ranks.

4.3.2 Modified Rank Aggregation

Let A=$\{a_1, a_2, \ldots, a_n\}$ be a set of attributes and let C=$\{c_1, c_2, \ldots, c_n\}$ be a set of classifiers for each attribute. Let I_C=$\{img_1, img_2, \ldots, img_N\}$ be an image collection. Let $p_l(img_i)$ be the probability assigned by the classifier c_l to the image img_i.

We aim at combining the probabilities given by classifiers $c_l \in C$, in order to compute a combined ranked list τ_A. The ranked list τ_A=(img_1, img_2, …, img_{n_s}) can be defined as a permutation of the collection I_C. A permutation τ_A is as a bijection from the set collection I_C onto the set $[N] = \{1, 2, \ldots, N\}$. For a permutation τ_A, we interpret $\tau_A(img_i)$ as the position (or rank) of image img_i in the ranked list τ_A.

The ranked list τ_A is computed based on a similarity measure $sim(\cdot)$. We can say that, if img_i is ranked before img_j in the ranked list τ_A (that is, $\tau_A(img_i) < \tau_A(img_j)$), then the dissimilarity measure $sim(img_i) \geq sim(img_j)$. The rank aggregation algorithm aims at defining the similarity measure $sim(\cdot)$ in terms of probabilities given by the set of classifiers C.

The first step of the algorithm computes the similarity measure by multiplying the classifiers probabilities, as follows:

$$sim(img_i) = \prod_{l=1}^{n} p_l(img_i) \tag{4.1}$$

Given this initial similarity measure, we can obtain an initial ranked list τ_A. Since the top positions of the ranked list τ_A represent the main region of interest, we apply a second step which aims at improving the ranked list at top positions. We used an approach inspired by the RL-Sim Algorithm [40] for analyzing the relationships among

the images at the top N_S[1] positions of the initial ranked list.

First, we compute a dissimilarity measure $d(\cdot,\cdot)$ between any two given images $img_i, img_j \in I_C$ at the first top N_S positions of τ_A ($\tau_A(img_i) \leq N_S$), as follows:

$$d(img_i, img_j) = \sqrt[n]{\prod_{l=1}^{n} 1 + (p_l(img_i) - p_l(img_j))^2} \tag{4.2}$$

Given the computed dissimilarity measure, a ranked list τ_i is computed for each image at the top N_S positions of τ_A. We can say that, if img_x is ranked before img_y in the ranked list of img_i (that is, $\tau_i(img_x) < \tau_i(img_y)$), then $d(img_i,\ img_x) \leq d(img_i,\ img_y)$. Given two images img_i, img_j and their respective ranked lists τ_i, τ_j, a new and more effective distance measure between the two images can be computed by considering the similarity of ranked lists, at their top k positions[2] [40].

An approach to computing the similarity between two ranked lists τ_i and τ_i proposed in [13] is the intersection metric ψ, which measures the extent of overlap between τ_i and τ_i. Equation 4.3 formally defines the intersection metric ψ:

$$\psi(\tau_i, \tau_j, k) = \frac{\sum_{k_c=1}^{k} \mid kNN(img_i, k_c) \cap kNN(img_j, k_c) \mid}{k}, \tag{4.3}$$

where $kNN(img_i, k_c)$ is a set of top k_c images of the ranked list τ_i.

Finally, the top N_S positions of the ranked list τ_A are updated according to a new similarity measure computed for considering the similarity of top k ranked lists, as follows:

$$sim(img_i) = 1 + \frac{sim(img_i)}{\frac{\sum_{j=1}^{k} \psi(\tau_i, \tau_j, k) \times (j-1)}{\sum_{j=1}^{k} (j-1)}} \tag{4.4}$$

Note that the term $(j-1)$ aims at defining a higher weight for images at top positions of the ranked list τ_i.

[1] We used $N_S = 200$ in our experiments.

[2] We used $k = 15$ in our experiments.

Chapter 5

Experiments and Results

In this chapter, we present the experiments we performed to validate the approaches we discuss in this work. In this work, we represent a query as a set of attributes $Q = \{a_p, ..., a_q\}$. We consider 15 attributes: *asian*, *bangs*, *bald*, *beard*, *black skin*, *black hair*, *blond hair*, *chubby*, *glasses*, *gray hair*, *male*, *mustache*, *senior*, *white* and *youth* and represent them respectively as *as*, *bg*, *ba*, *be*, *bl*, *bah*, *boh*, *ch*, *gl*, *gh*, *ma*, *mu*, *se*, *wh* and *yo*.

The absence of an attribute is shown with an overline (e.g., $\overline{gl}$). For example, a query that contains *glasses*, *non-beard*, and *non-mustache* is represented by $Q = \{gl, \overline{be}, \overline{mu}\}$. Finally, the fusion functions $F : Q \rightarrow R$ product of probabilities, rank aggregation and rank position are denoted respectively as $F_{product}$, $F_{aggregation}$ and $F_{position}$.

In order to show the results in the same order in which we present our methods, we divide this chapter into three sections: datasets, results for simple queries and results for complex queries. First, we briefly explain the datasets used in our experiments. Thereafter, we present in detail the experiments and the obtained results for both methods for simple queries (Chapter 3) and for complex queries (Chapter 4).

5.1 Datasets

In this work, we performed our experiments using two datasets: Labeled Faces in the Wild (LFW) [23] and Public Figures Face Database (PubFig) [29]. These datasets are the most commonly used to evaluate visual attributes in the literature [12, 28, 29, 30, 47, 48]. We used the LFW dataset to build the best dictionaries and train all the attribute classifiers. Then, we used the PubFig dataset to evaluate the attribute classifiers using the best dictionaries and the best models generated in the LFW dataset. This cross-dataset process is another main contribution of this work since it presents a more robust way of comparing different attribute classifiers in circumstances closer to real-world conditions.

5.1.1 Labeled Faces in the Wild (LFW)

We used the Labeled Faces in the Wild (LFW) dataset [23] which comprises 13,000+ face images with faces designed for unconstrained face recognition. Here we used an LFW version whose images were aligned with funneling since it is not our purpose to validate any face registration algorithm. We emphasize, however, that any eye location technique (e.g., [21]) could be used to find the eyes in the images and perform eye-based geometric alignment. In [12], we evaluated six attributes in this dataset (now extended to consider the 15 of interest in this work), and we obtained good results in comparison with the state-of-the-art [29]. In this dataset, we used 6,000 images to train all our attribute classifiers and the remainder of the images to test. Thereby, we avoid the overlap when combined attributes. Figure 5.1 shows three examples of the aligned images contained in LFW dataset.

Figure 5.1: Examples of the aligned images contained in the LFW dataset.

5.1.2 Public Figures Face Database (PubFig)

The PubFig database [29] is a large, real-world face dataset comprising 58,797 images of 200 people collected from the internet. Unlike most other existing face datasets, these images are taken in completely uncontrolled situations with non-cooperative subjects. Thus, there is large variation in pose, lighting, expression, scene, camera, imaging conditions and parameters, etc.

We performed the experiments in 39,023 images from PubFig given that the authors made available only a list with the links to the original images, therefore many of the links in the list were broken. Due to the completely uncontrolled acquisition conditions present in the images of PubFig, we had to extract only the faces and then scale the images to a fixed size. Figure 5.2 depicts three examples of the images in PubFig.

Figure 5.2: Examples of the images contained in PubFig dataset.

5.2 Results for Simple Queries

In order to solve simple queries, we use the methods explained in Chapter 3. We present the results for both single level and multilevel approaches.

5.2.1 Single Level

In this section, we use the methods detailed in Chapter 3. Before going any further, we explore the importance of the number of words in the dictionary creation for each considered attribute classifier and the effectiveness of each binary classifier for finding one describable attribute at a time. We assess three vocabulary sizes: 100, 500 and 1,000 words, where half of the words refers to the presence of the attribute and half to its absence. We select the best-performing dictionary for each attribute.

For all attributes we consider, we use 1,000 training images and 500 testing images from the LFW dataset. We used 2-class SVM classifiers with a radial basis kernel. The SVM parameters were calculated for each training set, using the standard LibSVM's built-in grid search fine-tuning algorithm[1]. For the creation of the visual dictionaries, we evaluate three methods detailed below.

Method #1: In this method, we use a sparse-sampling approach to extracting the low-level features. We use an algorithm for extraction of points of interest to represent their visual content and to characterize their surrounding regions [12]. For this task we use all stages of the SURF [2] algorithm. After this, to build the visual dictionaries we use the well-know clustering algorithm k-means to represent visual attributes. This is basically the method we proposed in [12]. There we validated using six describable attributes. Then, we extend the validation to 15 attributes.

Table 5.1 shows the classification accuracy (#correctly classifications / #misclassifications) and the area under the Receiver Operating Curve (ROC) for each case of this method.

Table 5.1: Accuracy and area under the curve (AUC) for each facial attribute considering Method #1.

Attribute	Accuracy	AUC	Number of Words
Asian	73.80%	81.37%	100
Bald	83.40%	90.23%	100
Bangs	80.20%	88.50%	500
Beard	79.00%	87.40%	500
Black	79.20%	85.07%	1,000
Black Hair	81.60%	89.15%	500
Blond Hair	77.20%	85.85%	500
Chubby	69.80%	74.61%	1,000
Glasses	80.40%	88.35%	500
Gray Hair	85.60%	90.95%	500
Male	81.60%	90.82%	1,000
Mustache	84.80%	91.65%	500
Senior	85.40%	90.15%	1,000
White	81.60%	88.80%	1,000
Youth	80.40%	88.61%	1,000

The results corroborate our idea that, by introducing visual dictionaries, we achieve

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

significant improvements on the results in comparison to the result obtained in the state of the art [30]. In addition, we can note that the best-performing dictionary for each attribute has different vocabulary sizes. The reason is that the size of the regions, in some attributes, are different as Figure 3.1 depicts. Then, attributes with larger regions (e.g., male) have larger variations, so they need more visual words to be better represented.

Method #2: This method uses the dense-sampling approach following an idea proposed by [35] to extract low-level features. For each image in the dataset, we define a lattice of size 3×3 pixels across the image and extract all centers of the grid. Then, we use only the last two stages of the SURF [2] algorithm to characterize these points.

We constrain the descriptions of the points for each image according to the regions in Figure 3.1 and the describable attributes of interest. To build the visual dictionaries, we use a random selection process which selects random PoIs for representing the visual dictionaries of size 100, 500 and 1,000 words. This method is the first improvement that we have done with respect to the visual dictionaries creation and the results are depicted in Figure 5.3 and in Table 5.2.

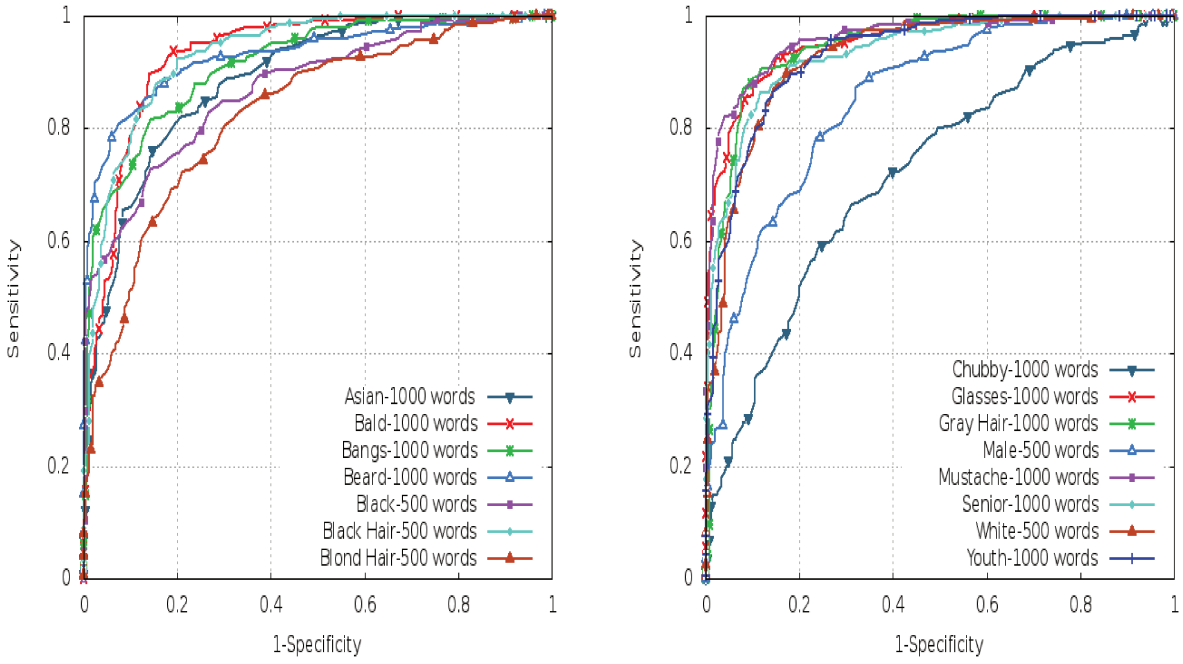


Figure 5.3: ROC curves obtained for the attribute classifiers using the Method #2.

Method #3: This method is similar to Method #2. In particular, we use a dense-sampling approach to extracting low-level features or points of interest in the images but, in this case, we use the clustering algorithm K-Means to select the most discriminative points of interest for each describable attribute and build the visual dictionaries. With this method, we managed to push the classification accuracies for some attributes even further. Using this method, we obtain an improved classification accuracy over the Methods #1 and #2, as well as over [30]. The results of this method are depicted in Figure 5.4 and in Table 5.2.

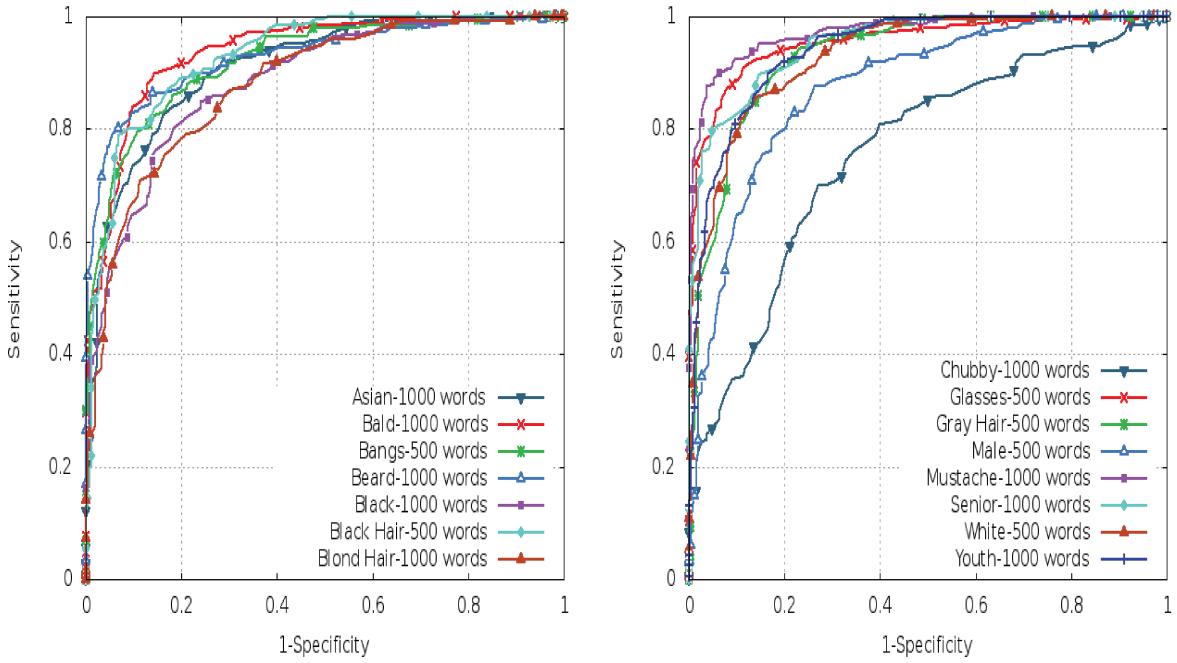


Figure 5.4: ROC curves obtained for the attribute classifiers using the Method #3.

In Table 5.2, we summarize the results obtained by using Methods #2 and #3. In order to compare our methods with [28], we also considered a characterization method based on the HoG (which is the most common method used to represent visual attributes) as well as SASI [5], one of the top-performers texture descriptors according to a recent survey [41]. Table 5.2 also brings such results.

Table 5.2: Accuracy and AUC for each visual attribute. Although the feature vector lengths of Methods #2, #3 and HoG vary, SASI feature vector is always of 64-d as this was the best vector length reported in [41].

	Method #2			Method #3			HoG			SASI	
Attribute	# Words	AUC	Accuracy	# Words	AUC	Accuracy	Vector	AUC	Accuracy	AUC	Accuracy
Asian	1,000	88.61%	80.60%	1,000	90.95%	**83.00%**	540	89.65%	80.60%	69.71%	64.60%
Bald	1,000	93.11%	87.00%	1,000	93.85%	86.80%	450	95.18%	**89.60%**	91.51%	84.00%
Bangs	1,000	91.36%	81.60%	500	92.14%	**83.20%**	450	93.36%	83.10%	86.46%	78.40%
Beard	1,000	92.90%	86.40%	1,000	92.46%	86.20%	432	93.65%	**89.40%**	87.26%	78.60%
Black	500	87.07%	78.20%	1,000	88.50%	**80.60%**	540	85.85%	79.00%	71.84%	66.20%
Black Hair	500	93.37%	**85.60%**	500	92.97%	84.40%	450	85.25%	78.60%	90.53%	82.00%
Blond Hair	500	82.67%	74.20%	1,000	87.90%	78.00%	450	88.51%	79.40%	91.10%	**84.60%**
Chubby	1,000	71.65%	67.40%	1,000	75.71%	71.40%	540	79.07%	**72.20%**	63.86%	59.60%
Glasses	1,000	94.65%	88.60%	500	95.55%	**88.60%**	567	89.15%	82.20%	87.19%	78.60%
Gray Hair	1,000	94.61%	**89.40%**	500	93.23%	85.40%	450	93.01%	84.80%	91.57%	84.00%
Male	500	85.36%	77.20%	500	87.48%	79.80%	540	89.87%	**82.00%**	78.86%	71.60%
Mustache	1,000	95.92%	89.20%	1,000	97.14%	**91.40%**	144	94.92%	87.80%	88.73%	80.20%
Senior	1,000	92.82%	**87.40%**	1,000	95.25%	86.40%	540	94.14%	84.20%	83.14%	75.40%
White	500	92.54%	**85.60%**	500	93.15%	85.20%	540	93.55%	81.20%	81.57%	72.40%
Youth	1,000	92.43%	86.00%	1,000	94.14%	**86.20%**	540	94.12%	80.20%	83.47%	74.40%

In Figure 5.5, we show a comparison among the three proposed methods, HoG, and SASI for the different attributes.

Discussion Dictionary sizes of 500 and 1,000 words are enough for a good image representation regardless the used method. In addition, note that except for the attribute *Chubby*, all other attributes present classification results (area under the curve) above 80%. For the 15 considered attributes, Dense Sampling-based methods (Methods #2 and #3) are more effective than either a HoG- or SASI-based solution in 10. Dense Sampling followed by Clustering (Methods #3) is more appropriate than all other methods in six out of 15 attributes.

More interesting than just counting which cases one method is better than other, is the analysis of the reasons for such. Confronting Dense Sampling-based solutions (Methods #2 and #3) and HoG/SASI, we clearly see that HoG is more interesting for *Bald*, *Beard*, *Blond Hair* and *Male* attributes. Interestingly, all of the cases here involve regions of high-texture/high-gradient changes which is where HoG descriptor theoretically would be more adequate (histogram of oriented gradients).

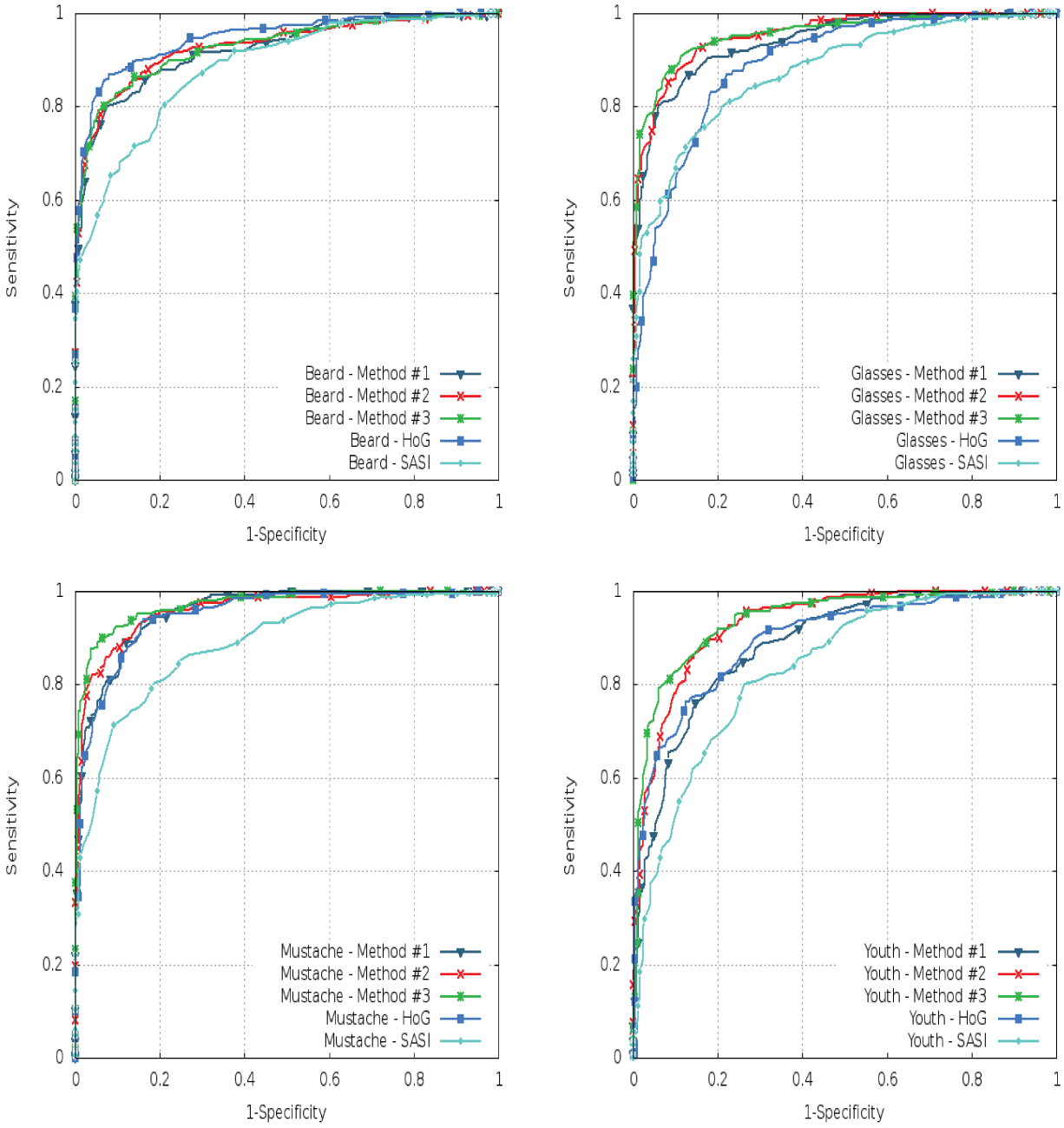


Figure 5.5: ROC curves obtained for the attribute classifiers using the three proposed methods Methods #1, #2, and #3 and the one based on histogram of oriented gradients (HoG) similar to the one used in [30] and SASI for four attributes of interest. As discussed, for some high-texture areas HoG outperforms the other methods (e.g., Beard Attribute on top-left corner) while Method #3 prevails in most of the other situations.

Performing a more elaborate analysis of this behavior, we devised two simple experiments (A and B). First, we calculated the co-occurrence matrices of all attributes and calculated the average entropy for each one [20]. One co-occurrence matrix is calculated for all the regions of a given attribute and the average entropy is calculated summing up all entropies (one for each co-occurrence matrix) and dividing by the number of images considered in the experiment.

The idea is that the higher the entropy, the higher the information gain on a given region and, therefore, more richness of details, an indication of a region with higher levels of texture. We base our idea on previous studies in the literature that associate high-entropy regions to textures [43, 22]. Second, similarly, we calculate gradient maps using the traditional Sobel operator [19][2] on the regions of interest and measure the average magnitudes of the gradients in such regions. The idea is that the higher the changes the more gradient exists in such regions.

Although Experiment A shows that for some cases, indeed, HoG goes well (e.g., bald = 0.84 average entropy, beard = 0.88 average entropy, blond hair = 0.83 average entropy), there is no final rule as Method #3 shines in some attribute regions as well (bangs = 0.85 average entropy). Experiment B also corroborates that there is no universal rule for when Method #3 outperforms both HoG and SASI (or vice-versa). For instance, the two attributes with the highest gradient changes are glasses and mustache and here HoG should go well as it is specially designed for capturing gradient changes. In both cases, however, Method #3 outperforms HoG and SASI.

In face of the two experiments, we conclude that the difference in performance is probably in the characterization power of Method #3 which performs a local analysis of the region (PoI characterization) followed by a generalization of the analysis (mid-level features through visual dictionaries). By sampling the lattice regularly and finding the most discriminant features by means of clustering, Method #3 seems to capture the best of both worlds for most features while HoG and SASI directly describes the regions globally.

The question mark is regarding the attribute *Chubby* which theoretically could favor any of the methods. However, this might be an artifact of the dataset used in which it is possible that there are more images of chubby people with any of the other attributes HoG shined (the average entropy for chubby = 0.95).

The dense sampling method is more appropriate in all other cases probably because there are more points of interest to better capturing the variations within the regions of interest. For the attribute *Black Hair*, for instance, the difference between Method #3 and HoG is about six percentage points. Interestingly, this attribute also shows high gradient but the dense sampling probably captures additional nuances that simple histogram of

[2]Similar results are obtained with other gradient detectors such as Canny [19].

orientations could not capture (as a matter of fact, the average entropy of black hair regions = 0.86, similar to bangs or blond hair). The same difference happens for *Glasses* and *Youth* attributes. Glasses show high difference in gradients (similar to mustache according to the experiment we described above) while youth shows very low gradient changes. In both cases, Methods #2 and #3 go well.

It is worth noting that as the analysis of each describable attribute is performed independently, nothing keeps us from using the most appropriate description approach for each attribute. We could, for instance, use Method #1 for describing the attribute *Black Hair* while using HoG for attributes like *Male* and *Bald*, Method #3 for attributes like *Glasses* and *Youth* or even SASI for attribute as *blond hair*. As everything is based on training examples, this task could be done automatically based on a small validation set of queries and simple verification algorithms (average entropy/texture + gradient analysis as we devised earlier).

5.2.2 Multilevel

We evaluate our multilevel approach in accordance to Section 3.2:

- Extract the regions according to each attribute (Figure 3.1),
- Scale the regions in the six levels,
- For each region at each level:
  - Describe the images using the sparse-sampling method,
  - Build the histogram according to the best dictionary (we use the same dictionary obtained on the single level),
  - Normalize the histogram,
  - Classify the image using the best model previously created (we use the same model evaluated on the single level)

In Table 5.3, we show the accuracies and AUCs obtained for each level in the testing set of LFW dataset. Thereafter, we combine the different levels using the methods that we propose in Section 3.2. For the testing set of LFW dataset we only evaluate three methods: Majority Voting, Majority Voting Best and Weighted Fusion. The accuracies and AUCs for these methods are depicted in Table 5.4.

Table 5.3: Accuracies and AUCs for each visual attribute, evaluated in the six levels.

Attribute Classifier	**LFW-TEST** (250 + 250−)											
	Accuracy						**Area under Curve (AUC)**					
	Level 1 (0.5×)	Level 2 (0.75×)	Level 3 (1×)	Level 4 (1.5×)	Level 5 (2×)	Level 6 (2.5×)	Level 1 (0.5×)	Level 2 (0.75×)	Level 3 (1×)	Level 4 (1.5×)	Level 5 (2×)	Level 6 (2.5×)
Asian	60.00%	73.60%	**76.80%**	72.80%	73.00%	73.60%	82.55%	83.23%	**86.15%**	81.80%	82.29%	84.03%
Bald	50.20%	74.40%	**81.40%**	78.20%	78.40%	76.60%	62.93%	85.06%	**88.68%**	88.05%	88.06%	87.26%
Bangs	53.80%	64.40%	**71.60%**	69.20%	68.20%	67.60%	66.94%	71.06%	**78.47%**	76.43%	75.23%	75.67%
Beard	51.00%	66.80%	**79.00%**	76.20%	77.80%	77.60%	55.57%	78.13%	85.73%	84.67%	**86.02%**	85.61%
Black	49.80%	73.80%	**78.20%**	75.40%	76.80%	73.80%	74.45%	84.29%	85.60%	84.47%	**86.04%**	82.99%
Black Hair	63.20%	71.60%	76.80%	**78.20%**	77.20%	76.60%	68.96%	78.81%	84.09%	**85.27%**	83.54%	84.04%
Blond Hair	55.40%	62.60%	**65.40%**	64.60%	64.80%	63.40%	60.43%	69.36%	**74.14%**	71.04%	71.70%	70.81%
Chubby	52.80%	**63.20%**	61.20%	61.60%	61.60%	61.20%	61.52%	67.05%	67.19%	67.08%	**68.66%**	66.50%
Glasses	50.00%	72.40%	**76.40%**	72.00%	72.00%	70.00%	50.00%	78.54%	84.73%	84.00%	**85.75%**	83.73%
Gray Hair	63.80%	72.60%	76.60%	76.00%	76.20%	**76.80%**	75.19%	81.35%	**84.89%**	84.34%	84.70%	83.60%
Male	74.40%	80.60%	**83.40%**	80.40%	81.80%	82.00%	86.81%	89.65%	**92.04%**	89.04%	90.57%	91.46%
Mustache	50.00%	50.00%	**77.60%**	74.80%	68.80%	71.80%	50.00%	50.00%	**87.77%**	84.96%	83.80%	84.84%
Senior	51.40%	77.40%	**81.80%**	78.80%	81.00%	78.00%	76.29%	86.75%	89.58%	88.64%	**90.82%**	89.39%
White	73.40%	77.40%	**84.40%**	80.20%	82.00%	81.20%	85.11%	88.29%	**92.13%**	88.35%	91.25%	90.20%
Youth	62.40%	68.80%	75.40%	75.80%	**77.20%**	75.20%	69.82%	77.23%	**84.76%**	83.92%	84.55%	83.50%

Table 5.4: Accuracies and AUCs using MV, MVB and WF to combine the levels

Attribute Classifier	**LFW-TEST** (250 + 250−)					
	Accuracy			**Area under Curve (AUC)**		
	Majority Voting	Majority Voting Best	Weighted Fusion	Majority Voting	Majority Voting Best	Weighted Fusion
Asian	78.00%	78.40%	**79.20%**	84.78%	**87.16%**	85.18%
Bald	81.00%	82.60%	**82.60%**	88.60%	88.67%	**89.09%**
Bangs	70.60%	71.60%	**72.60%**	77.46%	77.17%	**79.34%**
Beard	80.20%	80.60%	**82.60%**	86.89%	86.90%	**88.22%**
Black	78.20%	**79.00%**	78.60%	86.63%	87.56%	**88.08%**
Black Hair	**79.60%**	78.00%	78.80%	85.98%	83.45%	**87.36%**
Blond Hair	**64.40%**	64.00%	63.40%	**73.45%**	71.45%	73.05%
Chubby	63.40%	**65.00%**	64.80%	68.15%	69.18%	**70.25%**
Glasses	76.00%	77.80%	**78.00%**	87.02%	86.64%	**88.35%**
Gray Hair	77.00%	77.60%	**77.80%**	84.94%	**85.70%**	85.63%
Male	85.80%	**86.80%**	86.40%	93.27%	93.23%	**94.48%**
Mustache	**78.00%**	76.60%	77.60%	87.07%	86.87%	**87.45%**
Senior	84.40%	84.20%	**86.40%**	90.66%	91.82%	**92.16%**
White	85.80%	**87.60%**	87.40%	92.60%	**94.51%**	93.59%
Youth	**78.20%**	76.60%	77.20%	**85.98%**	82.72%	85.94%

After this, we decided to perform our multilevel approach in the PubFig dataset effectively testing on a cross-dataset condition. To evaluate the accuracies in the six levels, we randomly selected, for each attribute, 250 positive images and 250 negative images from PubFig. In this dataset, we evaluated 14 attributes because this dataset has very few images of asian people, so we do not evaluate the attribute "asian". Table 5.5 shows the accuracies and AUCs obtained for each level in PubFig.

Table 5.5: Accuracies and AUCs for each visual attribute, evaluated in the six levels using the random images from PubFig.

Attribute Classifier	PUBFIG – RANDOM IMAGES (250 + 250−)											
	Accuracy						Area under Curve (AUC)					
	Level 1 (0.5×)	Level 2 (0.75×)	Level 3 (1×)	Level 4 (1.5×)	Level 5 (2×)	Level 6 (2.5×)	Level 1 (0.5×)	Level 2 (0.75×)	Level 3 (1×)	Level 4 (1.5×)	Level 5 (2×)	Level 6 (2.5×)
Bald	50.00%	50.80%	72.60%	**83.60%**	83.20%	80.20%	50.00%	46.93%	84.21%	91.97%	**92.64%**	92.57%
Bangs	50.00%	70.40%	**85.20%**	85.00%	82.60%	84.00%	50.00%	82.54%	91.56%	**94.54%**	94.08%	93.81%
Beard	50.00%	50.60%	78.20%	**86.40%**	82.60%	82.60%	50.00%	58.35%	87.71%	**94.49%**	94.19%	94.13%
Black	50.00%	77.60%	**85.20%**	83.20%	82.60%	81.60%	80.00%	87.36%	92.18%	**92.73%**	92.45%	92.36%
Black Hair	50.00%	55.40%	71.40%	79.20%	79.40%	**81.00%**	50.00%	67.89%	79.27%	87.99%	86.85%	**88.96%**
Blond Hair	50.00%	60.60%	74.20%	74.80%	**74.80%**	73.80%	50.00%	65.08%	**83.81%**	81.42%	81.87%	81.86%
Chubby	52.80%	58.60%	**64.00%**	63.20%	63.40%	63.40%	61.59%	61.85%	71.37%	70.70%	**72.19%**	69.63%
Glasses	52.00%	68.40%	81.80%	80.40%	**84.80%**	83.80%	60.90%	76.69%	90.02%	93.16%	**94.99%**	94.27%
Gray Hair	50.00%	61.20%	70.80%	78.60%	**81.00%**	80.00%	50.00%	72.20%	81.33%	88.23%	**89.90%**	88.69%
Male	71.20%	81.60%	**85.80%**	82.60%	82.00%	81.20%	83.18%	91.16%	94.27%	**94.44%**	93.61%	92.69%
Mustache	50.00%	50.00%	72.00%	81.20%	79.20%	**82.20%**	50.00%	50.00%	83.68%	89.65%	88.53%	**90.60%**
Senior	51.80%	78.60%	**83.80%**	83.40%	81.80%	79.80%	77.59%	88.73%	92.63%	94.05%	**94.14%**	93.13%
White	77.80%	83.00%	85.20%	**87.00%**	85.40%	84.60%	86.92%	93.09%	94.67%	**95.14%**	94.49%	93.73%
Youth	60.80%	76.80%	82.40%	**84.40%**	82.60%	83.60%	67.29%	84.37%	89.22%	**92.63%**	91.95%	91.86%

After testing for attributes in isolation for each level, we combine the different levels using the six methods proposed in section 3.2. The accuracies and AUCs for these methods are presented in Table 5.6. In this case, to evaluate MetaFusion we used the confidences obtained by the attribute classifiers in the LFW dataset. For the case of WeightedFusion, the proper weights giving the right importance for each classifier are also learned from in LFW.

Table 5.6: Accuracies and AUCs using MV, MVB, WF, MF and AF to combine the levels using PubFig.

Attribute Classifier	PUBFIG – RANDOM IMAGES (250 + 250−)									
	Accuracy					Area under Curve (AUC)				
	Majority Voting	Majority Voting Best	Weighted Fusion	Meta Fusion	Average Fusion	Majority Voting	Majority Voting Best	Weighted Fusion	Meta Fusion	Average Fusion
Bald	72.20%	86.20%	**86.40%**	85.60%	86.00%	81.86%	91.89%	**92.19%**	88.29%	92.19%
Bangs	89.40%	89.00%	89.80%	86.80%	**90.20%**	94.70%	95.18%	95.28%	93.67%	**95.32%**
Beard	89.40%	89.20%	90.40%	89.00%	**90.60%**	**95.30%**	94.36%	94.86%	93.83%	94.84%
Black	86.40%	85.80%	86.60%	**86.80%**	86.60%	92.88%	**93.95%**	93.55%	93.26%	93.52%
Black Hair	79.20%	78.60%	78.80%	**81.00%**	78.20%	89.02%	88.15%	**90.15%**	90.01%	90.05%
Blond Hair	**77.40%**	74.60%	75.80%	76.40%	76.60%	82.89%	80.99%	83.25%	**85.14%**	83.14%
Chubby	63.20%	65.00%	**65.40%**	63.00%	65.40%	72.24%	72.45%	**72.90%**	67.25%	72.89%
Glasses	**90.00%**	85.20%	89.60%	89.20%	89.20%	94.72%	85.41%	95.11%	93.01%	**95.14%**
Gray Hair	79.00%	78.60%	**79.00%**	78.00%	79.00%	86.75%	**88.80%**	88.03%	85.39%	87.97%
Male	**88.60%**	87.00%	88.00%	88.40%	87.60%	96.31%	96.50%	96.57%	**96.62%**	96.52%
Mustache	**85.20%**	82.00%	83.80%	81.40%	84.40%	90.17%	89.59%	91.86%	90.49%	**91.95%**
Senior	86.40%	86.20%	87.20%	86.60%	**88.40%**	94.76%	93.25%	**95.23%**	93.37%	95.23%
White	86.60%	**89.20%**	89.00%	86.20%	89.00%	95.11%	**97.38%**	96.42%	95.92%	96.38%
Youth	**85.40%**	82.40%	84.00%	85.00%	83.20%	92.16%	88.23%	92.36%	**92.89%**	92.00%

5.3 Results for Complex Queries

This section shows results for rank fusion techniques using the methods previously explained in Chapter 4. With k attributes, we have 2^k possible queries. We have considered $k = 15$ attribute classifiers and, consequently, their direct counterparts theoretically allowing $2^{15} = 32,768$ different combined queries. However, the actual number is smaller since some attributes are contradictory. Similar to the results for simple queries, we performed our experiments for both single level and multilevel approaches. Here, we present the results for a subset of all possible queries in LFW and PubFig datasets for single level. For multilevel, we evaluate in PubFig dataset as this is a more difficult and more complete setup [6].

5.3.1 Single Level

Score normalization is a fundamental task when dealing with discrepant values related to different attribute classifiers and is paramount before fusion. To overcome this problem, we normalize the scores using the traditional *z-norm* (subtract the overall mean score and divide by the overall standard deviation). To measure the effectiveness of each original rank and each rank resulting of a fusion, we assess the number of relevant images within the retrieved images for a fixed recall. Table 5.7 shows the precision for a subset of queries. In this case, we use the traditional rank aggregation explained in section 4.3.

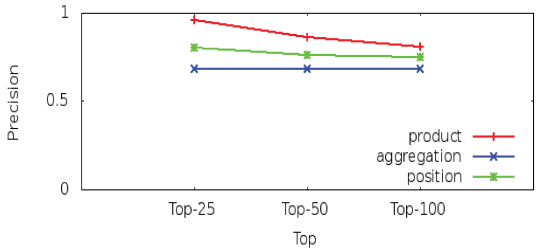
Queries with a single attribute normally have high precision results as Table 5.7 shows. An exception is noted for the attribute *asian*, due to the fact that LFW does not have a significant number of images of asian people for training. Although the database contains an acceptable number of asian images, the number of asian people is small when we remove the training set part. However, as we would expect, the non-*asian* precision is 100%.

We note that the traditional rank aggregation method does not yield good results for searches with two attributes. The reason is that only one vote is enough to put an image in the resulting rank. The precision of traditional rank aggregation, for most of the queries with two attributes, is approximately half of the precision of the product of probabilities approach. In searches with more than two attributes, however, the results are interesting with the three approaches disputing the lead. In some cases, like $Q = \{ma, ba, se\}$, traditional rank aggregation presents a huge difference in comparison with the product of probabilities as Table 5.7 shows along with other complex queries. For complex queries like $Q = \{wh, ma, se, ba, mu, gl\}$, rank position provides the highest accuracy. In general, we observed that for two-attribute queries product of probabilities are more adequate while for three traditional rank aggregation shows more interesting and after that there is no clear advantage to any of the three.

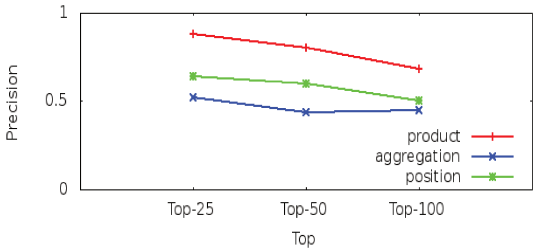
In order to measure the precision of our approach, we have analyzed the results returned in the top positions as Figure 5.6 depicts. As a result of such analysis, we have observed that the higher precision is obtained in the top 25 positions and it decreases as we analyze the next top positions. Maximizing the number of relevant results in the first positions represents an important advantage to ensure the quality of the retrieved results.

Table 5.7: Precision (%) of some selected complex queries in LFW dataset using the single level approach. We also show some selected queries with just one attribute (no fusion used) for reference.

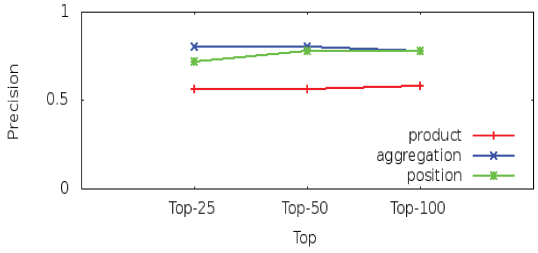
Q	Top-25			Top-50			Top-100		
	$F_{product}$	$F_{aggregation}$	$F_{position}$	$F_{product}$	$F_{aggregation}$	$F_{position}$	$F_{product}$	$F_{aggregation}$	$F_{position}$
$\{ma\}$	96.0	96.0	96.0	98.0	98.0	98.0	99.0	99.0	99.0
$\{gl\}$	100.0	100.0	100.0	100.0	100.0	100.0	99.0	99.0	99.0
$\{be\}$	80.0	80.0	80.0	66.0	66.0	66.0	62.0	62.0	62.0
$\{mu\}$	96.0	96.0	96.0	86.0	86.0	86.0	80.0	80.0	80.0
$\{\overline{be}\}$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$\{\overline{mu}\}$	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
$\{\overline{as}\}$	100.0	100.0	100.0	100.0	100.0	100.0	99.0	99.0	99.0
$\{gl, as\}$	**52.0**	36.0	44.0	**44.0**	36.0	40.0	42.0	41.0	**43.0**
$\{gl, be\}$	**52.0**	32.0	36.0	**40.0**	22.0	28.0	**26.0**	17.0	20.0
$\{mu, be\}$	**72.0**	64.0	68.0	60.0	**70.0**	64.0	55.0	51.0	55.0
$\{\overline{ma}, as\}$	**24.0**	16.0	16.0	**20.0**	16.0	18.0	**26.0**	15.0	17.0
$\{\overline{ma}, gl\}$	**44.0**	8.0	16.0	**32.0**	16.0	26.0	**24.0**	5.0	10.0
$\{\overline{ma}, \overline{gl}\}$	100.0	96.0	100.0	**98.0**	90.0	96.0	**96.0**	80.0	85.0
$\{mu, \overline{be}\}$	12.0	24.0	**28.0**	16.0	22.0	**24.0**	14.0	**16.0**	15.0
$\{ma, ba, se\}$	56.0	**80.0**	72.0	56.0	**80.0**	78.0	58.0	78.0	78.0
$\{\overline{ma}, bl, bah\}$	20.0	**32.0**	28.0	26.0	**34.0**	30.0	30.0	33.0	**34.0**
$\{ma, bl, bah\}$	**16.0**	8.0	8.0	**18.0**	10.0	8.0	**15.0**	11.0	10.0
$\{ma, se, wh\}$	36.0	40.0	**48.0**	38.0	42.0	**46.0**	40.0	39.0	**41.0**
$\{ma, \overline{bg}, se\}$	32.0	**36.0**	28.0	28.0	**30.0**	26.0	27.0	**33.0**	22.0
$\{ma, mu, se\}$	40.0	**44.0**	36.0	36.0	**40.0**	38.0	35.0	**36.0**	31.0
$\{\overline{ma}, boh, yo\}$	**72.0**	56.0	36.0	**68.0**	50.0	32.0	**61.0**	47.0	28.0
$\{\overline{ma}, bg, yo\}$	16.0	12.0	16.0	**18.0**	10.0	14.0	**15.0**	11.0	12.0
$\{\overline{mu}, \overline{be}, as, \overline{ba}\}$	**28.0**	12.0	16.0	18.0	20.0	**22.0**	16.0	18.0	**20.0**
$\{gl, \overline{mu}, \overline{be}, as\}$	**16.0**	8.0	12.0	**14.0**	10.0	11.0	**12.0**	9.0	11.0
$\{ma, \overline{gl}, as, \overline{ba}\}$	**12.0**	8.0	8.0	10.0	**12.0**	10.0	8.0	**9.0**	7.0
$\{ma, gl, mu, \overline{ba}\}$	**28.0**	16.0	20.0	**24.0**	18.0	22.0	21.0	23.0	**24.0**
$\{ma, mu, \overline{be}, \overline{ba}\}$	8.0	8.0	4.0	6.0	8.0	6.0	**9.0**	7.0	5.0
$\{\overline{gl}, \overline{mu}, \overline{be}, \overline{ba}\}$	28.0	**88.0**	72.0	22.0	**80.0**	68.0	21.0	**81.0**	65.0
$\{ma, \overline{gl}, mu, yo, \overline{ba}\}$	**16.0**	12.0	8.0	**14.0**	12.0	12.0	**13.0**	10.0	12.0
$\{ma, gl, mu, be, \overline{ba}\}$	8.0	**12.0**	4.0	6.0	**10.0**	6.0	7.0	**8.0**	7.0
$\{wh, ch, ma, gl, bg\}$	**32.0**	20.0	16.0	**36.0**	22.0	14.0	**31.0**	19.0	13.0
$\{\overline{gl}, \overline{mu}, \overline{be}, \overline{as}, \overline{ba}\}$	**100.0**	92.0	96.0	**98.0**	90.0	94.0	**97.0**	93.0	95.0
$\{ma, bg, mu, be, as, yo\}$	**12.0**	8.0	8.0	6.0	6.0	**8.0**	5.0	**6.0**	4.0
$\{wh, ma, se, ba, mu, gl\}$	20.0	24.0	**28.0**	18.0	20.0	**24.0**	16.0	19.0	**21.0**
$\{ma, \overline{gl}, \overline{mu}, \overline{be}, as, \overline{ba}\}$	**24.0**	12.0	16.0	**18.0**	10.0	14.0	**15.0**	8.0	11.0



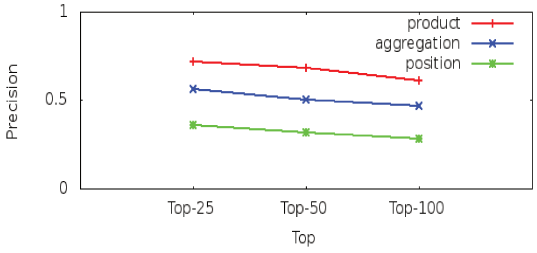
(a) $Q = \{ma, gl\}$



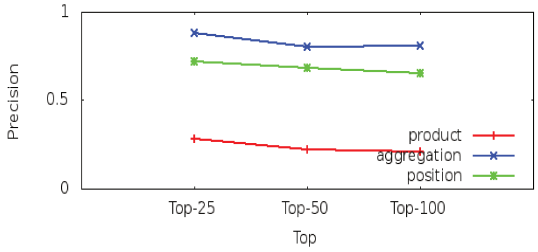
(b) $Q = \{gl, ba\}$



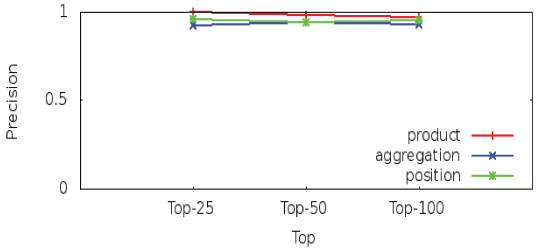
(c) $Q = \{ma, ba, se\}$



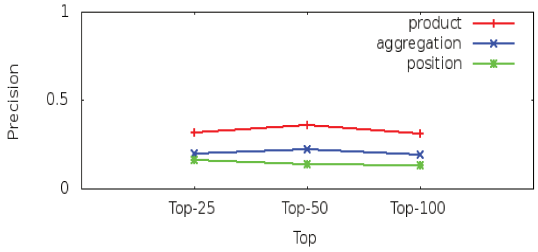
(d) $Q = \{\overline{ma}, boh, yo\}$



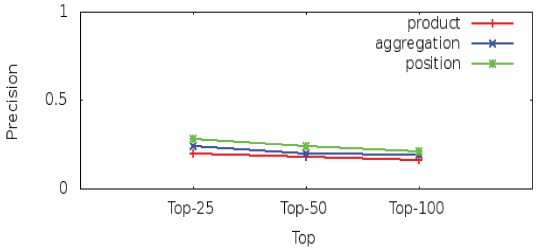
(e) $Q = \{\overline{gl}, \overline{mu}, \overline{be}, \overline{ba}\}$



(f) $Q = \{\overline{gl}, \overline{mu}, \overline{be}, \overline{as}, \overline{ba}\}$



(g) $Q = \{wh, ch, ma, gl, bg\}$



(h) $Q = \{wh, ma, se, ba, mu, gl\}$

Figure 5.6: Product of Probabilities vs. Traditional Rank Aggregation vs. Rank Position approaches. Some selected queries in LFW dataset. Note that for the query we mentioned in the Abstract, *white chubby male* wearing *glasses* and with *bangs* (lower left corner), we have a precision of 32% for recall of 25 (8 out of 25) using product of probabilities.

For complex queries in PubFig dataset, we evaluate product of probabilities (MethodC1) and the modified rank aggregation (MethodC2) explained in section 4.3. Table 5.8 shows the precisions obtained using these methods in the PubFig dataset for some selected queries.

Table 5.8: Precision (%) of some selected queries in PubFig dataset using the single level approach.

Q	Top-25		Top-50		Top-100	
	MethodC1	MethodC2	MethodC1	MethodC2	MethodC1	MethodC2
$\{ma, be\}$	92.0	92.0	84.0	86.0	76.0	77.0
$\{ma, bl\}$	84.0	84.0	74.0	78.0	55.0	55.0
$\{ma, se\}$	88.0	80.0	88.0	86.0	85.0	85.0
$\{ma, wh\}$	68.0	68.0	78.0	76.0	81.0	81.0
$\{mu, be\}$	88.0	88.0	84.0	84.0	78.0	76.0
$\{\overline{ma}, wh\}$	92.0	92.0	94.0	96.0	95.0	98.0
$\{\overline{ma}, yo\}$	96.0	96.0	92.0	92.0	94.0	92.0
$\{\overline{ma}, \overline{gl}\}$	92.0	92.0	96.0	96.0	97.0	96.0
$\{ma, se, wh\}$	84.0	80.0	76.0	76.0	80.0	82.0
$\{ma, gl, se\}$	96.0	96.0	90.0	92.0	90.0	88.0
$\{ma, mu, se\}$	24.0	24.0	28.0	28.0	32.0	31.0
$\{ma, ba, se\}$	72.0	72.0	64.0	70.0	56.0	65.0
$\{ma, be, mu\}$	92.0	96.0	92.0	94.0	76.0	82.0
$\{\overline{ma}, boh, yo\}$	52.0	56.0	48.0	52.0	55.0	54.0
$\{\overline{ma}, bg, yo\}$	76.0	72.0	70.0	68.0	69.0	69.0
$\{ma, \overline{bg}, se\}$	84.0	88.0	86.0	86.0	87.0	84.0
$\{\overline{ma}, bl, bah\}$	56.0	48.0	36.0	40.0	29.0	32.0
$\{ma, gl, mu, be\}$	68.0	72.0	52.0	52.0	51.0	53.0
$\{ma, be, mu, wh\}$	84.0	80.0	76.0	76.0	76.0	72.0
$\{ma, wh, se, be\}$	44.0	40.0	42.0	42.0	42.0	42.0
$\{\overline{ma}, boh, bg, wh\}$	80.0	84.0	78.0	78.0	72.0	75.0
$\{ma, \overline{mu}, \overline{be}, \overline{ba}\}$	80.0	72.0	80.0	74.0	69.0	75.0
$\{\overline{gl}, \overline{mu}, \overline{be}, \overline{ba}\}$	92.0	88.0	94.0	92.0	95.0	92.0
$\{ma, \overline{gl}, mu, yo, \overline{ba}\}$	80.0	76.0	70.0	68.0	61.0	54.0
$\{ma, gl, mu, be, \overline{ba}\}$	36.0	40.0	28.0	26.0	21.0	18.0
$\{\overline{ma}, wh, bg, boh, yo\}$	76.0	76.0	66.0	72.0	68.0	57.0
$\{ma, gl, ba, se, wh\}$	52.0	52.0	36.0	42.0	37.0	37.0

5.3.2 Multilevel

In PubFig dataset, we also considered the multilevel approach to solving complex queries. In this case, we use the weighted fusion method for combining the different levels, thereby generating one ranked list based on the six levels. For example, for the query $Q = \{ma, gl\}$, we use weighted fusion to generate two ranked lists, one for the attribute *"male"* and other for the attribute *"glasses"*. After that, to combine these ranked lists, we proposed the following four methods based on the fusion techniques explained in Chapter 4:

MethodC3: To combine the ranked lists, this method uses the product of probabilities to multiply the scores in each one of the ranked lists.

MethodC4: This method use the modified rank aggregation to combine the ranked lists.

MethodC5: In this method, for each attribute in the query, we first add the scores of the weighted fusion list and the scores in the single level. Then, we use the modified rank aggregation to combine the attributes in the complex query.

MethodC6: In this method, for each attribute in the query, we first add the scores of each one of the six levels. Then, we use the modified rank aggregation to fusion the attributes in the complex query.

Table 5.9 depicts the precisions obtained using the explained methods above for some selected queries in PubFig dataset. The precisions obtained in this dataset are higher compared to the precisions obtained in LFW dataset, this occurs mainly because PubFig has many more images than LFW. In some queries ($\{mu, be\}$, $\{ma, be, mu\}$, etc.) in Table 5.9 we obtained 100% of precision in Top-25, this is an important advantage, because in the face search engines we must ensure that the images in the top positions meet the attributes present in the query.

In Table 5.9, we note that by using multilevel approach, apparently we obtained similar precisions in all methods. Thereafter, for a better understanding of the experiments, we decided to analyze statistically the results. For this, we used all the precisions obtained for each complex query in Table 5.9. In addition, we analyzed Table 5.9 results in conjunction with Table 5.8 to demonstrate statistically if the multilevel approach is different than the single level.

First, we used the well-known Analysis of Variance (ANOVA) to determine if our methods represent a statistically significant factor. For this, performed an ANOVA test

over the obtained results. According to [14], we define the null hipothesis H_0: all the methods means are equal. Table 5.10 shows the result obtained by using ANOVA. If the p value is lower than 0.05, then the hypothesis H_0 is rejected. In this case, $p = 2.2e-16$, it means that all the method means are not equal.

Table 5.9: Precision (%) of some selected queries in PubFig dataset using the multilevel approach.

	Top-25				Top-50				Top-100			
Q	MethodC3	MethodC4	MethodC5	MethodC6	MethodC3	MethodC4	MethodC5	MethodC6	MethodC3	MethodC4	MethodC5	MethodC6
$\{ma, be\}$	92.0	92.0	92.0	88.0	88.0	82.0	90.0	76.0	79.0	82.0	84.0	80.0
$\{ma, bl\}$	84.0	80.0	80.0	84.0	80.0	78.0	78.0	80.0	70.0	70.0	79.0	71.0
$\{ma, se\}$	96.0	100.0	100.0	100.0	98.0	100.0	100.0	96.0	96.0	100.0	100.0	97.0
$\{ma, wh\}$	96.0	96.0	96.0	96.0	98.0	96.0	98.0	96.0	96.0	96.0	97.0	97.0
$\{mu, be\}$	100.0	96.0	88.0	96.0	94.0	90.0	92.0	90.0	94.0	92.0	93.0	92.0
$\{\overline{ma}, wh\}$	96.0	96.0	96.0	96.0	98.0	98.0	98.0	98.0	98.0	98.0	99.0	99.0
$\{\overline{ma}, yo\}$	88.0	88.0	88.0	84.0	92.0	92.0	94.0	86.0	96.0	95.0	94.0	89.0
$\{\overline{ma}, \overline{gl}\}$	100.0	96.0	100.0	96.0	98.0	98.0	98.0	98.0	98.0	98.0	99.0	98.0
$\{ma, se, wh\}$	88.0	92.0	84.0	92.0	90.0	90.0	84.0	90.0	91.0	91.0	88.0	88.0
$\{ma, gl, se\}$	92.0	96.0	96.0	88.0	88.0	88.0	96.0	84.0	87.0	89.0	93.0	86.0
$\{ma, mu, se\}$	72.0	68.0	76.0	68.0	70.0	72.0	68.0	66.0	63.0	66.0	67.0	66.0
$\{ma, ba, se\}$	84.0	96.0	96.0	84.0	80.0	86.0	86.0	82.0	74.0	89.0	84.0	81.0
$\{ma, be, mu\}$	100.0	96.0	100.0	96.0	98.0	96.0	100.0	96.0	96.0	97.0	100.0	95.0
$\{\overline{ma}, boh, yo\}$	68.0	64.0	64.0	60.0	56.0	60.0	56.0	50.0	53.0	51.0	50.0	51.0
$\{\overline{ma}, bg, yo\}$	72.0	64.0	68.0	64.0	60.0	64.0	60.0	60.0	56.0	60.0	60.0	57.0
$\{ma, \overline{bg}, se\}$	96.0	96.0	96.0	96.0	96.0	96.0	96.0	96.0	90.0	92.0	91.0	91.0
$\{\overline{ma}, bl, bah\}$	52.0	52.0	52.0	52.0	40.0	42.0	52.0	46.0	37.0	39.0	43.0	39.0
$\{ma, gl, mu, be\}$	84.0	88.0	88.0	88.0	70.0	72.0	88.0	74.0	70.0	69.0	63.0	68.0
$\{ma, be, mu, wh\}$	96.0	92.0	80.0	88.0	90.0	90.0	90.0	90.0	80.0	83.0	85.0	80.0
$\{ma, wh, se, be\}$	56.0	56.0	60.0	48.0	56.0	44.0	58.0	52.0	54.0	54.0	55.0	51.0
$\{\overline{ma}, boh, bg, wh\}$	84.0	88.0	76.0	80.0	78.0	84.0	84.0	82.0	80.0	85.0	82.0	84.0
$\{ma, \overline{mu}, \overline{be}, \overline{ba}\}$	92.0	92.0	88.0	84.0	90.0	90.0	88.0	86.0	85.0	88.0	88.0	88.0
$\{\overline{gl}, \overline{mu}, \overline{be}, \overline{ba}\}$	100.0	92.0	96.0	92.0	96.0	96.0	98.0	92.0	96.0	98.0	98.0	94.0
$\{ma, \overline{gl}, mu, yo, \overline{ba}\}$	76.0	88.0	80.0	80.0	72.0	72.0	72.0	68.0	62.0	66.0	65.0	56.0
$\{ma, gl, mu, be, \overline{ba}\}$	52.0	44.0	48.0	40.0	52.0	56.0	38.0	56.0	51.0	50.0	30.0	48.0
$\{\overline{ma}, wh, bg, boh, yo\}$	80.0	76.0	76.0	56.0	78.0	64.0	68.0	56.0	72.0	67.0	61.0	40.0
$\{ma, gl, ba, se, wh\}$	64.0	80.0	80.0	80.0	58.0	66.0	66.0	64.0	54.0	60.0	58.0	58.0

Table 5.10: Analysis of Variance (ANOVA) of our six proposed methods.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
Methods	5	0.9484	0.18968	39.567	2.2e-16

We also used a method for comparing multiple hypotheses. In this case, we use Tukey Honest Significant Differences (TukeyHSD) [17]. Table 5.11 shows the results obtained by using TukeyHSD, the p-values were calculated according to [17].

Table 5.11 shows that the comparisons: MethodC3 - MethodC1 ($p < 0.05$), MethodC4 - MethodC1 ($p < 0.05$), MethodC5 - MethodC1 ($p < 0.05$), MethodC6 - MethodC1 ($p < 0.05$), MethodC3 - MethodC2 ($p < 0.05$), MethodC4 - MethodC2 ($p < 0.05$), MethodC5 - MethodC2 ($p < 0.05$), MethodC6 - MethodC2 ($p < 0.05$) are statistically different. Interestingly, such comparisons are comparisons between methods using single level and methods using multilevel.

Finally, using this analysis and Tables 5.9 and 5.8, we can see that the multilevel approach is better than the single level method for characterizing images.

Table 5.11: Multiple comparisons analysis between our six proposed methods using TukeyHSD.

	diff	lwr	upr	p-value
MethodC2 - MethodC1	-0.001071429	-0.03164291	0.029500053	0.9999986
MethodC3 - MethodC1	0.093571429	0.06299995	0.124142910	0.0000000
MethodC4 - MethodC1	0.098452381	0.06788090	0.129023863	0.0000000
MethodC5 - MethodC1	0.094761905	0.06419042	0.125333386	0.0000000
MethodC6 - MethodC1	0.072142857	0.04157138	0.102714339	0.0000000
MethodC3 - MethodC2	0.094642857	0.06407138	0.125214339	0.0000000
MethodC4 - MethodC2	0.099523810	0.06895233	0.130095291	0.0000000
MethodC5 - MethodC2	0.095833333	0.06526185	0.126404815	0.0000000
MethodC6 - MethodC2	0.073214286	0.04264280	0.103785767	0.0000000
MethodC4 - MethodC3	0.004880952	-0.02569053	0.035452434	0.9974991
MethodC5 - MethodC3	0.001190476	-0.02938101	0.031761958	0.9999976
MethodC6 - MethodC3	-0.021428571	-0.05200005	0.009142910	0.3405295
MethodC5 - MethodC4	-0.003690476	-0.03426196	0.026881005	0.9993506
MethodC6 - MethodC4	-0.026309524	-0.05688101	0.004261958	0.1375164
MethodC6 - MethodC5	-0.022619048	-0.05319053	0.007952434	0.2801331

Figure 5.7 shows, graphically, the results in Table 5.11. The intervals not crossing the dashed line in 0.0 represent the methods that are statistically different. In this case, we use a 95% family-wise confidence level ($p < 0.05$).

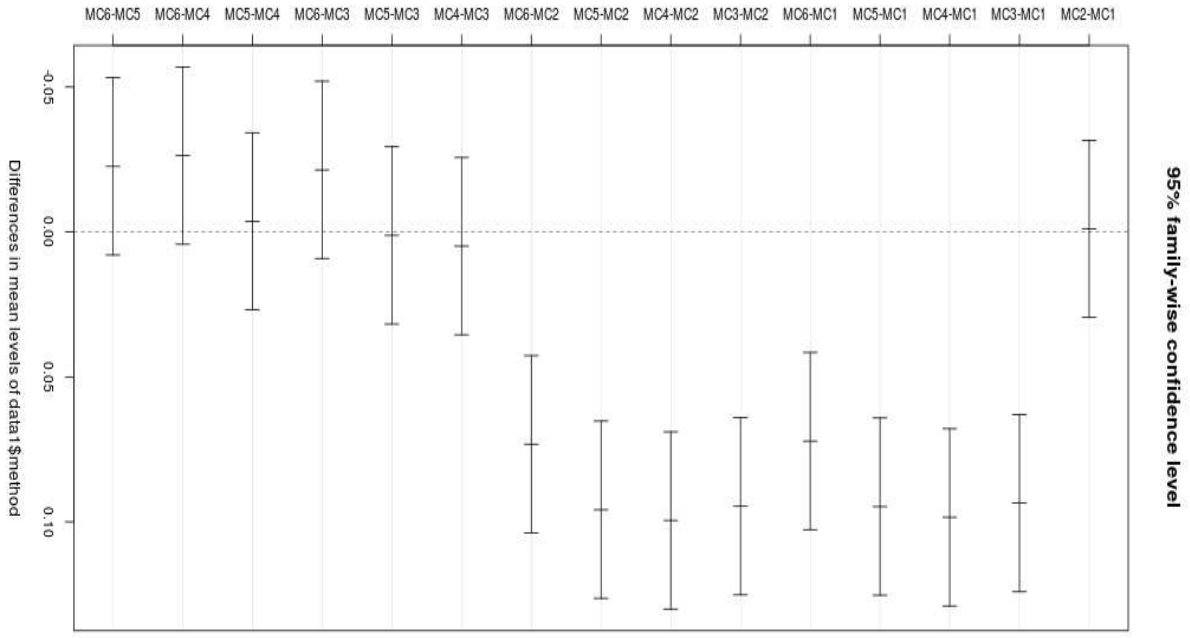
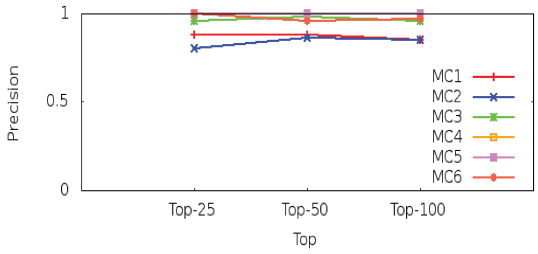
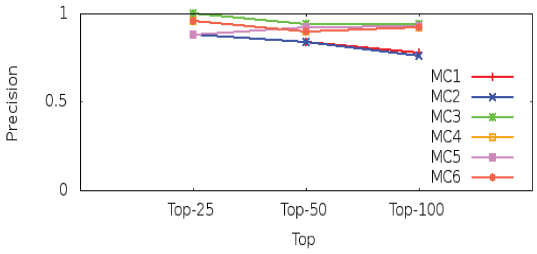


Figure 5.7: Multiple comparisons of means between our proposed methods using the TukeyHSD [17].

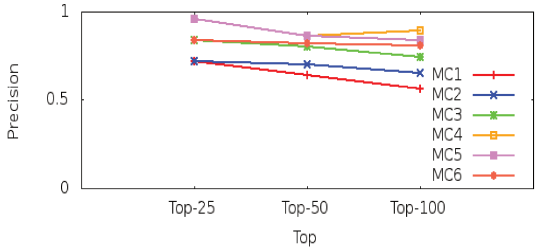
In addition, we measure the precision of our methods in the PubFig dataset, Figure 5.8 shows the results returned in the top positions for each one of our methods evaluated in PubFig. As well as in our experiments in LFW, in PubFig we also obtained the higher precision in the top 25 positions, ensuring the quality of the retrieved results.



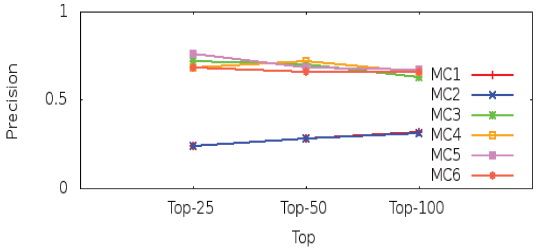
(a) $Q = \{ma, se\}$



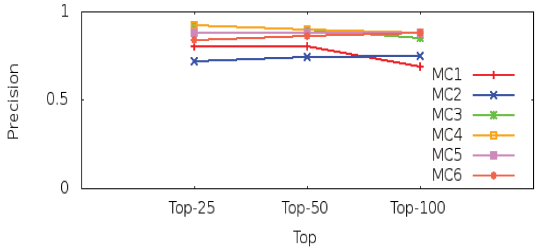
(b) $Q = \{mu, be\}$



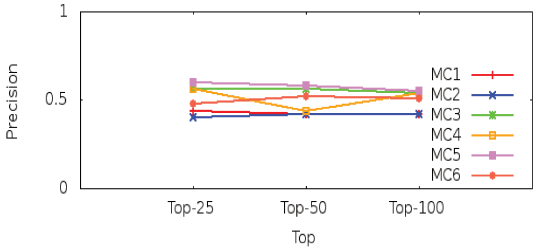
(c) $Q = \{ma, se, ba\}$



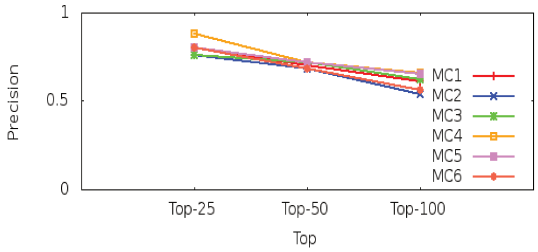
(d) $Q = \{ma, se, mu\}$



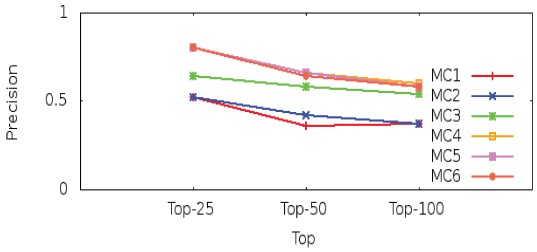
(e) $Q = \{ma, \overline{mu}, \overline{be}, \overline{ba}\}$



(f) $Q = \{ma, wh, se, be\}$



(g) $Q = \{ma, yo, \overline{gl}, mu, \overline{ba}\}$



(h) $Q = \{ma, se, wh, ba, gl\}$

Figure 5.8: MethodC1 vs. MethodC2 vs. MethodC3 vs. MethodC4 vs. MethodC5 vs. MethodC6. Precisions for some selected queries in PubFig dataset.

Chapter 6

Conclusion

In this work, we have discussed how to automatically train visual feature classifiers and associate these features to text describable attributes allowing one to perform high-level queries to a database of images without using text annotations. The train stage is performed by using the LFW database, afterwards, the test stage evaluates the classifiers with both LFW and PubFig datasets. We showed performance in line with recent attribute classifiers from the literature [30] and important texture descriptors such as SASI [5].

We showed that the use of visual dictionaries is worthwhile to learn and represent features in a common and standard form. We have shown that for many visual attributes it is more interesting to characterize the images using a dense-sampling approach to creating visual dictionaries. However, there are some attributes in which histogram of gradients or HoG-based features are more appropriated although there is no final rule for deciding about such cases.

We performed a multilevel approach to characterizing the images in different scales. Then, we evaluated some methods for combining the scores obtained in the different levels. In simple queries, we showed that the use of our multilevel approach improves the accuracies in most of the attribute classifiers.

For dealing with complex queries (more than one attribute), we evaluated in the LFW dataset three approaches from the state of the art for rank fusion (product of probabilities, rank aggregation and rank position) using the attribute classifiers' outputs. We have built 15 attribute classifiers, but the incorporation of classifiers for new attributes is straightforward. Then, we assessed our approaches in the PubFig dataset, where we evaluated two methods for single level and four methods for multilevel approaches. For complex queries, we showed statistically that the multilevel approach also improved the accuracies obtained in a single level.

In addition, as each attribute classifier is independent, we can use the most appropriate characterization method for each attribute. This opens a whole new branch of research

as many attributes are best represented using non-gradient based descriptors such as the ones involving color features or even shape-based features. We aim at investigating these new description forms as we add more attributes to the framework. A good start point is the complete study performed by [41] in which the authors discuss the pros and cons of several image descriptors in the literature and present a comparative study of global color and texture descriptors for web image retrieval.

Finally, we now aim at investigating other classifier fusion techniques (e.g., [38, 39]) to improve the results for even more complex queries. Furthermore, other normalization techniques may be used to reduce the effects of noise, improving the performance achieved by the visual dictionaries (e.g., [49]). To date, we evaluated the multilevel approach using a sparse-sampling, we now aim at performing experiments using the dense-sampling approach. Another future direction is to investigate techniques to measure the level of presence or absence of an attribute and be able to perform queries such as "*white male* in the **sixties**, **partially** *bald* with a **dense** *mustache*, wearing *glasses*", in which we make use of describable attributes (emphasized in italics) and modifiers (emphasized in bold).

Bibliography

[1] W. Yip Andrew and Pawan Sinhao. Contribution of color to face recognition. *Perception*, 31:995–1003, 2002.

[2] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, pages 1–14, 2006.

[3] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T.PAMI)*, 19(7):711–720, jul 1997.

[4] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, 2010.

[5] A. Çarkacıoğlu and F. T. Yarman-Vural. SASI: A Generic Texture Descriptor for Image Retrieval. *Pattern Recognition*, 36(11):2615–2633, 2003.

[6] Giovani Chiachia. *Learning person-specific face representations*. PhD thesis, University of Campinas, Institute of Computing, Campinas, Brazil, 2013.

[7] G. W. Cottrell and J. Metcalfe. Empath: face, emotion, and gender recognition using holons. In *Neural Information Processing Systems (NIPS)*, pages 564–571, 1990.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *European Conference on Computer Vision (ECCV)*, pages 1–14, 2004.

[9] A. Datta, R. Feris, and D. Vaquero. Hierarchical ranking of facial attributes. In *IEEE Intl. Conference on Face and Gesture (F&G)*, pages 36–42, 2011.

[10] Eduardo Alves do Valle Jr. *Local-Descriptor Matching for Image Identification Systems*. PhD thesis, Université de Cergy-Pontoise École Doctorale Sciences et Ingénierie, Cergy-Pontoise, France, June 2008.

[11] J. Fabian, R. Pires, and A. Rocha. Visual Words Dictionaries and Fusion Techniques for Searching People through Textual and Visual Attributes. Pattern Recognition Letters, 2013.

[12] J. Fabian, R. Pires, and A. Rocha. Searching for people through textual and visual attributes. In *Conference on Graphics, Patterns and Images (SIBGRAPI), 2012*, pages 276 –282, aug. 2012.

[13] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. In *ACM-SIAM Symposium on Discrete algorithms (SODA'03)*, pages 28–36, 2003.

[14] Julian J. Faraway. *Practical Regresion and ANOVA using R.* 2002.

[15] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 524–531, 2005.

[16] V. Ferrari and A. Zisserman. Learning visual attributes. In *Neural Information Processing Systems (NIPS)*, pages 1–8, December 2007.

[17] Miller R. G. *Simultaneous Statistical Inference.* Springer, 1981.

[18] B. Golomb, D. Lawrence, and T. Sejnowski. Sexnet: a neural network identifies sex from human faces. In *Neural Information Processing Systems (NIPS)*, pages 572–577, 1990.

[19] Rafael Gonzalez and Richard Woods. *Digital Image Processing.* Prentice-Hall, 3 edition, 2007.

[20] Robert M Haralick and K Shanmugam. Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics (SMC-3)*, 6(1):610–621, 1973.

[21] Brian Heflin, Walter Scheirer, Anderson Rocha, and Terrance E. Boult. *Pattern Recognition, Machine Intelligence and Biometrics: Expanding Frontiers*, chapter A Look at Eye Detection for Unconstrained Environments, pages 361–387. Number ISBN 978-3-642-22406-5 in 1. Springer, 2011.

[22] Byung-Woo Hong, S. Soatto, Kangyu Ni, and T. Chan. The scale of a texture and its application to segmentation. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[23] G. Huang, M. Ramesh, T. Berg, and E. Learned-miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments, 2007.

[24] Herbert F. Jelinek, Ramon Pires, Rafael Padilha, Siome Goldenstein, Jacques Wainer, Terry Bossomaier, and Anderson Rocha. Data fusion for multi-lesion diabetic retinopathy detection. In *IEEE Intl. Symposium on Computer-based Medical System (CBMS)*, Rome, Italy, 2012.

[25] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *Intl. Conference on Computer Vision (ICCV)*, volume 1, pages 604–610, 2005.

[26] J. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[27] R. Kemp, G. Pike, P. White, and A. Musselman. Perception and recognition of normal and negative faces: the role of shape from shading and pigmentation cues. *Perception*, 25:37–52, 1996.

[28] N. Kumar, P. Belhumeur, and S. Nayar. Facetracer: A search engine for collections of images with faces. In *European Conference on Computer Vision (ECCV)*, pages 340–353, 2008.

[29] N. Kumar, A. C. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *IEEE Intl. Conference on Computer Vision (ICCV)*, pages 365–372, 2009.

[30] Neeraj Kumar, Alexander C. Berg, Peter N. Belhumeur, and Shree K. Nayar. Describable visual attributes for face verification and image search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T.PAMI)*, 33(10):1962–1977, October 2011.

[31] L. Lam and C. Y. Suen. Optimal combinations of pattern classifiers. *PRL*, 16(9):945–954, 1995.

[32] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958, 2009.

[33] Yu-Ting Liu, Tie-Yan Liu, Tao Qin, Zhi-Ming Ma, and Hang Li. Supervised rank aggregation. In *International Conference on World Wide Web (WWW'2007)*, pages 481–490, 2007.

[34] David Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, February 2004.

[35] E. Nowak, F. Jurie, and B. Triggs. Sampling strategies for bag-of-features image classification. In *European Conference on Computer Vision (ECCV)*, pages 490–503, 2006.

[36] R. Nuray and F. Can. Automatic ranking of information retrieval systems using data fusion. *Information Processing & Management*, 42(3):595–614, 2006.

[37] U. Park, S. Liao, B. Klare, J. Voss, and A. K. Jain. Face finder: Filtering a large face database using scars, marks and tattoos. Technical Report TR11, Michigan State Univ., 2011.

[38] D. Pedronette and R. da S. Torres. Exploiting contextual information for image re-ranking and rank aggregation. *Intl. Journal of Multimedia Information Retrieval (JMIR)*, 1(1):115–128, 2012.

[39] D. Pedronette and R. da S. Torres. Exploiting pairwise recommendation and clustering strategies for image re-ranking. *Information Sciences (IS)*, 207(1):19–34, 2012.

[40] Daniel Carlos Guimarães Pedronette and Ricardo da S. Torres. Image re-ranking and rank aggregation based on similarity of ranked lists. *Pattern Recognition*, 46(8):2350 – 2360, 2013.

[41] O. A. B. Penatti, E. Valle, and R. S. Torres. Comparative Study of Global Color and Texture Descriptors for Web Image Retrieval. *Journal of Visual Communication and Image Representation*, 23(2):359–380, 2012.

[42] Ramon Pires, Jacques Wainer, Herbert F. Jelinek, and Anderson Rocha. Retinal image quality analysis for automatic diabetic retinopathy detection. In *25th Conference on Graphics, Patterns and Images (SIBGRAPI)*, page To appear., Ouro Preto, Brazil, August 2012.

[43] Liliana Lo Presti and Marco La Cascia. Entropy-based localization of textured regions. In *Intl. Conference on Image Analysis and Processing (ICIAP)*, pages 616–625, 2011.

[44] F. Roberts. *Discrete Mathematical Models with Applications to Social, Biological, and Environmental Problems.* Prentice Hall, 1976.

[45] Anderson Rocha, Tiago Carvalho, Herbert F. Jelinek, Siome Goldenstein, and Jacques Wainer. Points of interest and visual dictionaries for automatic retinal lesion detection. *IEEE Transactions on Biomedical Engineering (T.BME)*, 59(8):2244–2253, 2012.

[46] Frans Schalekamp and Anke Zuylen. Rank aggregation: Together were strong. In *11th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 38–51, 1998.

[47] W. Scheirer, N. Kumar, K. Ricanek, T. Boult, and P. Belhumeur. Fusing with context: a bayesian approach to combining descriptive attributes. In *IEEE Intl. Joint Conference on Biometrics (IJCB)*, pages 1–8, 2011.

[48] Walter Scheirer, Neeraj Kumar, Peter N. Belhumeur, and Terrance E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2933–2940, June 2012.

[49] Walter Scheirer, Anderson Rocha, Ross Michaels, and Terrance E. Boult. Extreme value theory for recognition score normalization. In *European Conference on Computer Vision (ECCV)*, pages 481–495, 2010.

[50] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.

[51] P. Viola and M. Jones. Robust real-time face detection. *Intl. Journal of Computer Vision (IJCV)*, 57:137–154, 2004.