# Exploiting Words and Pictures

by

Tamara Lee Berg

B.S. (University of Wisconsin, Madison) 2001

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor David A. Forsyth, Co-Chair
Professor Jitendra Malik, Co-Chair
Professor Dan Klein
Professor Stephen E. Palmer

Spring 2007

The dissertation of Tamara Lee Berg is approved:

Professor David A. Forsyth, Co-Chair Date

Professor Jitendra Malik, Co-Chair Date

Professor Dan Klein Date

Professor Stephen E. Palmer Date

University of California, Berkeley

Spring 2007

Exploiting Words and Pictures

by

Tamara Lee Berg

## Abstract

Exploiting Words and Pictures

by

Tamara Lee Berg

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor David A. Forsyth, Co-Chair

Professor Jitendra Malik, Co-Chair

There are billions of images with associated text available on the web. Some common areas where pictures and words are naturally linked include: web pages, captioned photographs, and video with speech or closed captioning. The central question that needs to be solved in order to organize these collections effectively is how to extract images in which specified objects are depicted from large pools of pictures with noisy text. This problem is challenging, because the relationship between words associated with an image and objects depicted within the image is often complex.

This thesis demonstrates that, for many situations, collections of illustrated material can be exploited by using information from both the images themselves and from the associated text. The first project demonstrates that one can build a large collection of labeled face images by: identifying faces in images; identifying names in captions; then linking the faces and the names. The process of linking uses the fact that images of the same person tend to look more similar — in appropriate features — than images of different people. Furthermore, the structure of the language in a caption often supplies important cues as to which of the named people actually appear in the image. The second project shows that relations between words and images are strong, even when the text has a less formal structure than captions do. Images retrieved from the internet are classified as containing one of a set of animals or not, using both text that appears near the image and a set of simple image appearance descriptors. Animals are notoriously difficult to identify, because their appearance changes quite dramatically; however, this combination of words and weak appearance descriptors gives us a rather accurate classifier. The third project deals with the tendency of users to attach labels to images that do not belong there, typically because labels are attached to a whole set of images rather than to each image individually. This means that, for example, many images labeled with "Chrysler building" do not in fact depict that building. However, the ones that do tend to look similar in an appropriate sense, and it is possible to find images that are iconic representations of such a category using this cue.

Professor David A. Forsyth, Co-Chair Date

Professor Jitendra Malik, Co-Chair Date

# Acknowledgements

There have been many people without whom this thesis would not have been possible.

Thanks to Alex Berg for everything always and a day.

Thank you to my advisor, David Forsyth, for helping and guiding me throughout my graduate career even after moving half-way across the country. David can always be counted on to provide a source of inspiration, interesting problems, and of course the proper definition for the term “monkey”. Thanks to Jitendra Malik for being a great teacher and for bringing me along to Yahoo Research. I would also like to thank the other members of my thesis committee, Dan Klein for providing insight into NLP and for being a great teaching mentor, and Steve Palmer for showing me the human side of vision.

Thank you to my many wonderful co-authors: Alex Berg, Jaety Edwards, Ryan White, Michael Maire, Ye-Whye Teh and Erik Learned-Miller who helped make the work in this thesis possible. Thanks to Ashley Eden for a great friendship and for providing just enough distractions to keep me sane. And, thank you to Katie Weber, Tyler Wellensiek, Amy Wolf and Adria Smith for being true friends for life since (at least) the first day of 6th grade. Also thanks to many of the other students that made Berkeley a great place to be including but not limited to Andrea Frome, Leslie Ikemoto, Charless Fowlkes, Hao Zhang, Okan Arikan, Deva Ramanan, Brian Milch, Greg Mori, Alyosha Efros, Xiaofeng Ren, Jana Kosecka, Hayley Iben, and many many others...

Finally thank you to my brother, Max, for being a great friend and chef. And thank you to my parents, Arnold and Man-Li Miller, for always believing in me and telling me I could do whatever I set my mind to.

*Dedicated to Man-Li and Arnold Miller*

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

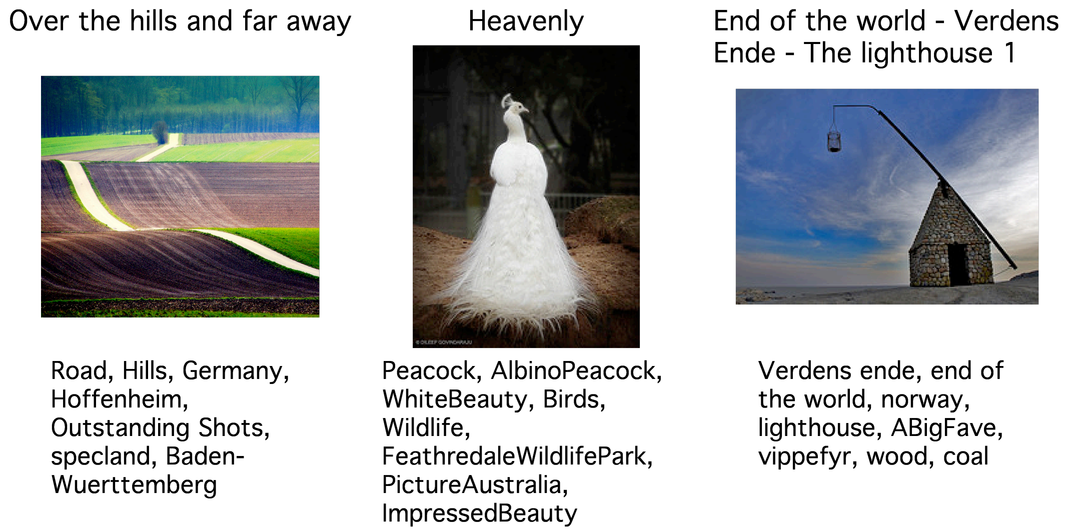

Over the hills and far away
Heavenly
End of the world - Verdens Ende - The lighthouse 1
Road, Hills, Germany, Hoffenheim, Outstanding Shots, specland, Baden-Wuerttemberg
Peacock, AlbinoPeacock, WhiteBeauty, Birds, Wildlife, FeathredaleWildlifePark, PictureAustralia, ImpressedBeauty
Verdens ende, end of the world, norway, lighthouse, ABigFave, vippefyr, wood, coal

Figure 1.1: *Images from Flickr.com including text tags associated with each image.*

Words and pictures are often naturally linked. This can be seen in the availability of many vast collections containing images with associated text. Sources for these include: collections of museum material; digital library collections; any video with sound or closed captioning; images collected from the web with their enclosing web pages; or captioned news images. The amount of multi-modal data accessible on the web is enormous and literally growing exponentially. One source for consumer photographs is Flickr (figure 1.1), an on-line photo sharing site where users can upload photos, and share them with their friends, family, or with the world. Flickr currently has over 500 million photographs and is expanding by a million new uploads per day. Many of these photos have been associated by the user with text strings (tags). With the growing popularity of sites like Flickr, Google Video, and YouTube, the amount of visual data, and also of visual data associated with some sort of text will only increase in coming years.

In order to provide users with effective access to these collections there are several tasks that we would like to be able to perform. We would like to organize the collections automatically or with a small amount of user input. We would like to be able to search the collections with high accuracy. And, we would like to be able to browse the collections in natural ways.

Current access to these collections through indexing sites like Google image search or Flickr search use purely text based methods to guide search. As can be seen in figure 1.2, this leads to partial success (*e.g.* top row first three images), but also to many falsely labeled images (*e.g.* top row, right-most image, bottom row right-most image). This happens because the text associated with an image often does not describe the content of the image. For example, although the right-most image in the bottom row does not depict a monkey, it is highly ranked for a query on the term monkey because it is from the website monkey-business.com. In order to perform well at any of the three key tasks described, we will have to utilize information extracted from both the images and their associated text.

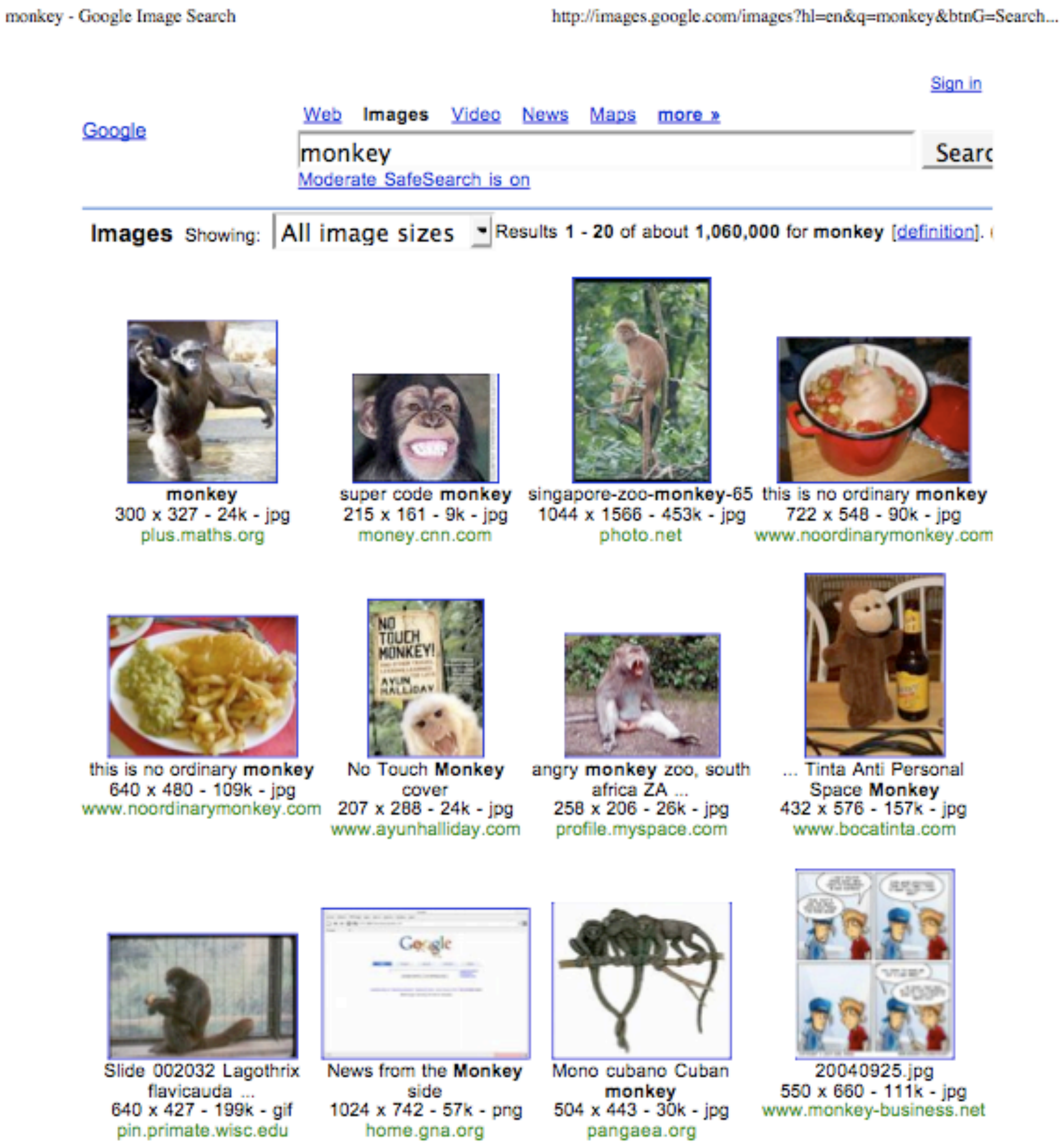
We want to exploit the fact that in these collections, pictures and their associated annotations provide complementary information. In figure 1.1, we can see this illustrated in the middle photograph depicting a purse. Notice that nowhere in the caption is it mentioned that the purse is black. However, this would be an extremely simple feature for a computer vision algorithm to extract from the image. Also, just by visually examining the dress on the right it would be extremely difficult for a computer (or person) to determine that the dress is made of linen, though this could be determined by analyzing the associated caption. This complementary nature suggests that combining information from images with information from any associated text could be quite beneficial. In this thesis I will describe several projects where we have shown this to be true.

## 1.2 Outline

The rest of this thesis proceeds as follows:

Chapter 2 will present a broad literature survey of related research, giving an overview of work on organizing large collections of images using: textual information, image information, and combined text and image information.

In Chapter 3, I will describe a system to automatically label faces in news pho-


monkey - Google Image Search
http://images.google.com/images?hl=en&q=monkey&btnG=Search...
Sign in
Google
Web
Images
Video
News
Maps
more »
Searc
Moderate SafeSearch is on
Images
Showing:
All image sizes

monkey

Results 1 - 20 of about 1,060,000 for monkey [definition].

monkey
300 x 327 - 24k - jpg
plus.maths.org

super code monkey
215 x 161 - 9k - jpg
money.cnn.com

singapore-zoo-monkey-65
1044 x 1566 - 453k - jpg
photo.net

this is no ordinary monkey
722 x 548 - 90k - jpg
www.noordinarymonkey.com

this is no ordinary monkey
640 x 480 - 109k - jpg
www.noordinarymonkey.com

No Touch Monkey cover
207 x 288 - 24k - jpg
www.ayunhalliday.com

angry monkey zoo, south africa ZA ...
258 x 206 - 26k - jpg
profile.myspace.com

... Tinta Anti Personal Space Monkey
432 x 576 - 157k - jpg
www.bocatinta.com

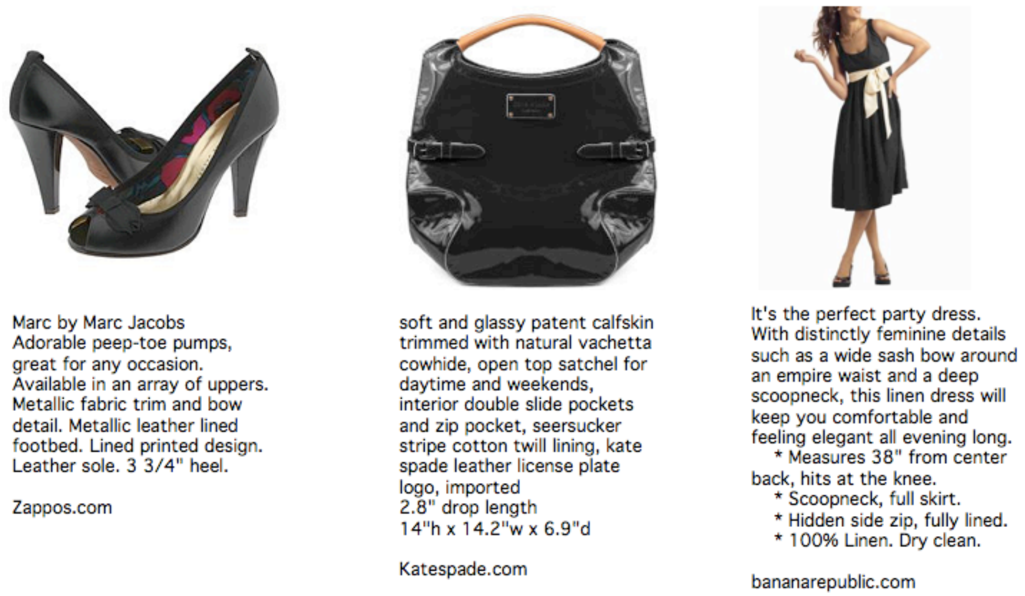
Slide 002032 Lagothrix flavicauda ...
640 x 427 - 199k - gif
pin.primate.wisc.edu

News from the Monkey side
1024 x 742 - 57k - png
home.gna.org

Mono cubano Cuban monkey
504 x 443 - 30k - jpg
pangaea.org

20040925.jpg
550 x 660 - 111k - jpg
www.monkey-business.net

Figure 1.2: *Top ranked results for a Google image search on the query string "monkey" (from March 27, 2007). Notice that some of the images portray monkeys (*e.g. *top row first three images) while others do not (*e.g. *top row right-most image bottom row right-most image). These images are highly ranked by Google image search because current image search engines consider only text information for ranking images and these images are associated with the word monkey (*e.g. *bottom row right most image is from the website: www.monkey-business.net).*



Marc by Marc Jacobs
Adorable peep-toe pumps, great for any occasion. Available in an array of uppers. Metallic fabric trim and bow detail. Metallic leather lined footbed. Lined printed design. Leather sole. 3 3/4" heel.

Zappos.com

soft and glassy patent calfskin trimmed with natural vachetta cowhide, open top satchel for daytime and weekends, interior double slide pockets and zip pocket, seersucker stripe cotton twill lining, kate spade leather license plate logo, imported
2.8" drop length
14"h x 14.2"w x 6.9"d

Katespade.com

It's the perfect party dress. With distinctly feminine details such as a wide sash bow around an empire waist and a deep scoopneck, this linen dress will keep you comfortable and feeling elegant all evening long.
* Measures 38" from center back, hits at the knee.
* Scoopneck, full skirt.
* Hidden side zip, fully lined.
* 100% Linen. Dry clean.

bananarepublic.com

Figure 1.3: *These photographs and associated captions demonstrate the complementary nature of images and text. In the middle photograph of a purse notice that nowhere in the caption is it mentioned that the bag is black. However, this would be an extremely simple feature for a computer vision algorithm to extract. Also, looking at the dress on the right, it would be extremely difficult for a computer (or even a person) to determine that the dress is made of linen, though this could be determined by looking at the associated caption.*

tographs [Berg *et al.*, 2004a; Berg *et al.*, 2004b; Berg *et al.*, In Submission]. These photographs are distributed by the Associated Press along with captions. The task is to extract faces from the photographs, names from the captions and then automatically associate the correct name with each face. This is a reduced version of the standard face recognition problem where the question is, “Given a face, whose face is this?” The question we pose is, “Given a face and a small set of possible names, which (if any) is the correct identity?”

The basic intuition is that very often pictures that share a name in their associated captions are also going to share a face. If, for example, we have a picture with the names George Bush and Colin Powell detected in the associated caption and two faces detected in the image, and another picture where we detect the names George Bush and Tony Blair along with two faces, then we can guess that the face in common between these two pictures must be that of George Bush.

To find the correct assignments, we will make use of two advances: proper name detection in natural language processing, and face detection in computer vision. These allow us to concentrate on the correspondence problem between the relevant entities, faces in the image and names in the text. In addition, the captions for news photographs are written by professional writers to convey what is portrayed in the photograph and are fairly stylized. We take advantage of this stylization to build models of language context (Section 3.5).

In Chapter 4, I will describe a project to classify animal images from the web [Berg and Forsyth, 2006]. There are many more relevant images available online than are returned by an image search query on sites such as Google and Yahoo. One significant cause is the lack of image information in ranking schemes for search results. We attempt to address this by using Google *Text* search to identify pages with content related to a query (*e.g.* monkeys) and then re-rank all images on those pages using both image features and text. This task is easier than object recognition – instead of looking through all images to find monkeys we only look at images on pages likely to be about monkeys. This and the combination of image features with text features produces surprisingly good search results for the animal query terms used in evaluation.

There are several way sin which this project proves to be a more challenging problem than that of Chapter 3. First it moves beyond the faces of Chapter 3 to even more challenging visual categories, animals. Animals in particular are quite difficult for computer vision systems to recognize because they can take on a wide range of aspects, configurations and appearances. Animal categories also typically contain multiple species with varied appearances that can make recognition problematic. On the text side of the problem, we also move from the relatively stylized nature of captions to free text on web pages. To deal with these challenges we will take advantage of some recent language modeling advances and utilize the power of combining multiple relatively simple image and text based cues.

Finally, in Chapter 5, I will present a project on finding iconic images for a set of monument categories [Berg and Forsyth, 2007]. Here we define iconic as a large, cleanly depicted example of a given category from a characteristic viewpoint. Some object categories are more commonly photographed than others. One such type of category is monuments such as the Golden Gate Bridge or the Chrysler Building that are visited by thousands of people each day. For example, on Flickr there are 83,270 photographs tagged with Eiffel Tower. Among these photos there are some that are very characteristic representations of the category and many photos that are poor representations. Being able to determine which photos are the most iconic has useful implications for both browsing and search.

The word based component is smaller for this project than for the previous two systems. We use a text based search to guide the initial data collection, but rely on purely image based methods for the bulk of the method. One could imagine adding more input from the text information by analyzing any other tags associated with the image to determine whether they suggest that the photo belongs to the specified category or not.

I conclude in Chapter 6 with a summary of the datasets we have produced, some baseline recognition experiments performed on our face dataset, and with some description of how the products of this thesis have been used in other research.

# Chapter 2

# Previous Work

There are three main areas of related work: using textual information to classify images, using image content to classify images, and using a combination of text and image information for classification. Text based systems provide natural query interaction, but since they don't use any image information may provide noisy results. Image based systems most often provide similarity based results where a sample image is used to retrieve similar looking images. Chapter 5 will present some results related to this area of research for the image based classification of iconic images.

Being required to provide a sample image to retrieve similar looking images may be a less useful or familiar form of interaction for the user than natural language based interactions. Systems that use a combination of word and text based classification can provide both natural interaction and good image similarity. The systems I will describe in Chapter 3 and Chapter 4 build on this line of work, using combinations of word and text information for various tasks.

## 2.1 Words

Many of the first image retrieval systems, including the commercially successful Yahoo image search and Google image search use text analysis for image retrieval. These systems apply effective text based methods to the image retrieval task without ever looking at a single pixel.

**Automatic text based image retrieval systems** work by annotating images with words extracted from the web pages containing the images and then applying text based retrieval methods to search these annotated collections of images. Since not all web pages containing images also contain text some systems utilize hypertext information in addition to the surrounding web page text. In Harmandas *et al.* [Harmandas *et al.*, 1997], connectivity information is used to induce textual annotations of images. Images are represented using the text found on pages linked to them in one or two step links. They then use standard information retrieval techniques to index these new image representations. PicASHOW [Lempel and Soffer, 2001], applies co-citation based approaches and PageRank influenced methods to the application of image retrieval.

**Text based indexing** has been used for the organization and keyword retrieval of images since long before the advent of the web, with common collections including the Getty collection of images and Corbis. This area of work concentrates on developing text based ontologies and classification schemes for image description. Rasmussen [Rasmussen, 1997] provides a review of the efforts to manually or automatically index images for later lookup and use.

**Our work** will build on and utilize various related methods from the natural language community, including advances in language topic modeling (Section 4.3.1), and in maximum entropy models which we use to learn an effective model for language context (Section 3.5.2). We will show results on using these natural language models for labeling faces in news photographs (Section 3.6), and re-ranking images for search (Section 4.4).

## 2.2 Pictures

### 2.2.1 CBIR

There has been much work on classifying images according to their content for content based image retrieval (CBIR). These systems only use image information for classification and typically retrieve images by measuring their similarity to a given query image. Some of the major projects are described here, and more in depth reviews are provided in [Rui *et al.*, 1997; Smeulders *et al.*, 2000; Datta *et al.*, 2005].

**Color:** QBIC, from IBM [Flickner *et al.*, 1995] was one of the earliest CBIR systems. QBIC allowed the user to paint a query by selecting from a color wheel and then retrieve images with similar colors and color layouts. PicToSeek [Gevers and Smeulders, 2000] classified images into a set of descriptive classes (portraits, indoor/outdoor scenes and synthetic) and then used color features to index the images in each of these classes.

**Other Features:** Other features like shape and texture have also been used in addition to color for image classification and retrieval. The Photobook system [Pentland *et al.*, 1996] used eigenimages to describe appearance and textural features for comparing image similarities. The Virage search engine [Bach *et al.*, 1996] provided an open framework for developers to plug in image features to solve specific image management problems. CIRES [Iqbal and Aggarwal, 2002], uses higher-level image structures such as line segments, "U" junctions and parallel lines to aid image retrieval. Several groups have used various image features to automatically re-rank search results [Fergus *et al.*, 2005; Fergus *et al.*, 2004; Tong and Chang, 2001]. We perform a similar image re-ranking task in Chapter 4 although unlike these methods, we incorporate both text and image information in the re-ranking process.

**Region Based:** Some image retrieval systems compute image similarity based on properties of individual image regions. In the NeTra system [Ma and Manjunath, 1999] a user is presented with the segmented regions of the image and selects the regions that they would like to match along with the attribute (such as color) to be used for evaluating similarity. The Simplicity system [Wang *et al.*, 2001] divides images into types, graphs vs photograph and textured vs non-textured. These extracted types allow them to use some semantically-adaptive search methods. These methods first apply a segmentation model to divide the image into regions that ideally correspond to different objects and then use those regions for retrieval.

These region based methods are related to the system for ranking iconic images described in Chapter 5. For this project, we use a binary segmentation method to divide the image into regions roughly corresponding to the subject and background of the image. We then rank images according to how similar they are to a set of example images. The specific segmentation method we use is a Markov Random field models dating back to Geman and Geman [Geman and Geman, 1984] and studied by many others since. We use a Markov Random field segmentation described by Boykov and Kolmogorov [Boykov and Kolmogorov, 2004]. This is an implementation of a min-cut/max-flow algorithm to compute a two label segmentation efficiently.

Many people believe that segmentation and recognition are linked in some natural way. There have been some papers showing that segmentation can help improve recognition results. Barnard *et al.* [Barnard *et al.*, 2003b] show that different possible segmentations can be judged according to how well they predict words for regions and that word prediction can be improved by using these segmentations. Liebe and Schiele [Leibe *et al.*, 2006] use segmentation as a way of integrating individual image cues and show that this multi-cue combination scheme increases detection performance compared to any cue in isolation. We show in a simple user study (section 5.4.2) that segmentation can be helpful for selecting iconic images from a set of monument categories.

### 2.2.2 Object Recognition

All three projects described in this thesis have a link to general object recognition. The task of object recognition is somewhat different from that of content based image retrieval. Instead of trying to retrieve images that are similar to a query, the goal is to recognize what object or objects are present in visual data. The object categories that we focus on are: faces in Chapter 3, animals in Chapter 4, and monuments in Chapter 5.

**Object Recognition** has been thoroughly researched, but is by no means a solved problem. There has been a recent explosion of work in appearance based object recognition using local features, in particular on the Caltech-101 Object Categories Dataset introduced in [Fei-Fei *et al.*, 2004]. Some methods use constellation of parts based models trained using EM [Fergus *et al.*, 2003]. Others employ probabilistic models like pLSA or LDA [Sudderth *et al.*, 2005; Sivic *et al.*, 2005], or spatial pyramid matches [Lazebnik *et al.*, 2006; Grauman and Darrell, 2005]. Several of the most effective current recognition systems [Berg *et al.*, 2005; Zhang *et al.*, 2006; Frome *et al.*, 2006] use as their base feature, Geometric Blur, a shape descriptor which we employ in both for our project to classify animal images (Chapter 4), and in our face labeling project (Chapter 3). Object recognition is unsolved, but we show in Chapter 4 that whole image classification can be successful using fairly simple methods.

There has been some preliminary work on voting based methods for image classification in the Caltech-101 Dataset using geometric blur features [Berg, 2005]. In an alternative forced choice recognition task this method produces quite reasonable results (recognition rate of 51%) as compared with the best previously reported result using deformable shape matching (45%) [Berg *et al.*, 2005] [1].Chapter 4 uses a modified voting method for image retrieval that incorporates multiple sources of image and text based information.

**Animals** are demonstrably among the most difficult classes to recognize [Berg *et*

[1] At the time of publishing two new methods based on spatial pyramid matching [Lazebnik *et al.*, 2006] and k-NN SVMs [Zhang *et al.*, 2006] have since beat this performance with respectively 56% and 59% recognition rates for 15 training examples per class.

*al.*, 2005; Fei-Fei *et al.*, In Press]. This is because animals often take on a wide variety of appearances, depictions and aspects. Animals also come with the added challenges of articulated limbs and the fact that multiple species while looking quite different in appearance have the same semantic category label, *e.g.* "African leopards", "black leopards" and "clouded leopards".

There has been some work on recognizing animal categories using deformable models of shape [Ramanan *et al.*, 2005; Schmid, 2001]. However, they concentrate on building a single model for appearance and would not be able to handle the large changes in aspect or multiple species that we find in our data ( 4).

**Automatic Subject Determination:** There has been some previous work on automatically determining the subject of photographs related to our work in Chapter 5. Li *et al.* [Li *et al.*, 1999] automatically determine the object of interest in photographs. However, their focus is on images with low depth of field. Banerjee and Evans [Banerjee and Evans, 2004] propose an in-camera main subject segmentation algorithm that uses camera controls to automatically determine the subject. Since we collect our images from the web we cannot use this method. The work most related to ours in this area is Luo *et al.* [Luo *et al.*, 2001] who use region segmentation and probabilistic reasoning to automatically determine subjects in unconstrained images, although they do this in a very different manner than our method.

### 2.2.3 Faces

There has been much work related specifically to recognizing faces, the subject of Chapter 3. The previous work on recognizing faces in photographs without the benefit of textual information is extensive and we review some work in this area. We also review some work on linking faces with other types of data.

#### 2.2.3.1 Face Recognition

We review only important points, referring readers to Zhao *et al.* for a comprehensive general survey of the area [Zhao *et al.*, 2003]. Further reviews appear in [Gross *et al.*, 2001; Yang *et al.*, 2002; Phillips *et al.*, 2002]. Early work uses nearest neighbor classifiers based on pixel values, typically dimensionality reduced using principal component analysis (PCA) [Sirovich and Kirby, 1987; Turk and Pentland, 1991]. Linear discriminant methods offer an improvement in performance [Belhumeur *et al.*, 1997]. More recently, it has been shown that models based on 3D structure, lighting, and surface appearance [Blanz and Vetter, 2003; Phillips *et al.*, 2002] or appearance based methods that explicitly model pose [Gross *et al.*, 2004] give better recognition accuracy, but can be somewhat hard to fit for arbitrary faces.

Face recognition is known to be difficult, and applications have failed publicly [Scheeres, 2002]. Philips and Newton show that the performance of a face recognition system on a data set can largely be predicted by the performance of a baseline algorithm, such as principal component analysis, on the same data set [Phillips and Newton, 2002]. Since recognition systems work well on current face data sets, but poorly in practice, this suggests that the data sets currently used are not representative of real world settings. Because current data sets were captured in the lab, they may lack important phenomena that occur in real face images. To solve face recognition, systems will have to deal well with a data set that is more realistic, with wide variations in color, lighting, expression, hairstyle and elapsed time.

#### 2.2.3.2 Linking Faces with Other Data

It appears to be considerably simpler to choose one of a few names to go with a face than it is to identify the face. This means one might be able to link faces with names in real data sets quite successfully. Very good face detectors are now available (important samples of this huge literature include [Poggio and Sung, 1995; Rowley *et al.*, 1996a; Rowley *et al.*, 1996b; Rowley *et al.*, 1998a; Rowley *et al.*, 1998b; Sung and Poggio, 1998; Ioffe and Forsyth, 2001; Viola and Jones, 2004; Yang *et al.*, 2002; Schneiderman and Kanade, 2000; Mikolajczyk, 2002]); we use the detector of [Mikolajczyk, 2002]. Attempts to link names and faces appear quite early in the literature. Govindaraju *et al.* describe a method that finds faces using an edge curve criterion, and then links faces to names in captions by reasoning about explicit relational information in the caption (they give the example of the caption "Cardinal O'Connor (center), George Bush (left) and Michael Dukakis...") [Govindaraju *et al.*, 1989]. There is a description of an expanded version of this system, which uses language semantics even more aggressively (for example, the system possesses the knowledge that a portrait is a face surrounded by a frame, p. 53), in [Srihari, 1995]. Zhang *et al.* show that a text based search for an image of a named individual is significantly improved by testing to see whether returned images contain faces [Zhang *et al.*, 1999]. Naaman *et al.* show that labels used frequently for "nearby" images constrain the labels that can be used for the current face image [Naaman *et al.*, 2005].

Satoh and Kanade work with video and a transcription [Satoh and Kanade, 1997].

They represent faces using principal components, identify named entities in the transcript, and then build a smoothed association between principal component faces and names that appear nearby in the transcript. Similar faces appearing near two instances of a name reinforce the association function; different names appearing near similar faces weaken it. The system operates on some 320 face instances (taken from some 4.5 hours of video) and 251 name instances, and reports names strongly associated with a given face. Satoh *et al.* describe a variant of this system, which is also capable of reading captions overlaid on video frames; these prove to be a strong cue to the identity of a face [Satoh *et al.*, 1999]. The method is comparable with multiple-instance learning methods (above). Yang and Hauptmann describe a system for learning such association functions [Yang and Hauptmann, 2004].

Houghton works with video, transcriptions, automatically interpreted video captions and web pages (from news and other sources), to build a database of named faces [Houghton, 1999]. The question of correspondence is not addressed; the data appears to contain only single face/single name pairs. Houghton's system will produce an N-best list of names for query faces.

An important nuisance in news video are anchor persons, whose faces appear often and are often associated with numerous names. Song *et al.* detect and remove anchor persons and then use a form of multiple-instance learning to build models of two well-known individuals from video data [Song *et al.*, 2004].

Yang *et al.* compare several forms of multiple-instance learning for attaching one of a set of possible labels to each face image [Yang *et al.*, 2005b]. In their problem, each image has a set of possible name labels, and one knows whether the right label appears in that set (there are 234 such images) or not (242). There are approximately 4.7 available labels for each face image. The paper compares four multiple-instance algorithms, each in two variants (one either averages over correspondences between a face and labels, or chooses the best correspondence) and each with two types of training data (only positive bags vs. all bags), and two supervised methods. Multiple-instance methods label between 44% and 60% of test images correctly and supervised methods label between 61% and 63% of test images correctly.

Methods to label faces in consumer images are described in [Zhang *et al.*, 2003; Zhang *et al.*, 2004]. In this problem, the user acts as an oracle — so there is no correspondence component — but the oracle must not be queried too often.

Fitzgibbon and Zisserman automatically discover cast listings in video using affine-invariant clustering methods on detected faces and are robust to changes in lighting, viewpoint and pose [Fitzgibbon and Zisserman, 2002]. More recently, Arandjelovic and Zisserman have extended this work to suppress effects of background surrounding the face, refine registration and allow for partial occlusion and expression change [Arandjelovic and Zisserman, 2005].

**Our efforts** in Chapter 3 differ from the work surveyed above in three important points. First, our typical data item consists of representations of several faces *and* of several names, and we must identify what, if any, correspondences are appropriate. Second, we reason explicitly about correspondence. This allows us to build discriminative models that can identify language cues that are helpful. Third, we operate at a much larger scale (approximately 30,000 face images), which can help to make correspondence reasoning more powerful.

## 2.3 Words and Pictures

There have also been some efforts to combine information from images with text for a variety of tasks including: search, the automatic labeling of images with keywords, image clustering, and labeling regions within images.

**Search:** Carson *et al.* demonstrate examples of joint image-keyword searches [Carson *et al.*, 2002]. The webSeer search engine [Swain *et al.*, 1997] indexes images using associated text plus some image content analysis based on attributes obtained from the image header (like image size, file type etc) and a classification of the images into photographs vs artificial images. WebSeek [Smith and Chang, 1997] uses text and url analysis along with color histograms and user relevance feedback to catalog visual information into a pre-defined taxonomy for browsing and search. Cascia *et al.* [La Cascia *et al.*, 1998] compute a representative vector based on analysis of text (latent semantic indexing) and visual analysis (color and orientation histograms). Quack *et al.* [Quack *et al.*, 2004] compute several low-level MPEG-7 visual features and use these along with keywords to index images. Joshi *et al.* show that one can identify pictures that illustrate a story by searching annotated images for those with relevant keywords, then ranking the pool of images based on similarity of appearance [Joshi *et al.*, 2004].

**Clustering:** Barnard *et al.* cluster Corel images and their keywords jointly to produce a browsable representation [Barnard and Forsyth, 2001]; the clustering method is due to Hofmann and Puzicha [Hofmann and Puzicha, 1998]. Barnard *et al.* show that this form of clustering can produce a useful, browsable representation of a large collection of annotated art in digital form [Barnard *et al.*, 2001]. Gao *et al.* [Gao *et al.*, 2005] cluster web images using low level image features and surrounding text in a bipartite graph co-partitioning framework. They use their system to present the user with web clusters representing different categories within a collection of images from the Photography Museums and Galleries of the Yahoo! Directory.

**Attaching keywords to images:** Clustering methods methods can typically be used to predict keywords from images, and accuracy at keyword prediction is used as one test of such methods (see also [Barnard *et al.*, 2003b]). There are two varieties of the prediction task: predicting words associated with an image (*auto-annotation*) and predicting words associated with particular image structures. Maron and Ratan attach keywords to images using *multiple-instance learning* [Maron and Ratan, 1998]. Multiple-instance learning is a general strategy to build classifiers from "bags" of labeled examples. Typically, one knows only that a bag contains or does not contain a positive example, but not which example is positive. Methods attempt to find small regions in the feature space that appear in all positive bags and no negative bags; one can visualize these methods either as a form of smoothing [Maron and Lozano-Pérez, 1998; Zhang and Goldman, 2001], fitting an SVM [Andrews *et al.*, 2003; Tao *et al.*, 2004], or using geometric reasoning [Dietterich *et al.*, 1997]. Comparisons between methods appear in [Ray and Craven, 2005]. Chen and Wang describe a variant multiple-instance learning method, and use it to predict keywords from regions [Chen and Wang, 2004]. Duygulu *et al.* use explicit correspondence reasoning to associate keywords with image regions [Duygulu *et al.*, 2002], using a statistical translation model from [Brown *et al.*, 1993]. Blei and Jordan use a variant of latent

Dirichlet allocation to predict words corresponding to particular image regions in an auto-annotation task [Blei and Jordan, 2003]. Barnard *et al.* demonstrate and compare a wide variety of methods to predict keywords, including several strategies for reasoning about correspondence directly [Barnard *et al.*, 2003a]. Li and Wang used 2-dimensional multi-resolution hidden markov models on categorized images to train models representing a set of concepts [Li and Wang, 2003]. They then used these concepts for automatic linguistic indexing of pictures. Jeon *et al.* demonstrate annotation and retrieval with a cross-media relevance model [Jeon *et al.*, 2003]. Lavrenko *et al.* used continuous space relevance models to predict the probability of generating a word given image regions for automatic image annotation and retrieval [Lavrenko *et al.*, 2003].

**Other activities:** Relations between text and images appear to be deep and complex. Barnard and Johnson show one can disambiguate the senses of annotating words using image information [Barnard and Johnson, 2005]. Yanai and Barnard use region entropy to identify words that have straightforwardly observed visual properties ("pink" does, "affectionate" does not) [Yanai and Barnard, 2005]. All this work has tended to emphasize general image constructs (such as regions), but one might instead use detectors and link the detector responses with words. Faces are of particular interest.

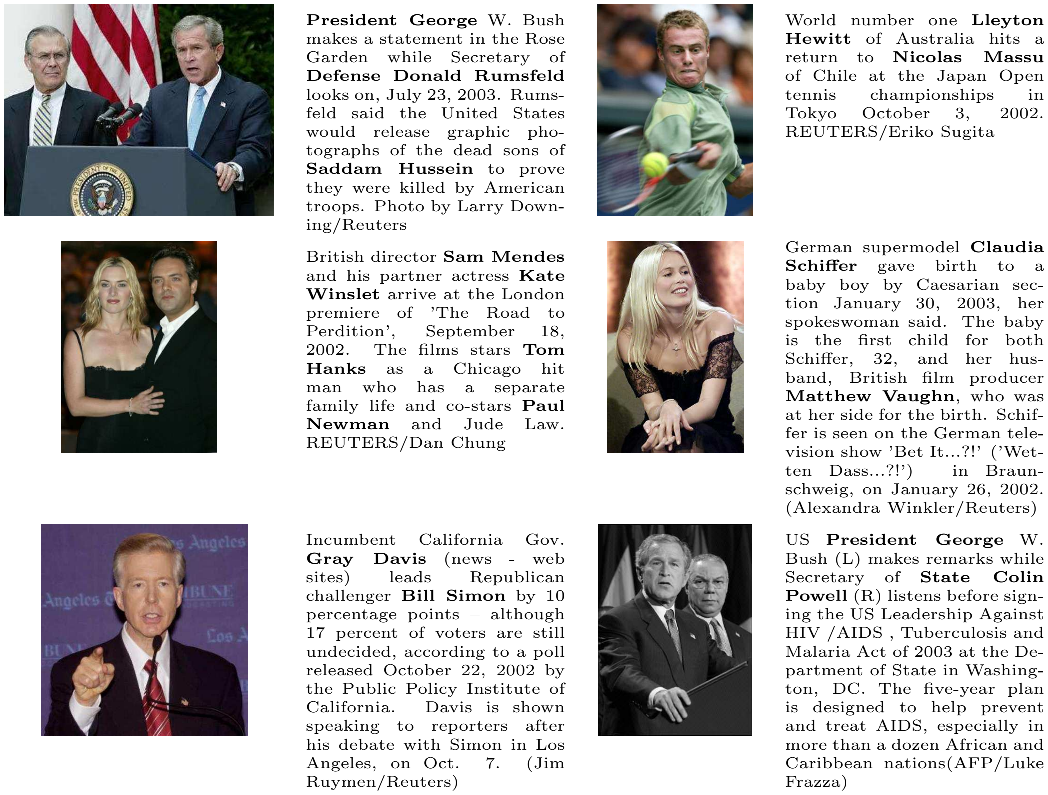
**This thesis** will present considerable evidence that using image information in conjunction with text information allows you to perform significantly better at a variety of tasks. In Chapter 3 we show that for the task of automatically labeling faces in news photographs, utilizing language context models in addition to face recognition techniques gives much higher accuracy than using purely image based information. In Chapter 4 several different kinds of cues, including text and image cues, are used to each rank a set of images independently. We show that the combination of text and image rankings performs favorably over both our purely text based ranking and also over Google's original ranking (Section 4.4).

# Chapter 3

# Names and Faces

In this chapter we show that a large and realistic face data set can be built from news photographs and their associated captions. Our automatically constructed face data set consists of 30,281 face images, obtained by applying a face finder to approximately half a million captioned news images. The faces are labeled using image information from the photographs and word information extracted from the corresponding caption. This data set is more realistic than usual face recognition data sets, because it contains faces captured "in the wild" under a wide range of positions, poses, facial expressions, and illuminations. After faces are extracted from the images, and names with context are extracted from the associated caption, our system uses a clustering procedure to find the correspondence between faces and their associated names in the picture-caption pairs.

The context in which a name appears in a caption provides powerful cues as to whether it is depicted in the associated image. By incorporating simple natural



**President George** W. Bush makes a statement in the Rose Garden while Secretary of **Defense Donald Rumsfeld** looks on, July 23, 2003. Rumsfeld said the United States would release graphic photographs of the dead sons of **Saddam Hussein** to prove they were killed by American troops. Photo by Larry Downing/Reuters

World number one **Lleyton Hewitt** of Australia hits a return to **Nicolas Massu** of Chile at the Japan Open tennis championships in Tokyo October 3, 2002. REUTERS/Eriko Sugita

British director **Sam Mendes** and his partner actress **Kate Winslet** arrive at the London premiere of 'The Road to Perdition', September 18, 2002. The films stars **Tom Hanks** as a Chicago hit man who has a separate family life and co-stars **Paul Newman** and Jude Law. REUTERS/Dan Chung

German supermodel **Claudia Schiffer** gave birth to a baby boy by Caesarian section January 30, 2003, her spokeswoman said. The baby is the first child for both Schiffer, 32, and her husband, British film producer **Matthew Vaughn**, who was at her side for the birth. Schiffer is seen on the German television show 'Bet It...?!' ('Wetten Dass...?!') in Braunschweig, on January 26, 2002. (Alexandra Winkler/Reuters)

Incumbent California Gov. **Gray Davis** (news - web sites) leads Republican challenger **Bill Simon** by 10 percentage points – although 17 percent of voters are still undecided, according to a poll released October 22, 2002 by the Public Policy Institute of California. Davis is shown speaking to reporters after his debate with Simon in Los Angeles, on Oct. 7. (Jim Ruymen/Reuters)

US **President George** W. Bush (L) makes remarks while Secretary of **State Colin Powell** (R) listens before signing the US Leadership Against HIV /AIDS , Tuberculosis and Malaria Act of 2003 at the Department of State in Washington, DC. The five-year plan is designed to help prevent and treat AIDS, especially in more than a dozen African and Caribbean nations(AFP/Luke Frazza)

Figure 3.1: *Some typical news photographs with associated captions from our data set. Notice that multiple faces may appear in a single picture and that multiple names may occur in a particular caption. Our task is to detect faces in these pictures, detect names in the associated captions and then correctly label the faces with names (or "NULL" if the correct name does not appear in the caption). The output of our system on these images appears in Figure 3.5.*

language techniques, we are able to improve our name assignment significantly. We examine two models of word context, a naive Bayes model and a maximum entropy model. Once our procedure is complete, we have an accurately labeled set of faces, an appearance model for each individual depicted, and a natural language model that can produce accurate results on captions in isolation.

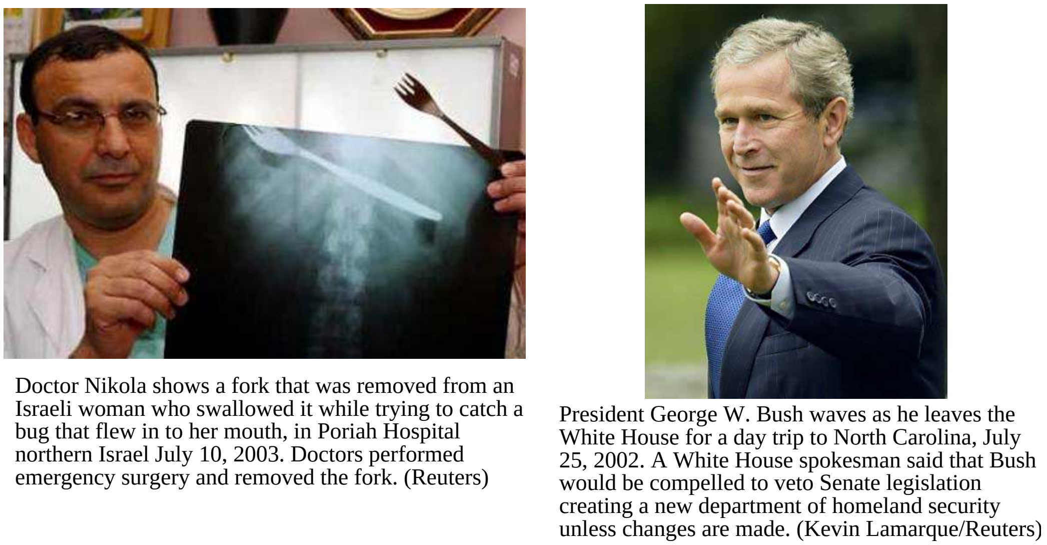
## 3.1 Introduction

This chapter shows how to exploit the success of face detection to build a rich and reasonably accurate collection of labeled faces. The input is a collection of news photographs with captions. Face detection extracts faces from each image while natural language processing finds proper names in the associated caption. For each photo/caption pair, a *data item*, the remaining step, to solve the assignment problem between names and faces, is the central challenge of this system.

We attack the assignment problem in two ways. First we develop an iterative method for determining correspondences for a large number of data items, along a familiar line of reasoning. If we knew an appearance model for the faces associated with each name, then finding a correspondence would be straightforward; similarly if we knew a correspondence then estimating an appearance model for the faces associated with each name would be straightforward. These observations lead to natural iterative algorithms. Second we show that there are contextual language cues that suggest particular names in a caption do not refer to a pictured face. These cues are learned and exploited in the iterative algorithms, improving the resulting correspondences.

### 3.1.1 Overview

We have collected a very large data set of captioned news images (section 3.2). We describe our construction of a face dictionary as a sequence of three steps. First, we detect names in captions using an open source named entity recognizer [Cunningham



Doctor Nikola shows a fork that was removed from an Israeli woman who swallowed it while trying to catch a bug that flew in to her mouth, in Poriah Hospital northern Israel July 10, 2003. Doctors performed emergency surgery and removed the fork. (Reuters)

President George W. Bush waves as he leaves the White House for a day trip to North Carolina, July 25, 2002. A White House spokesman said that Bush would be compelled to veto Senate legislation creating a new department of homeland security unless changes are made. (Kevin Lamarque/Reuters)

Figure 3.2: *In our initial set of photo-caption pairs, some individuals, like President Bush (***right***), appear frequently and so we have many pictures of them. Most people, however, like Dr. Nikola (***left***), appear only a few times or in only one picture. This distribution reflects what we would expect from real applications. For example, in airport security cameras, a few people, (e.g. airline staff) might be seen often, but the majority of people would appear infrequently. Studying how recognition systems perform under these circumstances and providing data sets with these features is necessary for producing reliable face recognition systems.*

*et al.*, 2002]. Next, we detect and represent faces, as described in section 3.3.3. Finally, we associate names with faces, using either a clustering method (section 3.4) or an enhanced method that analyzes text cues (section 3.5).

Our goal is more restricted than general face recognition in that we need only distinguish between a small number of names in the corresponding caption. There appear to be significant benefits in explicit correspondence reasoning, and we report results for name-face association that are a significant improvement on those of Yang *et al.* [Yang *et al.*, 2005b] described in Chapter 2.

The result is a labeled data set of faces, captured "in the wild." This data set displays a rich variety of phenomena found in real world face recognition tasks — significant variations in color, hairstyle, expression, etc. Equally interesting is that it does *not* contain large numbers of faces in highly unusual and seldom seen poses, such as upside down. Rather than building a database of face images by choosing arbitrary ranges of pose, lighting, expression and so on, we simply let the properties of a "natural" data source determine these parameters. We believe that in the long run, developing detectors, recognizers, and other computer vision tools around such a database will produce programs that work better in realistic everyday settings.

## 3.2 News Data Set

We have collected a data set consisting of approximately half a million news pictures and captions from Yahoo News over a period of roughly two years. Using Mikolajczyk's face detector [Mikolajczyk, 2002], we extract faces from these images. Using Cunningham *et al.*'s open source named entity recognizer [Cunningham *et al.*, 2002], we detect proper names in each of the associated captions. This gives us a set of faces and names resulting from each captioned picture. In each picture-caption pair, There may be several faces and several names. Furthermore, some faces may not correspond to any name, and some names may not correspond to any face. Our task is to assign one of these names or null (unnamed) to each detected face.

This collection differs from typical face recognition data sets in a number of important ways:

- **Pose, expression and illumination** vary widely. We often encounter the same face illuminated with markedly different colored light and in a very broad range of expressions. The parameters of the camera or post-processing add additional variability to the coloring of the photos. Spectacles and mustaches are common (Figure 3.5.4). There are wigs, images of faces on posters, differences in resolution and identikit pictures (e.g. Figure 3.5.4). Quite often there are multiple copies of the same picture (this is due to the way news pictures are prepared, rather than a collecting problem) or multiple pictures of the same individual in similar configurations. Finally, many individuals are tracked across time, adding an additional source of variability that has been shown to hamper face recognition substantially [Gross *et al.*, 2001].

- **Name frequencies** have the long tails that occur in natural language problems. We expect that face images follow roughly the same distribution. We have hundreds to thousands of images of a few individuals (e.g. *President Bush*), and a large number of individuals who appear only a few times or in only one picture (e.g. Figure 3.2). One expects real applications to have this property. For example, in airport security cameras a few people, security guards, or airline staff might be seen often, but the majority of people would appear infrequently. Studying how recognition systems perform under these circumstances is important.

- The sheer **volume** of available data is extraordinary. We have sharply reduced the number of face images we deal with by using a face detector that is biased to frontal faces and by requiring that faces be large and rectify properly. Even so, we have a data set that is comparable to, or larger than, the biggest available lab sets and is much richer in content. Computing kernel PCA and linear discriminants for a set this size requires special techniques (section 3.3.3.1).

One important difficulty is that our face detector cannot detect lateral or three-quarter views. This is a general difficulty with face detectors (all current face detectors either can detect only frontal views, or are significantly less reliable for views that are not frontal [Yang *et al.*, 2002]), but it means that our data set contains only frontal or near-frontal views. We speculate that methods like ours could be made to work to produce a similar data set if one had a face detector that was aspect insensitive, but do not know what performance penalty there would be. For extremely large data sets, we expect that there may be little penalty. This is because, in a sufficiently large data set, we might reasonably expect to see many aspects of each individual in contexts where there is little ambiguity. For smaller data sets the problem would be much more challenging and would require more sophisticated representations.

## 3.3 Finding and Representing Faces

To deal with the large quantity of data, we establish a pipeline that takes in images and outputs a description based on a rough alignment of facial features. Subsequently, we compare faces in this domain.

Our pipeline is as follows. For each news picture we,

1. Detect faces in the images (Section 3.3.1). We confine our activities to large, reliably detected faces, of which 44,773 are found.

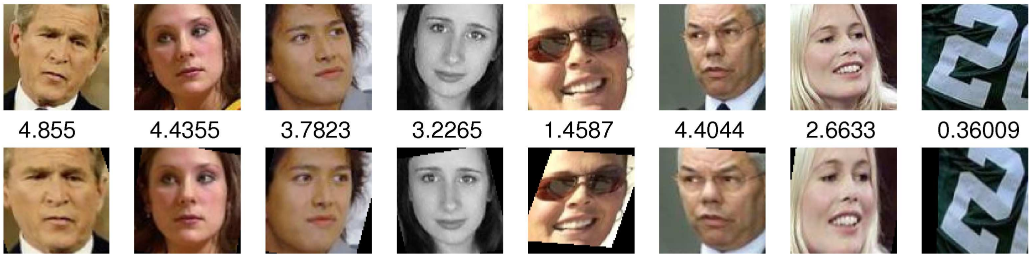


Figure 3.3: *The face detector can detect faces in a range of orientations, as the* **top row** *shows. Before clustering the face images we rectify them to a canonical pose* **bottom row**. *The faces are rectified using a set of SVM's trained to detect feature points on each face. Using gradient descent on SVM outputs, the best affine transformation is found to map detected feature points to canonical locations. Final rectification scores for each of these faces are shown* **center** *(where larger scores indicate better performance). This means that incorrect detections, like the rightmost image can be discarded because of their poor rectification scores.*

2. Rectify those faces to a canonical pose (Section 3.3.2). We discard faces where the rectifier cannot find good base points, resulting in 34,623 faces.

3. Identify faces with at least one proper name identified in the associated caption, leaving 30,281 faces.

4. Transform this set of faces into a representation suitable for the assignment task (Section 3.3.3).

### 3.3.1 Face detection

For face detection, we use Mikolajczyk's implementation [Mikolajczyk, 2002] of the face detector described by Schneiderman and Kanade [Schneiderman and Kanade, 2000]. To build this face detector, a training set of face and non-face images is used to determine the probability of a new image being a face. Each image in the training set is decomposed into a set of wavelet coefficients which are histogrammed so that each bin corresponds to a distinct set of coefficients; a probability model then determines whether the image is a face image or a non-face image. We threshold on face size (86x86 pixels or larger) and detection score to obtain 44,773 face images.

### 3.3.2 Rectification

The next stage in our pipeline is an alignment step. While the detector detects only frontal or near frontal faces, these faces are still subject to small out of plane rotations and significant in-plane rotations and scales. We will use an appearance feature to compare face images, and so would like to reduce within-class variance and increase between-class variance. Within-class variance in appearance features can be significantly reduced by moving each face image to a canonical frame (where eyes, nose, mouth, etc. lie close to canonical locations), a procedure we call *rectification*. We will rectify by using a novel procedure to identify a set of base points in the image, then apply a full plane affine transformation to move these base points to canonical locations. Images where base points can not be identified sufficiently well will be rejected.

Notice that rectification could suppress features that help identify individuals. For example, some individuals have larger faces than others do, and rectification suppresses this property, thereby reducing between-class variance. In this application, the suppression of within-class variance obtained by rectifying faces seems to outweigh the loss of between class variance. We speculate that in a sufficiently large data set, rectification may be unnecessary, because one would have enough examples of any individual's face in any view; we have no reason to believe our data set is anywhere large enough for this to apply.

#### 3.3.2.1 Identifying Base Point Locations

We train five support vector machines (SVMs) as feature detectors for several features on the face (corners of the left and right eyes, corners of the mouth, and the tip of the nose) using a training set consisting of 150 hand clicked faces. We use the geometric blur feature of Berg and Malik [Berg and Malik, 2001b] applied to gray-scale patches as the features for our SVM.

The geometric blur descriptor first produces sparse channels from the grey scale image. In this case, these are half-wave rectified oriented edge filter responses at three orientations, yielding six channels. Each channel is blurred by a spatially varying Gaussian with a standard deviation proportional to the distance to the feature center. The descriptors are then sub-sampled and normalized. Initially image patches were used as input to the feature detectors, but replacing patches with the geometric blurred version of the patches produced significant gains in rectification accuracy. Using geometric blur features instead of raw image patches was a necessary step to making our rectification system effective.

We compute the output value for each SVM at each point in the entire image and multiply with a weak prior on location for each feature. This produces a set of five feature maps, one for each base point. The initial location of each base point is obtained as the maximal point of each map.

#### 3.3.2.2 Computing the Rectification

We compute an initial affine map from canonical feature locations to the initial locations for the base points using least squares. However, accepting a small decrease in the SVM response for one base point may be rewarded by a large increase in the response for another. We therefore maximize the sum of SVM responses at mapped canonical feature locations using gradient descent, with the initial affine map as a start point. The image is then rectified using the resulting map, and the value of the optimization problem is used as a score of the rectification. The value indicates how successful we have been at finding base points; a small score suggests that there is no set of points in the image that (a) looks like the relevant face features and (b) lies near to the result of an affine map applied to the canonical points.

We filter our data set by removing images with poor rectification scores, leaving 34,623 face images. This tends to remove the face detectors false positives (Figure 3.2; center number – larger numbers indicate a better score). Each face is then automatically cropped to a region surrounding the eyes, nose and mouth in the canonical frame, to eliminate effects of background on recognition. The RGB pixel values from each cropped face are concatenated into a vector and used as a base representation from here on.

### 3.3.3 Face Representation

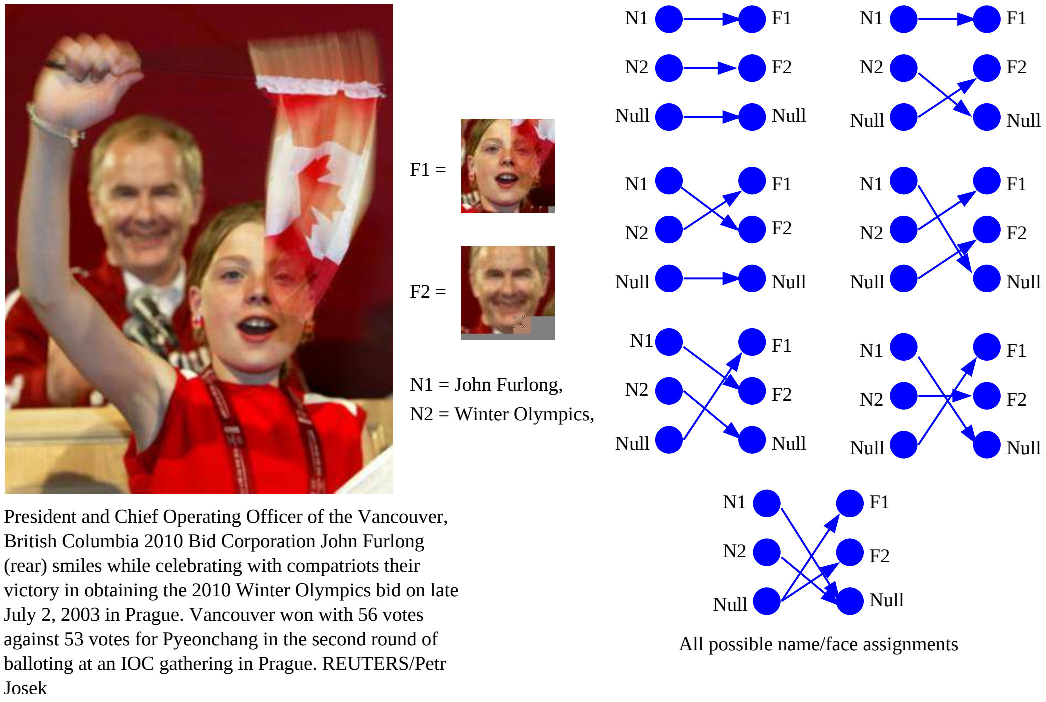
We wish to represent faces appearances as vectors in a space where, if one uses Euclidean distance between points, examples of the same face are close together and examples of different faces are far apart. We must identify components of the base representation that tend to be similar for all faces, and discard them (or, better, keep components of the base variation that vary strongly over the data set). Of these, we must keep those that tend to co-vary with identity.

We use kernel principal component analysis (kPCA; see [Schölkopf *et al.*, 1998]) to identify components of the base representation that vary strongly over the data set. The result is a vector of kernel principal components. We apply linear discriminant analysis (LDA; see, for example [Hastie *et al.*, 2001]) to these vectors, to obtain a feature vector. Kernel principal component analysis is a standard method of dimension reduction that has been shown to be effective for face recognition (see, for example [Liu and Chen, 1999; Kim *et al.*, 2002; Yang *et al.*, 2000; Yang *et al.*, 2005a; Lu *et al.*, 2003; Zhao *et al.*, 2000]; Yang compares with principal components and with linear discriminant analysis and shows a strong advantage for kPCA combined with LDA [Yang, 2002]).

#### 3.3.3.1 Kernel Principal Components and the Nyström Approximation

**Kernel Principal Components Analysis** requires the following steps:

- Compute a kernel matrix, K, where $K_{ij} = K(\text{image}_i, \text{image}_j)$ is the value of a kernel function comparing $\text{image}_i$ and $\text{image}_j$. We use a Gaussian kernel with sigma set to produce reasonable kernel values.
- Center the kernel matrix in feature space by subtracting off average row, average column and adding on average element values.



President and Chief Operating Officer of the Vancouver, British Columbia 2010 Bid Corporation John Furlong (rear) smiles while celebrating with compatriots their victory in obtaining the 2010 Winter Olympics bid on late July 2, 2003 in Prague. Vancouver won with 56 votes against 53 votes for Pyeonchang in the second round of balloting at an IOC gathering in Prague. REUTERS/Petr Josek

Figure 3.4: *To assign faces to names, we evaluate all possible assignments of faces to names and choose either the maximum likelihood assignment or form an expected assignment. Here we show a typical data item (***left***), with its detected faces and names (***center***). The set of possible correspondences for this data item are shown at* **right**. *This set is constrained by the fact that each face can have at most one name assigned to it and each name can have at most one face assigned, but any face or name can be assigned to Null. Our named entity recognizer occasionally detects phrases like "Winter Olympics" which do not correspond to actual people. These names are assigned low probability under our language model, making their assignment unlikely. EM iterates between computing the expected value of the set of possible face-name correspondences and updating the face clusters and language model. Unusually, we can afford to compute all possible face-name correspondences since the number of cases is small. For this item, we correctly choose the best matching "F1 to Null", and "F2 to N1".*

- Compute an eigendecomposition of K, and project onto the normalized eigenvectors of K.

Writing $N$ for the number of data items, we have an $NxN$ kernel matrix. In our case, $N = 34,623$, and we cannot expect to evaluate every entry of the matrix.

We cannot use incomplete Cholesky decomposition (which would give a bound on the approximation error [Golub and Loan, 1996]), because that would require accessing all images for each column computation. However, the kernel matrix must have relatively low column rank; if it did not, there would be generalization problems, because it would be difficult to predict a column from the other columns (see [Williams and Seeger, 2001]). This suggests using the Nyström approximation method, which will be accurate if the matrix does have low column rank, and which allows the images to be accessed only once in a single batch rather than once for each column computation (cf. [Williams and Seeger, 2001; Fowlkes *et al.*, 2004]).

The Nyström method partitions the kernel matrix as:

$$K = \begin{bmatrix} A & B \\ B^T & C \end{bmatrix}$$

with $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{(N-n) \times n}$ and $C \in \mathbb{R}^{(N-n) \times (N-n)}$. To obtain $A$ and $B$ by selecting a base set $\mathcal{B}$ of the set of images $\mathcal{I}$ (in our case, 1000 images selected uniformly and at random). Then

$$A_{uv} = K(\mathrm{image}_u, \mathrm{image}_v) \text{ for } \mathrm{image}_u \in \mathcal{B},\ \mathrm{image}_v \in \mathcal{B},$$

where $K(\cdot, \cdot)$ is the kernel function, and

$$B_{lm} = K(\mathrm{image}_l, \mathrm{image}_m) \text{ for } \mathrm{image}_l \in \mathcal{B},\ \mathrm{image}_m \in \mathcal{I}.$$

Now Nyström's method approximates $K$ with the matrix obtained by replacing $C$ with $\hat{C} = B^T A^{-1} B$, yielding $\hat{K} = \begin{bmatrix} A & B \\ B^T & \hat{C} \end{bmatrix}$.

**Centering:** we center $\hat{K}$ as usual in kPCA, by writing $1_N$ for an $Nx1$ vector of ones, and then computing

$$\tilde{K} = \hat{K} - \frac{1}{N} 1_N \hat{K} - \frac{1}{N} \hat{K} 1_N + \frac{1}{N^2} 1_N \hat{K} 1_N.$$

Note that this is simplified by the fact that $\hat{K}$ is symmetric, and by observing that

$$\hat{K} 1_N = \begin{bmatrix} A 1_n + B 1_{N-n} \\ B^T 1_n + B^T A^{-1} B 1_{N-n} \end{bmatrix}.$$

It is convenient to write $\tilde{K} = \begin{bmatrix} \tilde{A} & \tilde{B} \\ \tilde{B}^T & \tilde{C} \end{bmatrix}$ where the dimensions of $\tilde{A}$ are those of $A$, etc.

**Approximate eigenvectors:** Let $\tilde{A}^{\frac{1}{2}}$ be the square root of $\tilde{A}$, and $S = \tilde{A} + \tilde{A}^{-\frac{1}{2}} \tilde{B} \tilde{B}^T \tilde{A}^{-\frac{1}{2}}$. Diagonalize $S$ as $S = U_s \Lambda_s U_s^T$. Then $\tilde{K}$ is diagonalized by

$$V = \begin{bmatrix} \tilde{A} \\ \tilde{B}^T \end{bmatrix} \tilde{A}^{-\frac{1}{2}} U_s \Lambda_s^{-\frac{1}{2}}.$$

Then we have $\tilde{K} = V \Lambda_s V^T$ and $V^T V = I$. Given this decomposition of $\tilde{K}$ we proceed as usual for kPCA, by normalizing the eigenvectors $V$ and projecting $\tilde{K}$ onto the normalized eigenvectors. This gives a dimensionality reduction of our images that makes the discrimination task easier.

**Quality of approximation:** It is difficult to verify that the approximation is accurate directly, because we are unable to form $K$, let alone evaluate its eigenvectors. However, we have some evidence the approximation is sound. First, the eigenvalues of $\tilde{A}$ tend to fall off quickly, despite the fact that the elements of $\mathcal{B}$ are chosen at random. This suggests that $K$ does, indeed, have low rank. Second, in practice the representation is quite effective.

#### 3.3.3.2 Linear Discriminants Analysis

The $i$'th face image is now represented by its vector of kernel principal components $\mathbf{v}_i$. Assume that the identity of each face is known. Then we can compute linear discriminants for these vectors in the usual way [Hastie *et al.*, 2001], writing $m$ for the number of classes, $\mathcal{C}_l$ for the set of elements in class $l$, $N_l$ for the number of samples in class $l$, $\mu_l$ for the mean of class $l$, and computing the within class variance $W$ and between class variance $B$ as

$$\begin{aligned} W &= \sum_{i=1}^{m} \sum_{x_j \in C_i} (x_j - \mu_i)(x_j - \mu_i)^T \\ B &= \sum_{i=1}^{m} N_i(\mu_i - \mu)(\mu_i - \mu)^T. \end{aligned}$$

LDA computes the projection $\mathbf{w}$ that maximizes the ratio,

$$\mathbf{w}_{opt} = argmax_{\mathbf{w}} \frac{\mathbf{w}^T B \mathbf{w}}{\mathbf{w}^T W \mathbf{w}},$$

by solving the generalized eigenvalue problem:

$$B\mathbf{w} = \lambda W\mathbf{w}.$$

We obtain a set of projection directions, which we stack into a matrix

$$W = \begin{bmatrix} \mathbf{w}_1^T \\ \dots \\ \mathbf{w}_n^T \end{bmatrix}.$$

The final representation for the $i$'th face is now $\mathbf{f}_i = W\mathbf{v}_i$. Notice that, in this coordinate system, the Mahalanobis distance to a class mean is given by the Euclidean distance.

Of course, not all of our images are labeled. However, we do have a subset of data where there was a single face detected in an image with a single name in the caption. We use these images to compute the linear discriminants.

## 3.4 Name Assignment by Simple Clustering

We have a set of $b$ "bags", each containing $F$ faces and $N$ names. We wish to identify correspondences between names and faces within each bag. Each name in a bag can belong to at most one face. If we had a cluster of face vectors for each individual, we could allocate the name whose cluster center is closest to each face (this would also require allocating each name only once, and not naming a face if all cluster centers are too far away). With the allocations, we could re-estimate cluster centers, and so on. This method is analogous to k-means clustering (see the textbook account in [Duda *et al.*, 2001], for example). There are advantages to generalizing the method with a probabilistic model: we can perform soft allocations of names to faces; we will be able to benefit from text features (section 3.5); and it is easier to reason explicitly about both faces without names and exclusion between names. To build a probabilistic model, we regard correspondence as a hidden variable, build a generative model for a bag given correspondence, obtain a complete data log-likelihood, and then estimate with the expectation-maximization (EM) algorithm. A variant estimation procedure, where one chooses the best correspondence rather than a weighted average of correspondences, performs better in practice.

### 3.4.1 A Generative Model for Bags

To obtain a bag of data, we first draw the number of faces $F$ from a distribution $P(F)$ and the number of names $N$ from a distribution $P(N)$. We then generate $N$ names $\mathbf{n}_i$, each with a context $\mathbf{c}_i$, as IID samples of $P(\mathbf{n}, \mathbf{c})$. The context is of no interest at this point, but we will use the idea below. In turn, each name and its context generates a binary variable *pictured*, which determines whether the name will generate a face in the image. For each name for which $\textit{pictured} = 1$ (the total number of such names cannot exceed $F$), a face $\mathbf{f}_i$ is generated from the conditional density $P(\mathbf{f}|\mathbf{n}_i, \theta)$, where $\theta$ are parameters of this distribution which will need to be estimated. The remaining faces are generated as IID samples from a distribution $P(f)$. We cannot observe which name generated which face, and must encode this information with a hidden variable.

For the moment, assume that we know a correspondence from names to faces for a particular bag. This is encoded as a partition of the names $\mathcal{N}$ in the bag into two sets, $\mathcal{D}$ being the names that generate faces and $\mathcal{U}$ being the names that do not, and a map $\sigma$, which takes a name index $\alpha$ to a face index $\sigma(\alpha)$. For convenience, we write the set of faces in the bag as $\mathcal{F}$. The likelihood of the bag is then

$$L(\theta, \sigma) = P(N)P(F) \left( \prod_{\alpha \in \mathcal{D}} P(\mathbf{f}_{\sigma(\alpha)} | \mathbf{n}_\alpha, \theta) \right) \left( \prod_{\gamma \in \mathcal{F} - \sigma(\mathcal{D})} P(\mathbf{f}_\gamma) \right) \left( \prod_{u \in \mathcal{N}} P(\mathbf{n}_u, \mathbf{c}_u) \right).$$

Notice that *pictured* does not appear explicitly here (it is implicit in the form of the likelihood).

**Implementation details:** $F$ and $N$ typically vary between one and five, and we see no advantage in regarding larger bags as different from smaller ones. We therefore regard $P(N)$ and $P(F)$ as uniform over the range of those variables, and so they play no part in the estimation. We use a uniform prior over names and contexts ($P(n_u, c_u)$), too, and they too play no further part in the estimation. We regard $P(\mathbf{f}_\gamma)$ as uniform; we will use only its logarithm, which will be a constant parameter. Our choice of coordinate system means we can regard $P(\mathbf{f}|\mathbf{n}, \theta)$ as a normal distribution, with mean $\theta_{\mathbf{n}}$ — which gives one cluster center per name — and covariance $\sigma_f^2 \mathcal{I}$. We choose a sigma to produce reasonable values ($\sigma = 0.1$), but do not fit this explicitly.

### 3.4.2 Estimation with EM

Of course, the correspondence between faces and names is unknown. However, for each bag there is a small set of possible correspondences. We construct an indicator variable $\delta(m, n)$, where

$$\delta(m,n) = \left\{ \begin{array}{ll} 1 & \text{if the } n\text{'th correspondence for the } m\text{'th data item actually occurs} \\ 0 & \text{otherwise} \end{array} \right\}$$

This indicator variable is unknown, but we will estimate it. If it were known, we could write the log-likelihood of the data set as

$$\sum_{m \in \text{data}} \left( \sum_{n \in \text{correspondences for the } m\text{'th data item}} \delta(m,n) \log L(\theta, \sigma_n) \right) .$$

We now estimate $\theta$ and $\delta(m, n)$ with EM. It is natural to regard this as a soft-count procedure. At the $i$'th iteration, we first estimate the expected value of the $\delta(m, n)$ conditioned on the previous parameter estimate $\theta^{(i)}$, then estimate $\theta^{(i+1)}$ by substituting these expected values in the likelihood and maximizing.

### 3.4.3 Estimation with Maximal Assignment

If the model is an accurate reflection of the data, then it is natural to average out hidden variables (rather than, say, simply maximizing over them), and doing so should give better estimates (e.g. [McLachlan and Krishnan, 1996]). However, expectation maximization is regularly outperformed in vision problems by the simpler — and statistically non-optimal — procedure of maximizing over the hidden variables (for example, randomized search for correspondences in fundamental matrix estimation [Torr and Murray, 1997; Torr and Zisserman, 1998]). We conjecture that this is because local models — in our case, $p(\mathbf{f}|\mathbf{n}, \theta)$ — may exaggerate the probability of large errors, and so the expectation step could weight poor correspondences too heavily.

Maximal assignment iterates two steps:

- Set the $\delta(m, n)$ corresponding to the maximum likelihood assignment to 1 and all others to 0.
- Maximize the parameters $P(f|n, \theta_f)$ using counts.

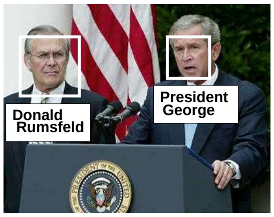
In practice, maximal assignment leads to better name predictions (section 3.6).

## 3.5 Clustering with Context Understanding

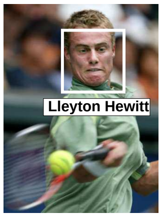
Up to this point, we've treated the caption as a bag of words. However, the context of the name is important. For example, consider the caption:

> *Sahar Aziz, left, a law student at the University of Texas, hands the business card identifying Department of the Army special agent Jason D. Treesh to one of her attorneys, Bill Allison, right, during a news conference on Friday, Feb. 13, 2004, in Austin, Texas. ... In the background is Jim Harrington, director of the Texas Civil Rights Project. (AP Photo/Harry Cabluck)*

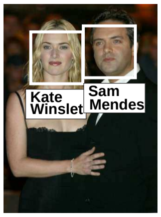
From the caption alone, we expect to see Sahar Aziz, Bill Allison and Jim Harrington in the picture, and we do not expect to see Jason D. Treesh. This suggests that a language model can exclude some names from consideration. In this section, we show how to build such a model into our framework (section 3.5.1); describe two plausible



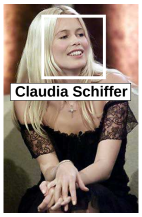
**President George** W. Bush makes a statement in the Rose Garden while Secretary of **Defense Donald Rumsfeld** looks on, July 23, 2003. Rumsfeld said the United States would release graphic photographs of the dead sons of **Saddam Hussein** to prove they were killed by American troops. Photo by Larry Downing/Reuters



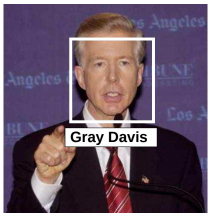
World number one **Lleyton Hewitt** of Australia hits a return to **Nicolas Massu** of Chile at the Japan Open tennis championships in Tokyo October 3, 2002. REUTERS/Eriko Sugita



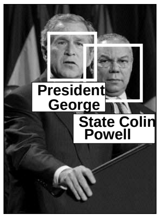
British director **Sam Mendes** and his partner actress **Kate Winslet** arrive at the London premiere of 'The Road to Perdition', September 18, 2002. The films stars **Tom Hanks** as a Chicago hit man who has a separate family life and co-stars **Paul Newman** and Jude Law. REUTERS/Dan Chung



German supermodel **Claudia Schiffer** gave birth to a baby boy by Caesarian section January 30, 2003, her spokeswoman said. The baby is the first child for both Schiffer, 32, and her husband, British film producer **Matthew Vaughn**, who was at her side for the birth. Schiffer is seen on the German television show 'Bet It...?!' ('Wetten Dass...?!') in Braunschweig, on January 26, 2002. (Alexandra Winkler/Reuters)



Incumbent California Gov. **Gray Davis** (news - web sites) leads Republican challenger **Bill Simon** by 10 percentage points – although 17 percent of voters are still undecided, according to a poll released October 22, 2002 by the Public Policy Institute of California. Davis is shown speaking to reporters after his debate with Simon in Los Angeles, on Oct. 7. (Jim Ruymen/Reuters)



US **President George** W. Bush (L) makes remarks while Secretary of **State Colin Powell** (R) listens before signing the US Leadership Against HIV /AIDS , Tuberculosis and Malaria Act of 2003 at the Department of State in Washington, DC. The five-year plan is designed to help prevent and treat AIDS, especially in more than a dozen African and Caribbean nations(AFP/Luke Frazza)

Figure 3.5: *Given an input image and an associated caption (images above and captions to the right of each image), our system automatically detects faces (white boxes) in the image and possible name strings (bold). We use a clustering procedure to build models of appearance for each name and then automatically label each of the detected faces with a name if one exists. These automatic labels are shown in boxes below the faces. Multiple faces may be detected and multiple names may be extracted, meaning we must determine who is who (e.g., the picture of* Claudia Schiffer*).*

such models (section 3.5.2); and describe two estimation methods (sections 3.5.3 and 3.5.4).

### 3.5.1 A Generative Model for Bags

Many of the caption phenomena that suggest a person is present are relatively simple, and a simple language model should exclude some names from consideration. There

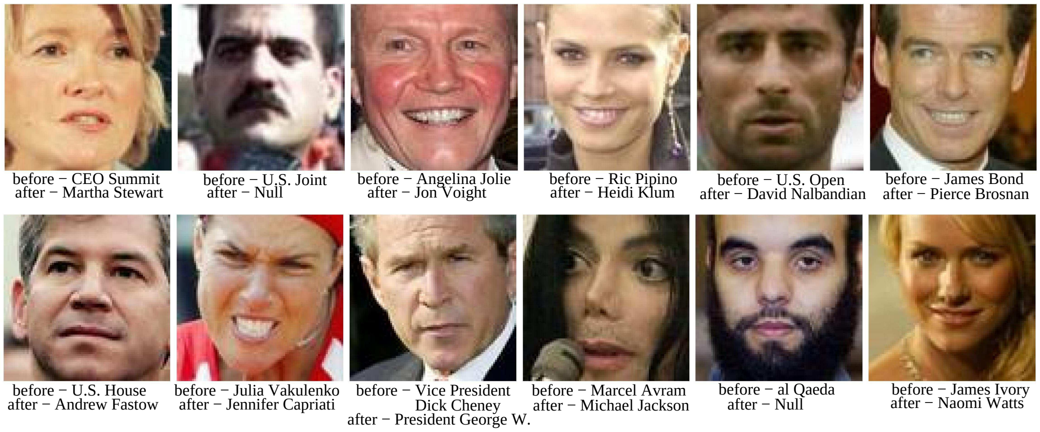
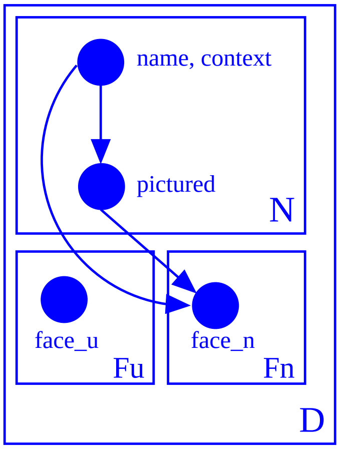


Figure 3.6: *This figure shows some example pictures with names assigned using our raw clustering procedure* (**before**) *and assigned using a correspondence procedure with incorporated language model* (**after**)*. Our named entity recognizer sometimes detects incorrect names like "CEO Summit", but the language model assigns low probabilities to these names making their assignment unlikely. When multiple names are detected like "Julia Vakulenko" and "Jennifer Capriati", the probabilities for each name depend on their context. The caption for this picture reads "American Jennifer Capriati returns the ball to her Ukrainian opponent Julia Vakulenko in Paris during..." "Jennifer Capriati" is assigned to the face given the language model because the context in which she appears (beginning of the caption followed by a present tense verb) is more likely to be pictured than that of "Jennifer Capriati" (middle of the caption followed by a preposition). For pictures such as the one above ("al Qaeda" to "Null") where the individual is not named, the language model correctly assigns "Null" to the face. As table 3.1 shows, incorporating a language model improves our face clusters significantly.*

are three important cases. First, our named entity recognizer occasionally marks phrases like "United Nations" as proper names. We can determine that these names do not refer to depicted people because they appear in quite different linguistic contexts from the names of actual people. Second, caption writers tend to name people who are actually depicted earlier in the caption. Third, caption writers regularly use depiction indicators such as "left", "(R)", "background".

Our generative model can be enhanced in a relatively straightforward way to take advantage of these phenomena. In section 3.4, we encoded *pictured* implicitly in the correspondence. We must now recognize *pictured* as a random variable, and incorporate it into the model. Doing so yields the following generative model:

To generate a data item:



1. Choose $N$, the number of names, and $F$, the number of faces.
2. Generate $N$ *name, context* pairs.
3. For each of these *name, context* pairs, generate a binary variable *pictured* conditioned on the context alone (from $P(\textit{pictured}|\text{context}, \theta_c)$).
4. For each *name, context* pair where *pictured* $=$ 1, generate a face from $P(\mathbf{f}|\mathbf{n}, \theta_f, \textit{pictured} = 1)$.
5. Generate $F - \sum \textit{pictured}$ other faces from $P(\mathbf{f})$.

We follow section 3.4.1 to obtain an expression for the likelihood of a bag conditioned on known correspondence. To obtain a bag of data, we first draw the number of faces $F$ from a distribution $P(F)$ and the number of names $N$ from a distribution $P(N)$. We then generate $N$ names $\mathbf{n}_i$, each with a context $\mathbf{c}_i$, as IID samples of $P(\mathbf{n}, \mathbf{c})$. In turn, each name and its context generates a binary variable *pictured*, which determines whether the name will generate a face in the image, from $P(\textit{pictured}|\text{context}, \theta_c)$. For each name for which $\textit{pictured} = 1$ (the total number of such names cannot exceed $F$), a face $\mathbf{f}_i$ is generated from the conditional density $P(\mathbf{f}|\mathbf{n}_i, \theta, \textit{pictured} = 1)$, where $\theta$ are parameters of this distribution which will need to be estimated. The remaining faces are generated as IID samples from a distribution $P(f)$. We cannot observe which name generated which face, and must encode this information with a hidden variable.

For the moment, assume that we know a correspondence from names to faces for a particular bag. Notice that this implicitly encodes *pictured*: names that have corresponding faces have $pictured = 1$, and the others have $pictured = 0$. The correspondence is encoded as a partition of the names $\mathcal{N}$ in the bag into two sets, $\mathcal{D}$ being the names that generate faces and $\mathcal{U}$ being the names that do not, and a map $\sigma$, which takes a name index $\alpha \in$ to a face index $\sigma(\alpha)$ or to Null if the name is not *pictured*. For convenience, we write the set of faces in the bag as $\mathcal{F}$. The likelihood of the bag is then

$$\begin{aligned} L(\theta_c, \theta_f, \sigma) \quad = \quad & P(N)P(F)\left(\prod_{u \in \mathcal{N}} P(\mathbf{n}_u, \mathbf{c}_u)\right) * \\ & \left(\prod_{\alpha \in \mathcal{D}} P(\mathbf{f}_{\sigma(\alpha)}|\mathbf{n}_\alpha, \theta_f, pictured = 1)P(pictured = 1|\mathbf{c}_\alpha, \theta_c)\right) * \\ & \left(\prod_{\gamma \in \mathcal{F}-\sigma(\mathcal{D})} P(\mathbf{f}_\gamma)\right)\left(\prod_{\beta \in \mathcal{U}}(P(pictured = 0|\mathbf{c}_\beta, \theta_c))\right). \end{aligned}$$

We need a model of $P(pictured|\mathbf{c}, \theta_c)$. Once we have a model, we must estimate $\theta_f$ (the parameters of the distribution generating faces from names) and $\theta_c$ (the parameters of the distribution generating *pictured* from context). All parameters are treated as before (section 3.4.1), except now we also fit a model of name context, $P(pictured = 1|\mathbf{c}, \theta_c)$.

### 3.5.2 Language Representation

We have explored two models for $P(pictured|\mathbf{c}, \theta_c)$. First, a naive Bayes model in which each of the different context cues is assumed independent given the variable pictured, and second, a maximum entropy model which relaxes these independence assumptions.

#### 3.5.2.1 Naive Bayes Model

For a set of context cues ($C_i$, for $i \in 1, 2, ...n$), our Naive Bayes model assumes that each cue is independent given the variable *pictured*. Using Bayes rule, the probability of being pictured given the cues is

$$\begin{aligned}
P(pictured|C_1, C_2, ...C_n) &= \frac{P(C_1, ...C_n|pictured)P(pictured)}{P(C_1, ..., C_n)} \\
&= \frac{P(C_1|pictured)...P(C_n|pictured)P(pictured)}{P(C_1, ..., C_n)} \\
&= \frac{P(pictured)}{P(C_1, ..., C_n)} \prod_i \frac{P(pictured|C_i)P(C_i)}{P(pictured)} \\
&= \frac{1}{Z} \frac{P(pictured|C_1)...P(pictured|C_n)}{P(pictured)^{n-1}}.
\end{aligned}$$

Line 1 is Bayes Rule. Line 2 follows from the naive Bayes assumption. Line 3 follows again by Bayes Rule. The $Z$ in line 4 is dependent only on the cues $C_1, ..., C_n$. We compute $P(pictured|C_1, ..., C_n)$ and $P(notpictured|C_1, ..., C_n)$ ignoring the Z term, and then normalize so that $P(pictured|C_1, ..., C_n)$ and $P(notpictured|C_1, ..., C_n)$ sum to 1.

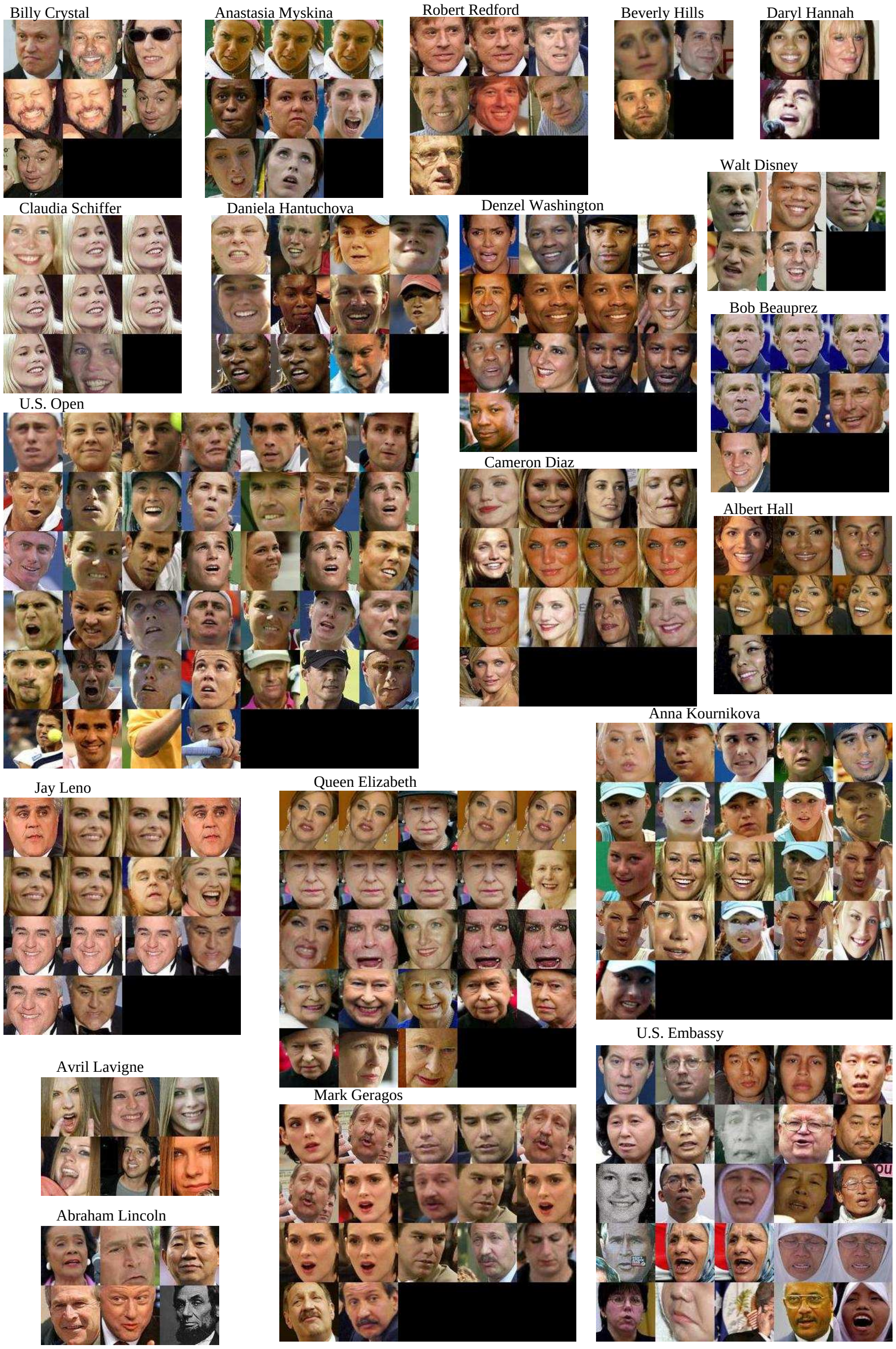


Figure 3.7: *Example clusters found using our basic clustering method (see section 3.4 for details). Note that the names of some clusters are not actual people's names (e.g. "U.S. Open", "Walt Disney") and that there are clusters with multiple errors ("Queen Elizabeth", "Jay Leno").*

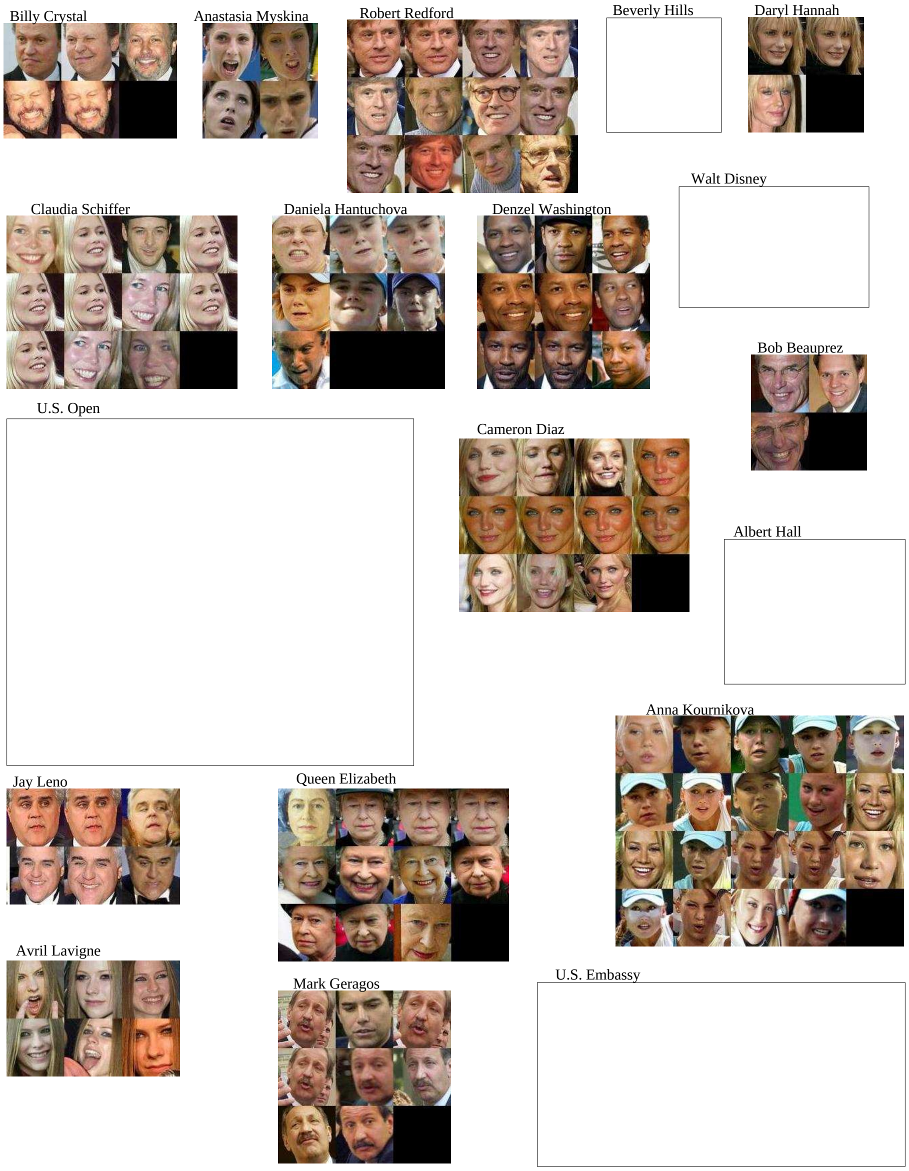


Figure 3.8: *The clusters of Figure 3.7 are improved through the use of language understanding (see section 3.5 for details). The context of a name within the caption often provides clues as to whether the name is depicted. By analyzing the context of detected names, our improved clustering gives the more accurate clusters seen above. The named entity recognizer occasionally marks some phrases like "U.S. Open" and "Albert Hall" as proper names. By analyzing their context within the caption, our system correctly determined that no faces should be labeled with these phrases. Incorporating language information also makes some clusters larger ("Robert Redford"), and some clusters more accurate ("Queen Elizabeth", "Bob Beauprez").*

**Implementation details:** The cues we use are: the part of speech tags of the word immediately prior to the name and immediately after the name within the caption (modeled jointly); the location of the name in the caption; and the distances to the nearest ", ", ".", "(", ")", "(L)", "(R)", "(C)", "left", "right", and "center" (these distances are quantized and binned into histograms). We tried adding a variety of other language model cues, but found that they did not increase assignment accuracy.

We use one distribution for each possible context cue, and assume that context cues are independent when modeling these distributions (because we lack enough data to model them jointly).

#### 3.5.2.2 Maximum Entropy Model

Maximum entropy models have been used extensively in natural language systems (e.g. [Berger *et al.*, 1996]). Maximum likelihood applied to these models — otherwise known as conditional exponential models — results in a model that is consistent with a chosen set of observed statistics of the data, but which otherwise maximizes entropy. An attraction of maximum entropy models is that they give a nice way of modeling a conditional distribution with a large number of features without having to observe every combination of those features. They also do not assume independence of features as the Naive Bayes model does and model conditional distributions directly rather than through the use of Bayes' rule.

Recall *pictured* is a binary variable. We are modeling $P(pictured|\mathbf{c}, \theta_c)$. We encode context as a binary vector, where an element of the vector is 1 if the corresponding context cue is true and zero if it is not. For the $i$'th context cue we define two indicator functions

$$f_i(x,y) = \begin{cases} 1 & \text{if } x(i) = 1 \text{ and } y = 0; \\ 0 & \text{otherwise.} \end{cases}$$

$$f_{2i}(x,y) = \begin{cases} 1 & \text{if } x(i) = 1 \text{ and } y = 1; \\ 0 & \text{otherwise.} \end{cases}$$

Our model is now

$$p(pictured|x, \theta_c) \propto exp \sum_j \theta_{c,j} f_j(x, pictured)$$

where $\theta_{c,j}$ is the weight of indicator function $j$.

**Implementation details:** We use the same cues as before except instead of binning the distance to the nearest ",", ".", "(", ")", "(L)", "(R)", "(C)", "left", "right", and "center" the corresponding cue is true if the the string is within 3 words of the name. We also define a separate cue for each binned location corresponding to the binned location cue used for the Naive Bayes model. For the Maximum Entropy model we also add cues looking for specific strings ("pictured", "shown", "depicted" and "photo").

### 3.5.3 Estimation with EM

EM is computed as described in section 3.4.2. The differences for each context model are described in section 3.5.3.1 and section 3.5.4.

#### 3.5.3.1 Estimating Depiction with Naive Bayes

We update the distributions, $P(pictured|C_i)$ and $P(pictured)$, at each iteration of EM process using maximum likelihood estimates based on soft counts. $P(pictured|C_i)$ is updated by how often each context appears describing an assigned name, versus how often that context appears describing an unassigned name. $P(pictured)$ is computed using soft counts of how often names are pictured versus not pictured.

Some indications of a name being pictured learned by the Naive Bayes model were: 1. The closer the name was to the beginning of the caption, the more likely it was of being pictured, 2. The "START" tag directly before the name was a very good indicator of the name being pictured, 3. Names followed by different forms of present tense verbs were good indications of being pictured, 4. The name being followed by "(L)", "(R)" and "(C)" were also somewhat good indications of picturedness.

#### 3.5.3.2 Estimating Depiction with Maximum Entropy Models

To find the maximum likelihood $p(y|x)$, we use improved iterative scaling, the standard algorithm for finding maximum entropy distributions, again using soft counts. Details of this model and algorithm are described in [Berger *et al.*, 1996].

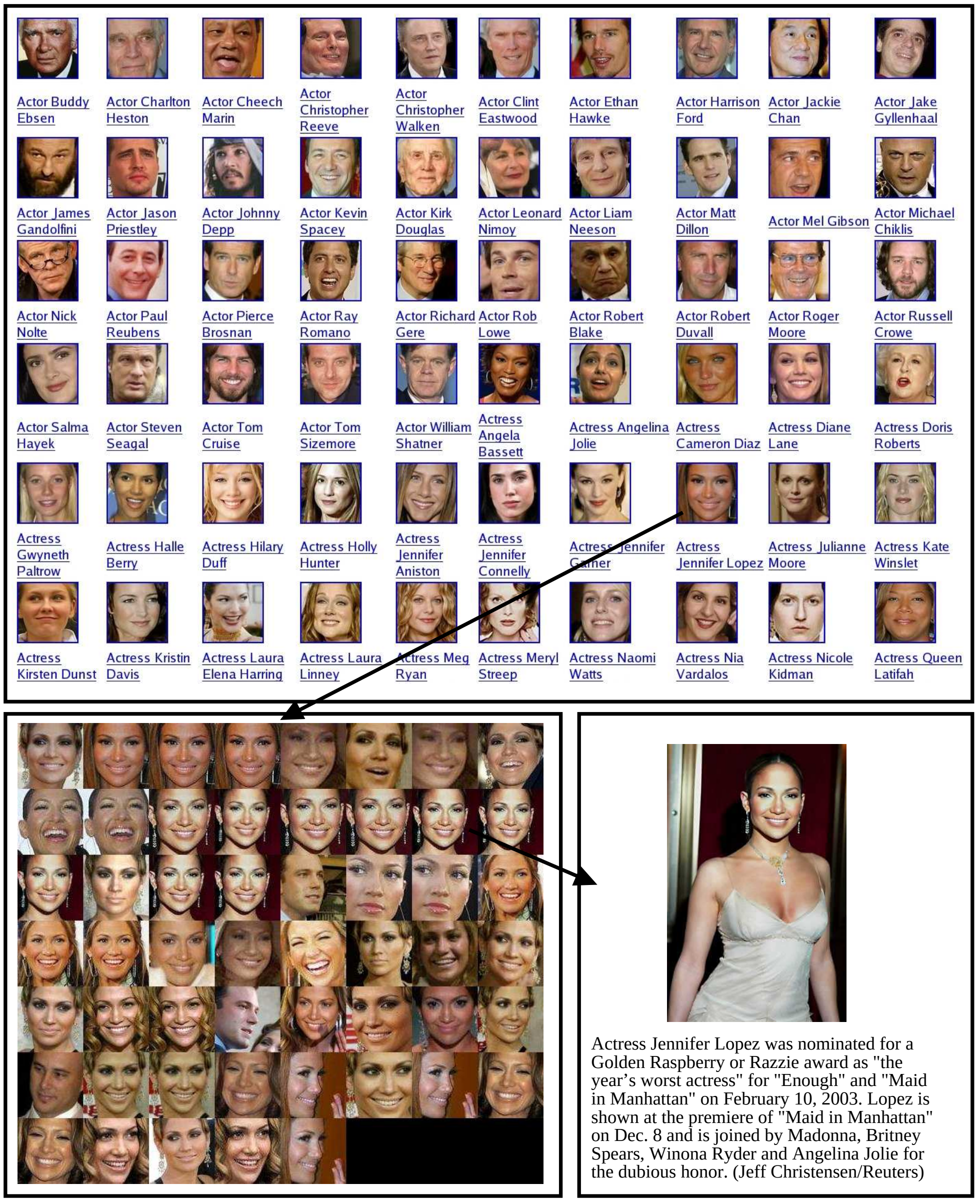
### 3.5.4 Estimation with Maximal Assignment

Estimation with maximal assignment is as before. However, both naive Bayes and maximum entropy language models no longer use soft counts. In effect, maximal assignment chooses a single correspondence, and so specifies which names are depicted.

| |
|---|
| **IN Pete Sampras IN** of the U.S. celebrates his victory over Denmark's **OUT Kristian Pless OUT** at the **OUT U.S. Open OUT** at Flushing Meadows August 30, 2002. Sampras won the match 6-3 7- 5 6-4. REUTERS/Kevin Lamarque |
| Germany's **IN Chancellor Gerhard Schroeder IN**, left, in discussion with France's **IN President Jacques Chirac IN** on the second day of the EU summit at the European Council headquarters in Brussels, Friday Oct. 25, 2002. EU leaders are to close a deal Friday on finalizing entry talks with 10 candidate countries after a surprise breakthrough agreement on Thursday between France and Germany regarding farm spending.(AP Photo/European Commission/HO) |
| 'The Right Stuff' cast members **IN Pamela Reed IN**, (L) poses with fellow cast member **IN Veronica Cartwright IN** at the 20th anniversary of the film in Hollywood, June 9, 2003. The women played wives of astronauts in the film about early United States test pilots and the space program. The film directed by **OUT Philip Kaufman OUT**, is celebrating its 20th anniversary and is being released on DVD. REUTERS/Fred Prouser |
| Kraft Foods Inc., the largest U.S. food company, on July 1, 2003 said it would take steps, like capping portion sizes and providing more nutrition information, as it and other companies face growing concern and even lawsuits due to rising obesity rates. In May of this year, San Francisco attorney **OUT Stephen Joseph OUT**, shown above, sought to ban Oreo cookies in California – a suit that was withdrawn less than two weeks later. Photo by Tim Wimborne/Reuters REUTERS/Tim Wimborne |

Figure 3.9: *Our new procedure gives us not only better clustering results, but also a natural language classifier which can be tested separately.* **Above:** *a few captions where detected names have been labeled with IN (pictured) and OUT (not pictured) using our learned language model. Our language model has learned which contexts have high probability of referring to pictured individuals and which contexts have low probabilities. We can use this model to evaluate the context of each new detected name and label it as IN or OUT. We observe an 85% accuracy of labeling who is portrayed in a picture using only our language model. The top 3 labelings are all correct. The last incorrectly labels "Stephen Joseph" as not pictured when in fact he is the subject of the picture. Some contexts that are often incorrectly labeled are those where the name appears near the end of the caption (usually a cue that the individual named is not pictured). Some cues we could add that should improve the accuracy of our language model are the nearness of words like "shown", "pictured", or "photographed".*

The conditional language models and appearance models are then learned with supervised data (it is known for every context whether it is depicted or not and also which face has been assigned to each name) using maximum likelihood.



Actress Jennifer Lopez was nominated for a Golden Raspberry or Razzie award as "the year's worst actress" for "Enough" and "Maid in Manhattan" on February 10, 2003. Lopez is shown at the premiere of "Maid in Manhattan" on Dec. 8 and is joined by Madonna, Britney Spears, Winona Ryder and Angelina Jolie for the dubious honor. (Jeff Christensen/Reuters)

Figure 3.10: *We have created a web interface for organizing and browsing news photographs according to individual. Our data set consists of 30,281 faces depicting approximately 3,000 different individuals. Here we show a screen shot of our face dictionary* **top**, *one cluster from that face dictionary (Actress Jennifer Lopez)* **bottom left** *and one of the indexed pictures with corresponding caption* **bottom right**. *This face dictionary allows a user to search for photographs of an individual as well as giving access to the original news photographs and captions featuring that individual. It also provides a new way of organizing the news, according to the individuals present in its photos.*

| Model | EM | MM |
|---|---|---|
| Baseline PCA Appearance Model, No Lang Model | $24 \pm .06\%$ | $44 \pm .04\%$ |
| kPCA+LDA Appearance Model, No Lang Model | $56 \pm .05\%$ | $67 \pm .03\%$ |
| kPCA+LDA Appearance Model + N.B. Lang Model | $72 \pm .04\%$ | $77 \pm .04\%$ |
| kPCA+LDA Appearance Model + Max Ent Lang Model | – | $78 \pm .04\%$ |

Table 3.1: **Above:** *To form an evaluation set, we randomly selected 1000 faces from our data set and hand labeled them with their correct names. Here we show what percentage of those faces are correctly labeled by each of our methods (clustering without a language model, clustering with our Naive Bayes language model and clustering with our maximum entropy language model) as well as for a baseline PCA appearance model. Standard deviation is calculated by dividing the test set into 10 subsets containing 100 faces each and calculating the deviation over the accuracies for these subsets. Incorporating a language model improves our labeling accuracy significantly. Standard statistical knowledge says that EM should perform better than choosing the maximal assignment at each step. However, we have found that using the maximal assignment works better than EM for both the basic clustering and clustering with a language model. One reason this could be true is that EM is averaging faces into the mean that do not belong.*

## 3.6 Results

Because this is an unsupervised task, it is not meaningful to divide our data into training and test sets. Instead, to evaluate our clusterings, we create an evaluation set consisting of 1000 randomly chosen faces from our data set. We hand label these evaluation images with their correct names (labeling with 'NULL' if the face was not named in the caption or if the named entity recognizer failed to detect the name in the caption). To evaluate a clustering, we can compel our method to associate a single name with each face (we use the name given by the maximum likelihood correspondence once the parameters have been estimated), and then determine how many faces in the evaluation set are correctly labeled by that name. This is a stern test; a less demanding alternative is to predict a ranked list of names for a given face,

| Classifier | labels corr. | IN corr. | OUT corr. |
|---|---|---|---|
| Baseline | 67% | 100% | 0% |
| EM Labeling with N.B. Language Model | 76% | 95% | 56% |
| MM Labeling with N.B. Language Model | 84% | 87% | 76% |
| MM Labeling with max ent Language Model | 86% | 91% | 75% |

Table 3.2: **Above:** *To form an evaluation set for text labeling, we randomly chose 430 captions from our data set and hand labeled them with IN/OUT according to whether that name was depicted in the corresponding picture. To evaluate how well our natural language module performed on labeling depiction we look at how our test set names were labeled. "labels correct" refers to the percentage of names that were correctly labeled, "IN correct" refers to the percentage of IN names that were correctly labeled, "OUT correct" refers to the percentage of OUT names that were correctly labeled. The baseline figure gives the accuracy of labeling all names as IN. Incorporating both our Naive Bayes and Maximum Entropy language models improve labeling significantly. As with the faces, the maximum likelihood procedure performs better than EM. Names that are most often mislabeled are those that appear near the end of the caption or in contexts that most often denote people who are not pictured.*

but this is harder to evaluate.

**kPCA+LDA is a reasonable model:** We test our appearance model against a commonly used baseline face representation of principal components analysis (PCA). In table 3.1 we see that the appearance only clustering using kPCA followed by LDA performs better than the PCA appearance model. kPCA plus LDA labels 67% of the faces correctly, while PCA labels 44% of the faces correctly.

**Maximal assignment performs better than EM:** In table 3.1, we see that the basic clustering correctly labels 56% of the test images correctly when estimated with EM (as in section 3.4.2), and 67% of the test images correctly when estimated with maximal assignment (as in section 3.4.3). For context understanding clustering, 72% of the faces are labeled correctly when estimated with EM (section 3.5.3), where as 77% of the faces are labeled correctly when estimated with maximal assignment (section 3.5.4). This clearly indicates that the maximal assignment procedure performs better than EM for our labeling task. We speculate that the Gaussian model of face features conditioned on a name places too much weight on faces that are far from the mean. One other possible explanation for this phenomenon is that MM is training the model under the exact conditions for which it is tested on (to get the top correspondence correct). It would be interesting to measure the average log probability of the correct correspondence on the evaluation set, which is what EM optimizes.

**Language cues are helpful:** Language cues are helpful, because they can rule out some bad labelings. Using the same test set, we see that context understanding clustering (section 3.5) labels 77% of the test faces correctly using a naive Bayes model and 78% of the faces correctly using a maximum entropy model (table 3.1).

**Vision reinforces language:** One consequence of our context understanding clustering method is a pure natural language understanding module, which can tell whether faces are depicted in captions from context alone (i.e. one looks at $P(pictured|\mathbf{c}, \theta_c)$). We expect that, if context understanding clustering works, this module should be reasonably accurate. The module is, indeed, accurate. We hand labeled the names in 430 randomly selected captions with "IN" if the name was depicted in the corresponding picture and "OUT" if it was not. On this evaluation set (without any knowledge of the associated images), the Naive Bayes model labeled 84% of the names correctly while the Maximum Entropy model labeled 86% of the names correctly (table 3.2). Based on these two tests, we conclude that these models perform approximately equivalently on our data set. Figure 3.9 shows some example captions labeled using the learned Maximum Entropy Context model. Similarly to the face classification task, the two models perform with approximately the same accuracy, though the Maximum Entropy model again has a slight advantage over the Naive Bayes model.

**Spatial context:** One could reasonably expect that caption features like “(left)” might directly suggest a correspondence, rather than just indicate depiction. However, incorporating this information into our context understanding model was not particularly helpful. In particular, we we built a maximum entropy model of face context given name context ($P(context_{face}|context_{name})$. The feature used for face context was location in the image, and for name context the features were “(L)”, “(R)”, “left” and “right”. The maximum entropy model correctly learned that “(L)” and “left” were good indicators of the face image being on the left side of the image, while “(R)” and “right” were good indicators of the face image being on the right side of the image. However, incorporating this model into our clustering scheme had little effect on the correctness of our labelings (only increasing the accuracy by 0.3%). The reasons this might be true are: 1. Only about 10% of all the names exhibited these context cues, 2. The names with these context cues are in general already correctly assigned by our system, and 3. The signal present in linking for example “left” and the image being on the left side of the image is fairly noisy, making their connection tentative.

**Scale:** The most natural comparison with our work is that of Yang *et al.* ([Yang *et al.*, 2005b], and described briefly in Chapter 2). This work applies various multiple-instance learning methods to learn the correct association of name to face for bags consisting of a single face and 4.7 names on average. There are 234 bags where the correct name appears in the bag, and 242 where it does not; methods label between 44% and 63% of test images correctly, depending on the method. Our method shows appreciable improvements. We conjecture that there are two places in which operating with very large scale data sets is helpful. First, kPCA estimates seem to give better representations when more images are used, perhaps because high-variance directions are more stably identified. Second, more data appears to simplify correspondence problems, because the pool of relatively easily labeled images will grow. Such images might consist of faces that happen to have only one possible label, or of groups of faces where there is little doubt about the labeling (for example, two faces which are very different; as another example, one familiar face and its name together with an unfamiliar face and its name). We conjecture that the absolute size of the "easy" set is an important parameter, because a large set of easy images will make other images easy to label. For example, an image that contains two unfamiliar faces and two unfamiliar names could be much easier to label if, in another image, one of these faces appeared with a familiar face. If this conjecture is true, the problem simplifies as one operates with larger data sets.

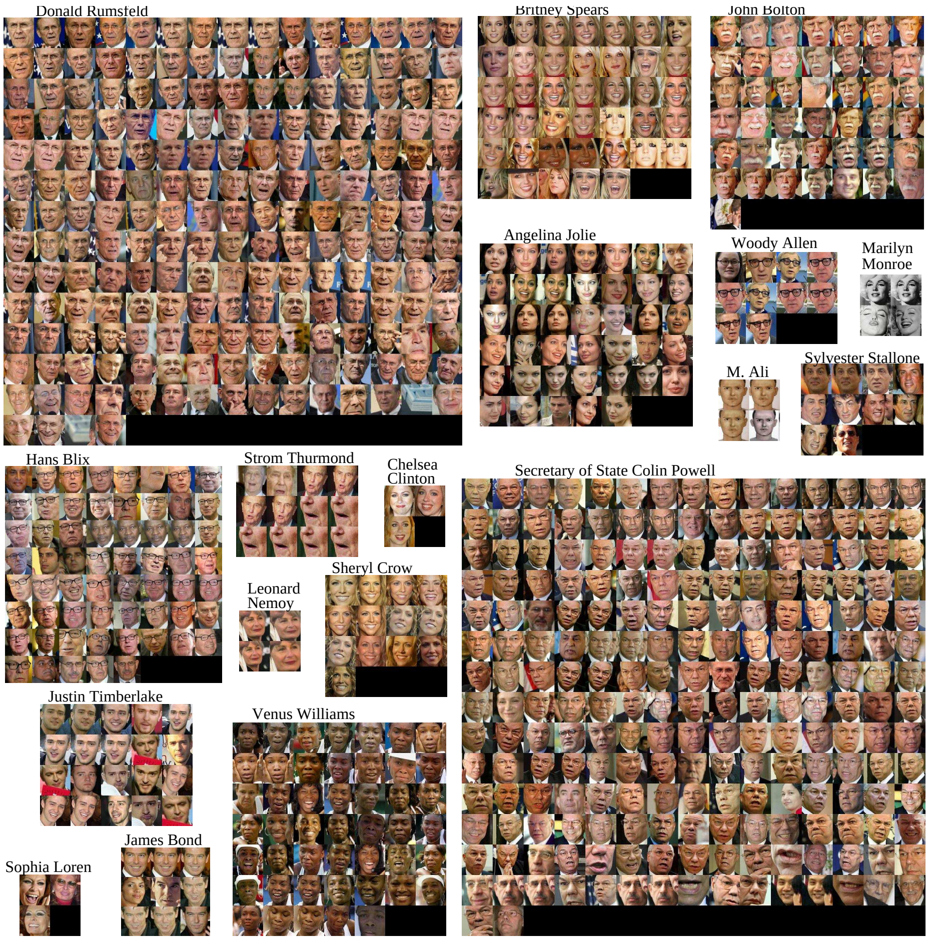


Figure 3.11: *The figure shows a representative set of clusters, illustrating a series of important properties of both the data set and the method. 1: Some faces are very frequent and appear in many different expressions and poses, with a rich range of illuminations (e.g. clusters labeled* Secretary of State Colin Powell, *or* Donald Rumsfeld*). 2: Some faces are rare, or appear in either repeated copies of one or two pictures or only slightly different pictures (e.g. cluster labeled* Chelsea Clinton *or* Sophia Loren*). 3: Some faces are not, in fact, photographs (*M. Ali*). 4: The association between proper names and face is still somewhat noisy, for example* Leonard Nemoy *which shows a name associated with the wrong face, while other clusters contain mislabeled faces (e.g.* Donald Rumsfeld *or* Angelina Jolie*). 5: Occasionally faces are incorrectly detected by the face detector (*Strom Thurmond*). 6: some names are genuinely ambiguous (*James Bond, *two different faces naturally associated with the name (the first is an actor who played James Bond, the second an actor who was a character in a James Bond film) . 7: Some faces appear in black in white (*Marilyn Monroe*) while most are in color. 8: Our clustering is quite resilient in the presence of spectacles (*Hans Blix, Woody Allen*), perhaps wigs (*John Bolton*) and mustaches (*John Bolton*).*

# Chapter 4

# Animals on the Web

We demonstrate a method for identifying images containing categories of animals. The images we classify depict animals in a wide range of aspects, configurations and appearances. In addition, the images typically portray multiple species that differ in appearance (*e.g.* uakari's, vervet monkeys, spider monkeys, rhesus monkeys, etc.). These animal categories are much more challenging and varied in appearance than the faces described in the previous chapter. Our method is accurate despite this variation and relies on four simple cues: text, color, shape and texture.

Visual cues are evaluated by a voting method that compares local image phenomena with a number of visual exemplars for the category. The visual exemplars are obtained using a clustering method applied to text on web pages. The only supervision required involves identifying which clusters of exemplars refer to which sense of a term (for example, "monkey" can refer to an animal or a bandmember).

Because our method is applied to web pages with free text, the word cue is ex-

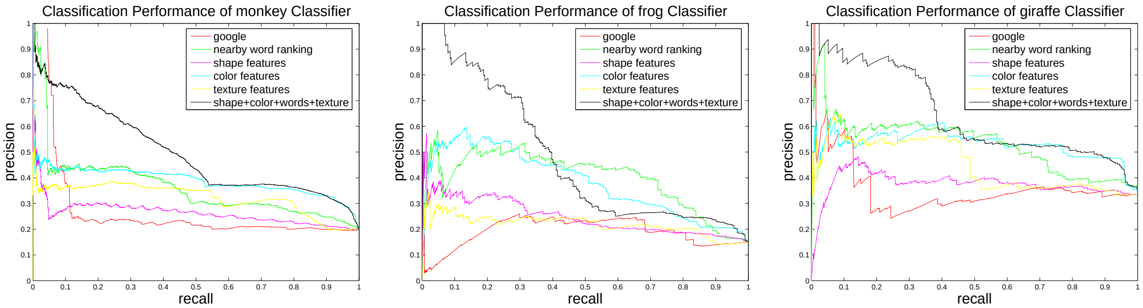


Figure 4.1: Classification performance on Test images (all images except visual exemplars) for the "monkey" (**left**), "frog" (**center**) and "giraffe" (**right**) categories. Recall is measured over images in our collection, not all images existing on the web. "monkey" results are on a set of 12567 images, 2456 of which are true "monkey" images. "frog" results are on a set of 1964 images, 290 of which are true "frog" images. "giraffe" results are on a set of 873 images, 287 of which are true "giraffe" images. Curves show the Google text search classification (**red**), word based classification (**green**), geometric blur shape feature based classification (**magenta**), color based classification (**cyan**), texture based classification (**yellow**) and the final classification using a combination of cues (**black**). Incorporating visual information increases classification performance enormously over using word based classification alone.

tremely noisy and much more difficult than the fairly stylized captions of Chapter 3. We show unequivocal evidence that visual information improves performance for our task and that a combination of simple visual features with text cues is quite powerful. Our method allows us to produce large, accurate and challenging visual data sets mostly automatically.

## 4.1 Introduction

There are currently more than 8,168,684,336[1] web pages on the Internet. A search for the term "monkey" yields 36,800,000 results using Google text search. There must be a large quantity of images portraying "monkeys" within these pages, but retrieving

[1] Google's last released number of indexed web pages

them is not an easy task as demonstrated by the fact that a Google image search for “monkey” yields only 30 actual “monkey” pictures in the first 100 results. Animals in particular are quite difficult to identify because they pose difficulties that most vision systems are ill-equipped to handle, including large variations in aspect, appearance, depiction, and articulated limbs.

We build a classifier that uses word and image information to determine whether an image depicts an animal. This classifier uses a set of examples, harvested largely automatically, but incorporating some supervision to deal with polysemy-like phenomena. Four cues are combined to determine the final classification of each image: nearby words, color, shape, and texture. The resulting classifier is very accurate despite large variation in test images. In figure 4.1 we show that visual information makes a substantial contribution to the performance of our classifier.

We demonstrate one application by harvesting pictures of animals from the web. Since there is little point in looking for, say, “alligator” in web pages that don’t have words like “alligator”, “reptile” or “swamp”, we use Google to focus the search. Using Google text search, we retrieve the top 1000 results for each category and use our classifier to re-rank the images on the returned pages. The resulting sets of animal images (fig 4.4) are quite compelling and demonstrate that we can handle a broad range of animals.

For one of our categories, “monkey”, we show that the same algorithm can be used to label a much larger collection of images. The data set that we produce from this set of images is startlingly accurate (81% precision for the first 500 images) and displays great visual variety (fig 4.7). This suggests that it should be possible to build

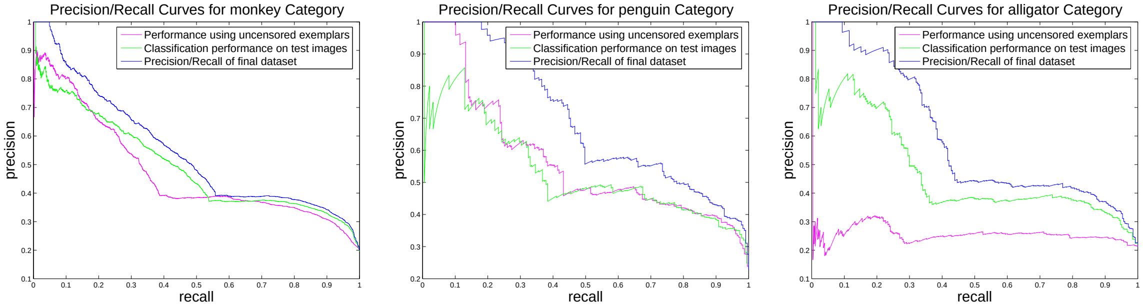


Figure 4.2: Our method uses an unusual form of (very light) supervisory input. Instead of labeling each training image, we simply identify which of a set of 10 clusters of example images are relevant. Furthermore, we have the option of removing erroneous images from clusters. For very large sets of images, this second process has little effect (compare the magenta and blue curves for "monkey" **left**), but for some smaller sets it can be helpful (*e.g.* "alligator" **right**). On a strict interpretation of a train/test split, we would report results only on images that do not appear in the clusters (**green**). However, for our application – building data sets – we also report a precision/recall curve for the accuracy of the final data set produced (**blue**). For larger data sets the curves reported for the classification performance and data set performance tend towards one another (**green** and **blue**). Recall is measured over images in our collection, not all images existing on the web. We show results for "monkey" (**left**) on a set of 12866 images containing 2569 "monkey" images, "penguin" (**center**) on a set of 985 images containing 193 "penguin" images, and "alligator" (**right**) on a set of 1311 containing 274 "alligator" images.

enormous, rich sets of labeled animal images with our classifier.

## 4.2 Dataset

We have collected a set of 9,972 web pages using Google text search on 10 animal queries: "alligator","ant", "bear", "beaver", "dolphin", "frog", "giraffe", "leopard", "monkey" and "penguin". From these pages we extract 14,051 distinct images of sufficiently large size (at least 120x120 pixels).

Additionally, we have collected 9,320 web pages using Google text search on 13 queries related to monkey: "monkey", "monkey primate","monkey species","monkey monkeys", "monkey animal", "monkey science","monkey wild","monkey simian","monkey new world","monkey old world", "monkey banana", "monkey zoo","monkey Africa". From these pages we extract 12,866 images of sufficient size, 2,569 of which are actual monkey images.

**Animals:** In addition to the aforementioned difficulties of visual variance, animals have the added challenge of having evolved to be hard to spot. The tiger's stripes, the giraffe's patches and the penguin's color all serve as camouflage, impeding segmentation from their surroundings.

**Web Pages and Images:** One important purpose of our activities is building huge reference collections of images. Images on the web are interesting, because they occur in immense numbers, and may co-occur with other forms of information. Thus, we focus on classifying images that appear on web pages using image and local text information.

Text is a natural source of information about the content of images, but the relationship between text and images on a web page is complex. In particular, there are no obvious indicators linking particular text items with image content (a problem that doesn't arise if one confines attention to captions, annotations or image names which is what has been concentrated on in previous work). All this makes text a noisy cue to image content if used alone (see the green curves in figure 4.1). However, this noisy cue can be helpful, if combined appropriately with good image descriptors and good examples. Furthermore, text helps us focus on web pages that may contain useful images.

## 4.3 Implementation

Our classifier consists of two stages, training and testing. The training stage selects a set of images to use as visual exemplars (exemplars for short) using only text based information (Secs 3.1-3.3). We then use visual and textual cues in the testing stage to extend this set of exemplars to images that are visually and semantically similar (Sec 3.4).

The training stage applies Latent Dirichlet Allocation (LDA) to the words contained in the web pages to discover a set of latent topics for each category. These latent topics give distributions over words and are used to select highly likely words for each topic. We rank images according to their nearby word likelihoods and select a set of 30 exemplars for each topic.

Words and images can be ambiguous (*e.g.* "alligator" could refer to "alligator boots" or "alligator clips" as well as the animal). Currently there is no known method for breaking this polysemy-like phenomenon automatically. Therefore, at this point we ask the user to identify which topics are relevant to the concept they are searching for. The user labels each topic as relevant or background, depending on whether the associated images and words illustrate the category well. Given this labeling we merge selected topics into a single relevant topic and unselected topics into a background topic (pooling their exemplars and likely words).

There is an optional second step to our training process, allowing the user to swap erroneously labeled exemplars between the relevant and background topics. This makes the results better, at little cost, but isn't compulsory (see figures 4.2 and 4.3). This amounts to clicking on incorrectly labeled exemplars to move them between topics. Typically the user has to click on a small number of images since text based labeling does a decent job of labeling at least the highest ranked images. For some of the 10 initial categories, the results are improved quite a bit by removing erroneous exemplars. Whereas, for the extended monkey category removal of erroneous exemplars is largely unnecessary (compare magenta and green in fig 4.2). This suggests that if we were to extend each of our categories as we did for the monkey class this step would become superfluous.

In the testing stage, we rank each image in the data set according to a voting method using the knowledge base we have collected in the training stage. Voting uses image information in the form of shape, texture and color features as well as word information based on words located near the image. By combining each of these modalities a better ranking is achieved than using any of the cues alone.

### 4.3.1 Text Representation

For each image, because nearby words are more likely to be relevant to the image than words elsewhere on the page, we restrict consideration to the 100 words surrounding the image link in its associated web page. The text is described using a bag of words model as a vector of word counts of these nearby words. To extract words from our collection of pages, we parse the HTML, compare to a dictionary to extract valid word strings and remove common English words.

LDA [Blei *et al.*, 2003] is applied to all text on the collected web pages to discover a set of 10 latent topics for each category. LDA is a generative probabilistic model where documents are modeled as an infinite mixture over a set of latent topics and

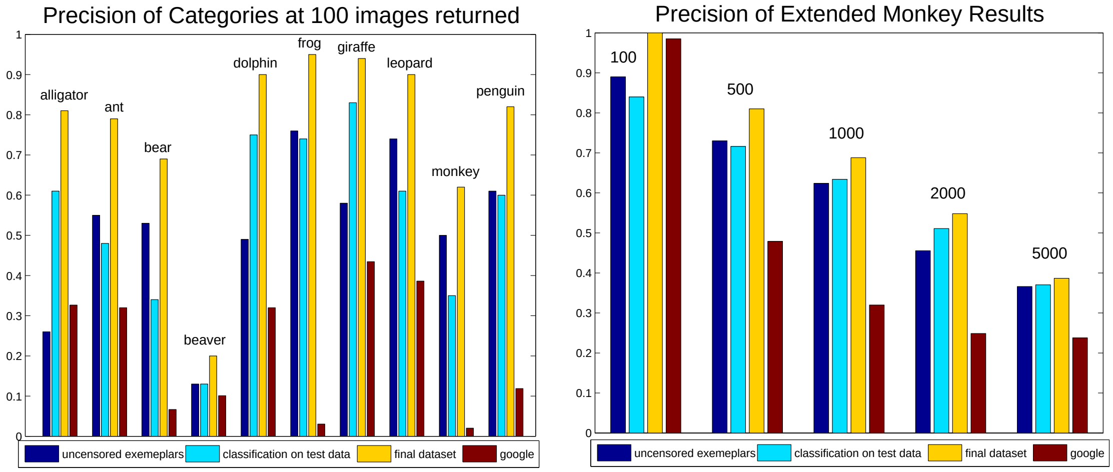


Figure 4.3: **Left:** Precision of the first 100 images for our 10 original categories: "alligator", "ant", "bear", "beaver", "dolphin", "frog", "giraffe", "leopard", "monkey", "penguin". Bar graphs show precision from the original Google text search ranking (**red**), for our classifier trained using uncensored exemplars (**blue**), and using corrected exemplars (**cyan**), described in section 4.3. One application of our system is the creation of rich animal data sets; precision of these data sets is shown in yellow. In all categories we outperform the Google text search ranking, sometimes by quite a bit ("giraffe", "penguin"). **Right:** Using multiple queries related to monkeys we are able to build an enormously rich and varied data set of monkey images. Here we show the precision of our data set (**yellow**) at various levels of recall (100, 500, 1000, 2000 and 5000 images). We also show the classification performance of the Google text search ranking (**red**) as well as two variations of our classifier, trained using uncensored (**blue**) and supervised exemplars (**cyan**) as described in section 4.3.

each topic is characterized by a distribution over words. Some of these topics will be relevant to our query while others will be irrelevant.

Using the word distributions learned by LDA, we extract a set of 50 highly likely words to represent each topic. We compute a likelihood for each image according to its associated word vector and the word likelihoods found by LDA.

### 4.3.2 Image Representation

We employ 3 types of image features, shape based geometric blur features, color features and texture features. We sample 50-400 local shape features (randomly at edge points), 9 semi-global color features and 24 global texture features per image.

The geometric blur descriptor [Berg and Malik, 2001a] first produces sparse channels from the gray scale image, in this case, half-wave rectified oriented edge filter responses at three orientations yielding six channels. Each channel is blurred by a spatially varying Gaussian with a standard deviation proportional to the distance to the feature center. The descriptors are then sub-sampled and normalized.

For our color representation we subdivide each image into 9 regions. In each of these regions we compute a normalized color histogram in RGB space with 8 divisions per color channel, 512 bins total. We also compute local color histograms with radius 30 pixels at each geometric blur feature point for use in gating of the geometric blur features as described in section 4.3.4.

Texture is represented globally across the image using histograms of filter outputs as in [Puzicha *et al.*, 1999]. We use a filter bank consisting 24 bar and spot type filters: first and second derivatives of Gaussians at 6 orientations, 8 Laplacian of Gaussian filters and 4 Gaussians. We then create histograms of each filter output.

### 4.3.3 Exemplar Initialization

Using LDA we have computed the likelihood of each image under each topic as described in section 4.3.1. We tentatively assign each image to its most likely topic. For each topic, we select the top 30 images – or fewer if less than 30 images are assigned – as exemplars. These exemplars often have high precision, a fact that is not surprising given that most successful image search techniques currently use only textual information to index images.

### 4.3.4 Shape, Color, Texture and Word Based Voting

For each image, we compute 3 types of features: shape, color and texture. For each type of feature we create two pools; one containing positive features from the relevant exemplars and the other negative features from the background exemplars. For each feature of a particular type in a query image, we apply a 1-nearest neighbor classifier with similarity measured using normalized correlation to label the feature as the relevant topic or the background topic.

For each visual cue (color, shape, and texture), we compute the sum of the similarities of features matching positive exemplars. These 3 numbers are used as the cue scores for the image. For each image, we normalize each cue score to lie between 0 and 1 by dividing by the maximum color, shape or texture cue score computed over all images. In this way the cues are used to independently rank the images (by labeling each image with a score between 0 and 1).

**Shape Feature Gating:** We modify the voting strategy for the shape feature voting. Shape features tend to match at two extremes, either the best match is a good one and has a high score or the match is a poor one with a lower score. We prune out low score matches from the voting process allowing features to vote only if their match score is quite good (normalized correlation score above 0.95). We also apply a color based gating to the geometric blur matches. If the local color of the

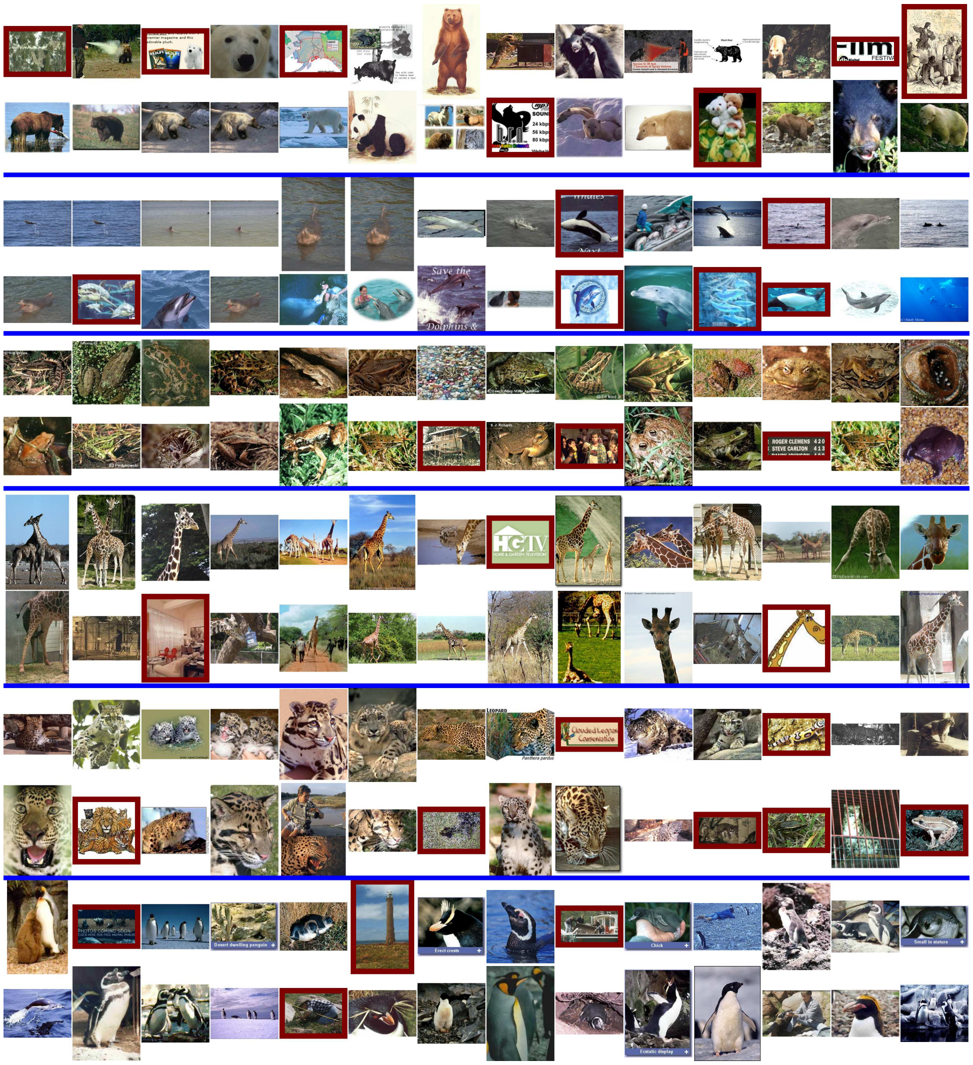


Figure 4.4: Top images returned by running our classifiers on a set of Test Images (the whole collection excluding visual exemplars) for the “bear”, “dolphin”, “frog”, “giraffe”, “leopard”, and “penguin” categories. Most of the top classified images for each category are correct and display a wide variety of poses (“giraffe”), depictions (“leopard” – heads or whole bodies) and even multiple species (“penguins”). Returned “bear” results include “grizzly bears”, “pandas” and “polar bears”. Notice that the returned false positives (dark red) are quite reasonable; teddy bears for the “bear” class, whale images for the “dolphin” class and leopard frogs and leopard geckos for the “leopard” class. Drawings, even though they may depict the wanted category are also counted as false positives

best match is significantly different from the query feature, we don't allow the feature to vote. Pruning and gating helps to disallow features from voting that are unsure about their votes and improves the shape feature voting performance significantly – especially in the higher recall range – as well as overall classification performance.

**Words:** We also compute a word score for each image by summing the likelihood under the relevant topic model found by LDA of words near the image on the associated page as described in section 4.3.1. We normalize the word score by dividing by the maximal word score over all images. This gives us a $4^{th}$ ranking of the images by labeling each image with a score between 0 and 1.

**Cue Combination:** We combine our 4 independent cue scores using a linear combination with convex weights. Currently the 4 cues are equally weighted. While equal weighting usually performs near to the optimal combination, some cues may perform better overall or for specific classes. For example, texture is a good cue for the "leopard" class while color is a good feature for the "frog" class. In the future we hope to learn this cue combination from evaluation on our training exemplars.

One **powerful** advantage of independent cue based voting is that it allows for the fact that each cue may work well for some images, but poorly for others. The errors made by each cue seem to be somewhat independent (see fig 4.1). Thus, by combining the different cues, we are able to achieve much better results than using any cue in isolation.

## 4.4 Results

We build quite effective classifiers for our initial 10 animal categories (see figure 4.3). For all categories our method (cyan) outperforms Google text search (red) in classification performance on the top 100 images returned. The giraffe and frog classifiers are especially accurate, returning 74 and 83 true positives respectively. Because exemplar based voting incorporates multiple templates per category we are able to retrieve images across different poses, aspects, and even species.

The top results returned by our classifiers are usually quite good. Figure 4.4 shows the top results returned by 6 of our classifiers: “bear”, “dolphin”, “frog”, “giraffe”, “leopard” and “penguin”. Even the false negatives returned by our classifier are often reasonable, teddy bears for the “bear” class, whale images for the “dolphin” class and leopard frogs for the “leopard” class. Drawings and toy animals, even though they may depict the correct category are counted as false positives.

**Visual Information** makes a substantial contribution to search (fig 4.1). Our classifier uses a combination of visual cues: color (cyan), shape (magenta), texture (yellow), and textual cues: nearby words (green). Their combined classification performance (black) outperforms the classification power of any single cue by a substantial margin. This shows that current text based systems could be improved by the use of visual information. We also significantly exceed the original Google ranking (red) which we compute over images based on the order of the associated page in the Google text search results.

While shape is the cue favored by most recent object recognition systems, it is often less informative than color or texture for our data set. This is due to the extreme variance in aspect and pose of animals. Color is often a good cue, especially for classes like “dolphin” and “frog”. Texture performs well for the “giraffe” and “leopard” classes. Word ranking works well for some classes (“bear”) and quite poorly for others (“penguin”). Because our cues were each used independently to rank images, we could easily incorporate a wider range of cues.

**Censored vs Uncensored Exemplars:** Our method uses an unusual form of (very light) supervisory input. Instead of labeling each training image, we simply identify which of a set of 10 clusters of example images are relevant. Furthermore, we have the option of removing erroneous images from clusters. For very large sets of images, this second process has little effect (compare the magenta and blue curves in figure 4.2 for “monkey”), but for smaller sets it can be helpful (*e.g.* “alligator”).

We believe that exemplars selected using LDA tend to be easier to classify using words because we selected them based on their high word likelihood. As a result, if we exclude them from testing, classification performance appears worse than it is. In figure 4.2, we show classifiers trained using both uncensored and censored exemplars. The uncensored case (magenta) is tested on the whole set of images, while the censored case is tested excluding the exemplars (green). The censored classifier should always perform better than the uncensored since it is provided with cleaner training data, but we see that in some cases the uncensored classifier has better accuracy. This is because by excluding the exemplar images, we bias our test set to be more difficult than the entire set of images returned by Google. This is not a phenomenon previously seen since exemplars are usually chosen at random from the set of images.

**Beaver** is the only class on which we perform poorly. Because the returned

Google results contain only 67 “beavers” in 1087 images and most returned pages don’t refer to “beavers”, LDA didn’t find a latent topic corresponding to the animal and the resulting classifier failed.

**Dataset:** We produce an extremely good data set of 10 animal categories containing pictures of animals in a wide variety of aspects, depictions, appearances, poses and species. Figure 4.2 shows precision recall curves for data sets produced for the “monkey”, “penguin” and “alligator” collections (blue), and figure 4.3 shows the precision of our data set (yellow) for the top 100 images of each category.

**Extended Monkey Category:** For the “monkey” class we collected a much larger set of images using multiple related queries. Having this much data allowed us to build an extremely powerful classifier using the same procedure as for the initial 10 categories. Figure 4.2 shows that our “monkey” classifier is startlingly accurate using both supervised (green) and uncensored exemplars (magenta).

The “monkey” data set that we produce is incredibly rich and varied. Figure 4.7 shows samples from the top 500 images in our monkey data set (top 10 lines of images), and samples from the bottom 7000 images (bottom 2 lines of images). In the data set we create 81% of the top 500 images are “monkey” pictures, and 69% of the top 1000 images are “monkeys”. Our monkey data set contains monkeys in a variety of poses, aspects, and depictions as well as a large number of monkey species and other related primates including lemurs, chimps and gibbons. Our results suggest that it should be possible to build enormous, clean sets of labeled animal images for many semantic categories.

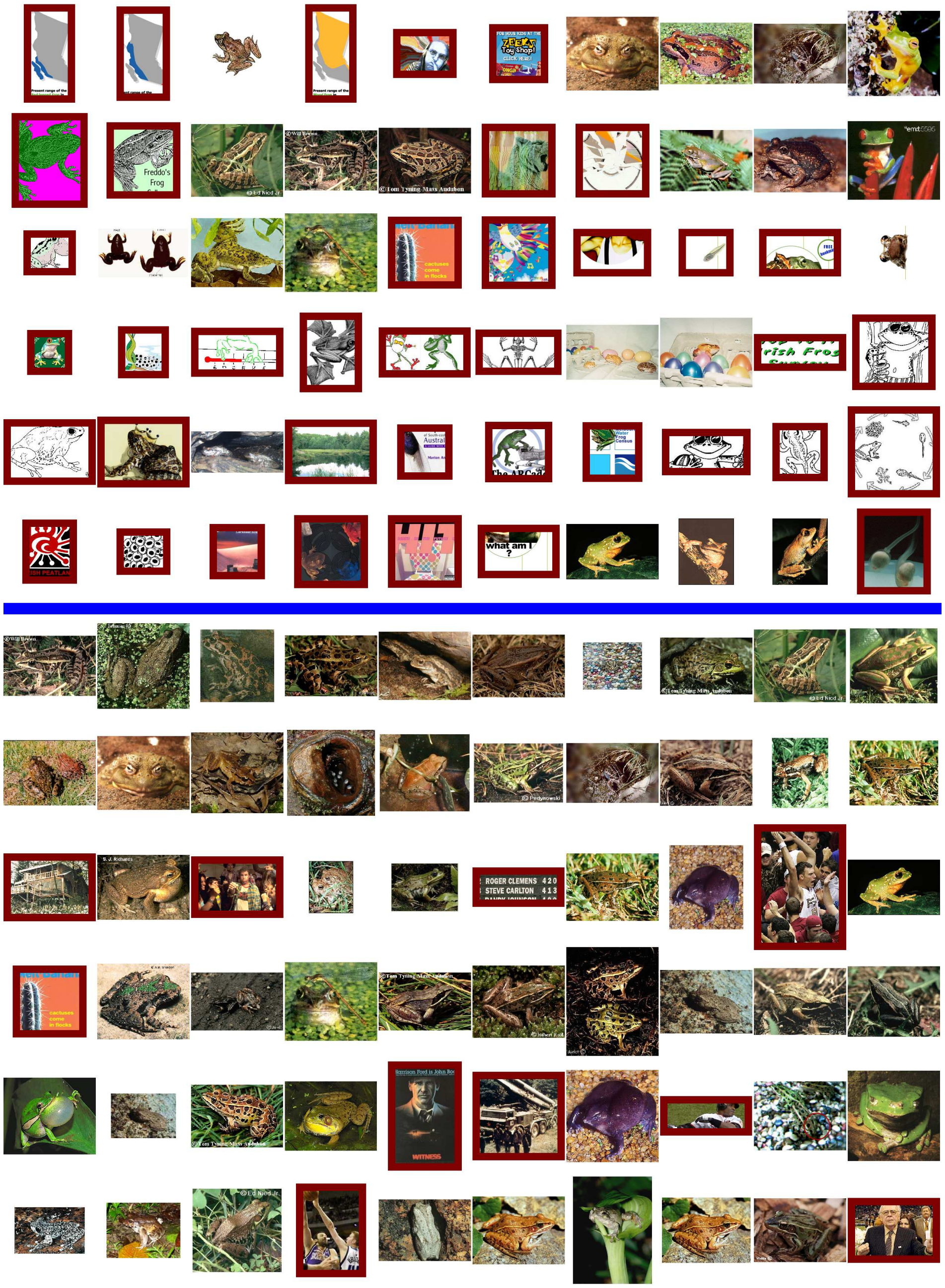


Figure 4.5: **Top:** test images classified using our text based cue for the "frog" class. **Bottom:** test images classified by four types of cues: text, color, shape and texture. False positives are outlined in red. Incorporating image based cues improves the classification accuracy significantly.

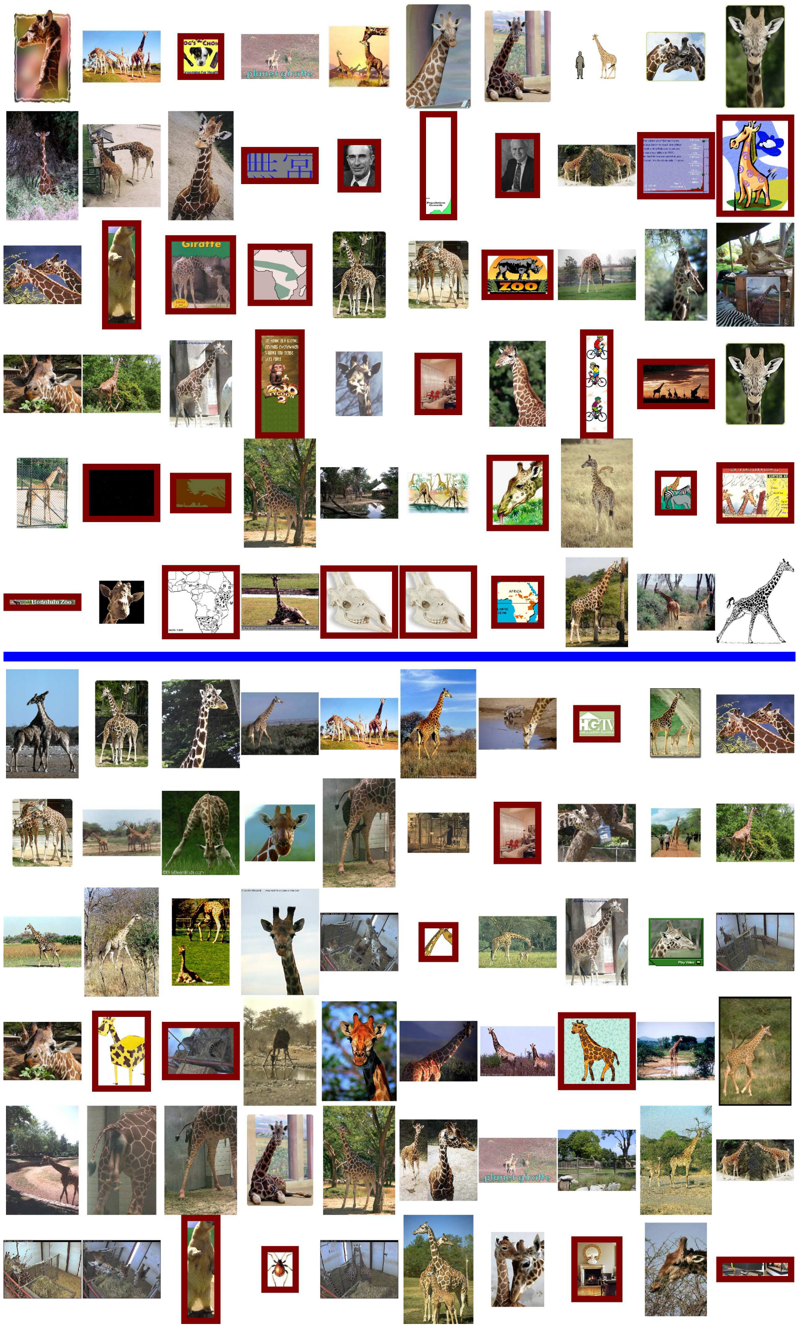


Figure 4.6: **Top:** test images classified using our text based cue for the "giraffe" class. **Bottom:** test images classified by four types of cues: text, color, shape and texture. False positives are outlined in red. The text classifier performs reasonably, but incorporating image based cues improves the classification accuracy significantly.

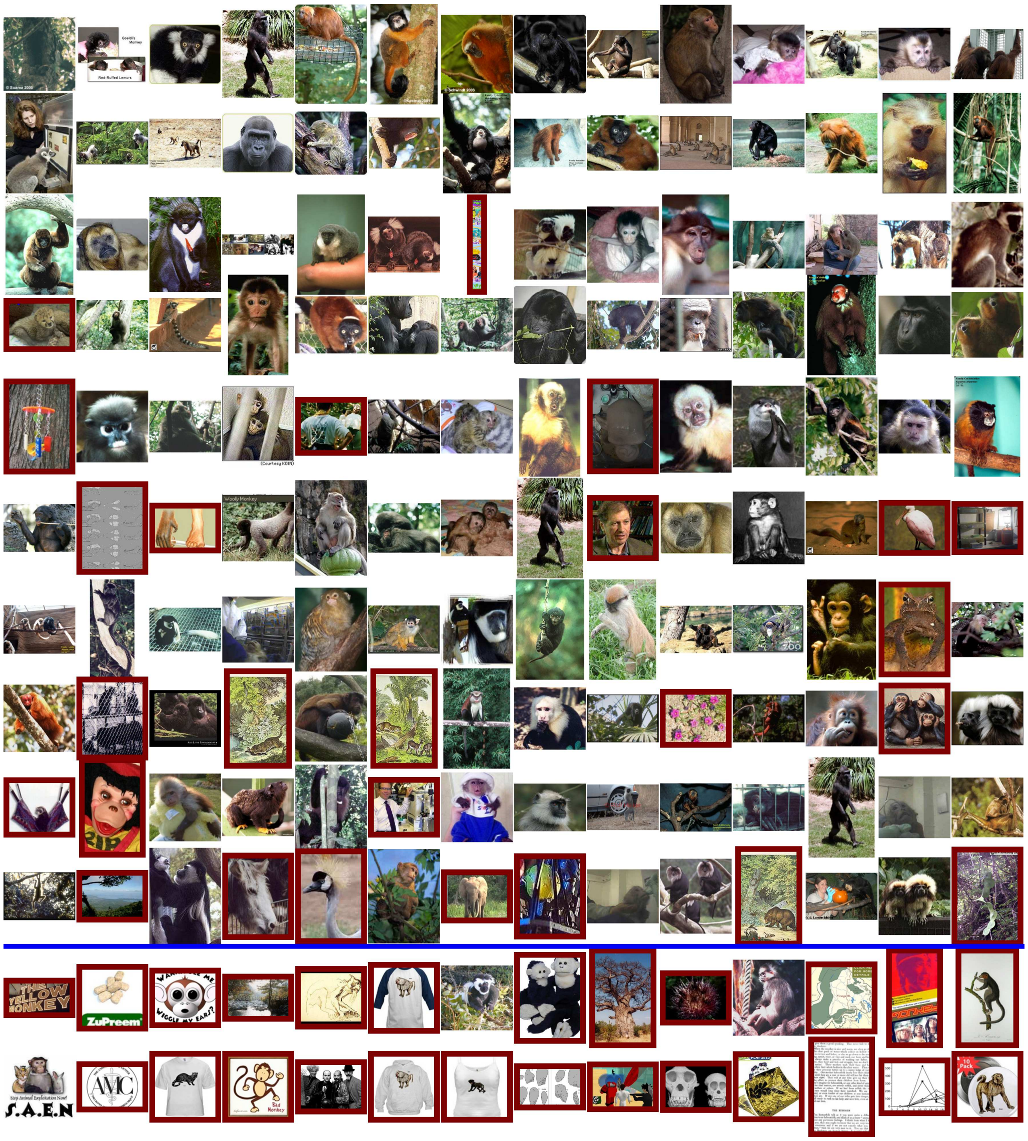


Figure 4.7: Images sampled from the data set of monkey images that we produce. The first 10 rows are sampled every $4^{th}$ image from the top 560 results, while the last two rows are sampled every $250^{th}$ image from the last 5,000-12,866 results with false positives bordered in red. Our monkey data set is quite accurate, with a precision of 81% for the top 500 images, and a precision of 69% for the top 1000 images. Deciding which images are relevant to a query doesn't have a single interpretation. We chose to include primates like apes, lemurs, chimps and gibbons, though we didn't include things such as monkey figurines (row 8, col 13), people (row 6, col 9), monkey masks (row 9, col 2) and monkey drawings (row 4, col 1), (row 8, col 4). Our results include a huge range of aspects and poses as well as a depictions in different settings (*e.g.* trees, cages and indoor settings).

# Chapter 5

# Iconic Images

We define an iconic image for an object category (*e.g.* Eiffel tower) as an image with a large clearly delineated instance of the object in a characteristic aspect. In this chapter we show that for a variety of objects such iconic images exist and argue that these are the images most relevant to that category.

Given a large set of images noisily labeled with a common theme, say a Flickr tag, we show how to rank these images according to how well they represent a visual category. In addition we also generate a binary segmentation for each image indicating roughly where the subject of the photograph is located. This segmentation procedure is learned from data on a small set of iconic images from a few training categories and then applied to several other test categories. To compare two images we compute similarity based on the shape and appearance of their respective subjects. We do so because the most important factor for similarity between two iconic images is the similarity between their depicted objects while their background regions may vary (*e.g.* the eiffel tower remains the eiffel tower on a cloudy or sunny day).

To test our method we compute three rankings of the data: a random ranking of the images within the category, a ranking using similarity over the whole image, and a ranking using similarity applied only within the subject of the photograph. We then evaluate the rankings qualitatively and with a user study.

## 5.1 Introduction

There are now many popular websites where people share pictures. Typically, these pictures are labeled, with labels indicating well-known objects depicted. However, the labelings are not particularly accurate, perhaps because people will label all pictures in a memory card with a particular label. This means, for example, that the photograph of the Eiffel Tower and a photograph of a friend taken in a nearby cafe will both have the label `eiffel tower`. Our user study results show that about half of the pictures for the categories we used on Flickr represent the category poorly.

All this means that these collections are hard to use for training object recognition programs, or, for that matter, as a source of illustrations, etc. We would like to rank such sets of images according to how well they depict the category. We refer to an image that depicts a category member well, from a good aspect and in an uncluttered way, as an **iconic image**. We believe that such iconic representations should exist for many categories, especially landmarks as we study in this chapter, because people tend to take many photographs of these objects and among this large number there will be many taken from similar characteristic views.

In this chapter, we show that iconic images can be identified rather accurately in natural data sets by segmenting images with a procedure that identifies foreground pixels, then ranking based on the appearance and shape of those foreground regions. This foreground/background segmentation also yields a good estimate of where the subject of the image lies.

### 5.1.1 Data

Our data set consists of photographs collected from Flickr for a set of 13 categories. We use all public photos uploaded over a period of one year containing that category in any associated text. Each category contains between 4,000 and 40,000 images. Four categories are used for training our segmentation algorithm: `colosseum`, `eiffel tower`, `golden gate bridge` and `stonehenge`. Nine categories are used for testing: `capital building`, `chrysler building`, `empire state building`, `lincoln memorial`, `sphinx`, `statue of liberty`, `sydney opera house`, `taj mahal` and `pyramid`.

## 5.2 Computing Segmentations

The goal of the segmentation portion of our method is to automatically detect the region of the image corresponding to the subject of the photograph. As such, we want to compute a binary segmentation of subject and background. Because this segmentation has only two labels we can use a very efficient min-cut/max-flow algorithm developed by Boykov and Kolmogorov [Boykov and Kolmogorov, 2004]. Images are modeled as a Markov Random Field where for an image, each pixel corresponds to a

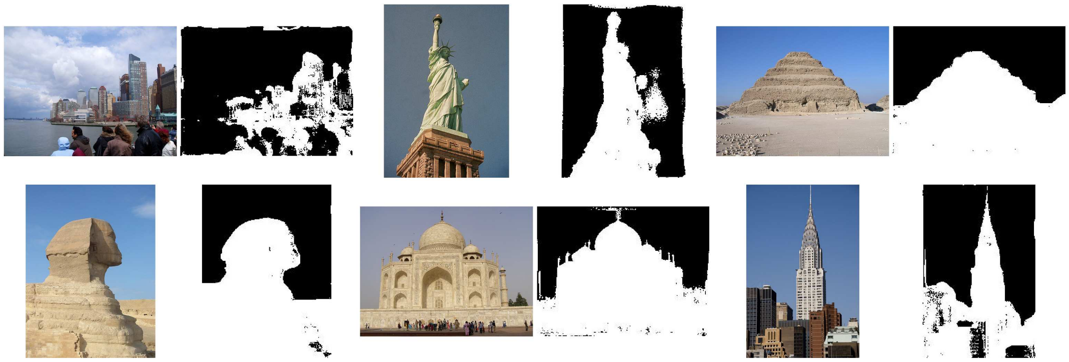


Figure 5.1: Some example segmentations of photographs into object and background labels. Our segmentation procedure is learned from a small set of 110 hand segmented iconic images from a few training categories (`eiffel tower`, `golden gate bridge`, `colosseum` and `stonehenge`). It is then applied to test images of previously unseen categories. While it is quite difficult to build a foreground/background segmentation algorithm that works on all images in general, our segmenter works well on iconic images with large, clearly delineated objects.

node of the graph, with edges between each node and the source and sink nodes, as well as edges between the pixel and its four neighboring pixels in the image.

Segmentation parameters are learned on a set of training images from 4 training categories and then applied to new images from test categories. The features used to compute our segmentations will be described in section 5.2.1 and computing the unary and binary potentials for the edge weights will be described in section 5.2.2.

### 5.2.1 Image Features

We compute 7 features describing each pixel: focus, texture, hue, saturation, value, vertical position and horizontal position. These features were selected because we tend to believe that the subject of a photograph is more likely to be sharp, textured, more vivid in color and brighter than the background. We also believe that the subject will be more likely to lie in either the middle of the photo or be placed at one of the intersections suggested by the rule of thirds (a common rule of good subject placement in photographs).

Focus is computed in a 3x3 window around each pixel as the average ratio of high pass energy to low pass energy. Texture is also computed in a 3x3 window by computing the average texture response to a set of 6 bar and spot filters. Hue, saturation and value correspond to their respective values at each pixel. Location for each pixel is represented as its $x$ location and $y$ location divided by the image width and height respectively. Each of these features has a value ranging between 0 and 1.

### 5.2.2 Learning Potentials

We use training data to learn how our features contribute to the probability of subject versus background and to the probability of a discontinuity between neighboring pixel labels. We use 110 training images from 4 categories (`colosseum`, `eiffel tower`, `golden gate bridge` and `stonehenge`) that have been hand segmented into object and background. These training images were selected to be highly iconic images with large, clearly delineated subjects.

There are two types of potentials necessary for our segmentation algorithm. The unary potentials correspond to the probability of a pixel being subject (edge weights between pixels and the source node) and the probability of a pixel being background (edge weights between pixels and the sink node). The second potential type are the binary potentials between neighboring nodes. These correspond to the probability of the labels being the same between neighboring nodes.

All feature vectors in the training images are clustered together using k-means clustering with 1000 clusters. The probability of subject and background, $P(source|pixel)$ and $P(sink|pixel)$, are computed for each cluster as the percentage of training pixels within the cluster labeled as object and background respectively. The probability of two neighboring pixels having the same label, $P(same|pixel_i, pixel_j)$ where $i$ and $j$ are neighboring pixels, is computed as the percentage of such occurrences given the pixel's cluster index and the neighboring pixel's cluster index.

### 5.2.3 Segmentation Results

For a test image, features are computed for each pixel. These features are associated with the index of the closest cluster center. Each pixel then inherits the source and sink probabilities of its cluster index. Each pair of neighboring pixels is assigned the pre-computed probability of having the same label given their cluster indices. We compute the edges for the image's graph as the logs of these probabilities (where edges have symmetric weights) and run the min-cut/max-flow algorithm on them.

We don't expect the segmentation to work perfectly for images in general as determining figure/ground segmentations is quite a difficult task. However, by definition the images that are iconic should have a large object instance in the midst of a fairly uncluttered background. Thus, these images should be relatively easy to segment. As long as our segmenter works on these images it should help to determine which of the large pool of images are the representative ones.

In figure 5.1 we show some segmentation results on 6 example images. In each of these images the segmentation algorithm is able to automatically determine the

| | With Segmentation | | | Without Segmentation | | | Random | | |
|---|---|---|---|---|---|---|---|---|---|
| category | 1s | 2s | 3s | 1s | 2s | 3s | 1s | 2s | 3s |
| pyramid | 0.7879 | 0.1919 | 0.0202 | 0.4242 | 0.3636 | 0.2121 | 0.2600 | 0.2300 | 0.5100 |
| lincoln | 0.7273 | 0.2121 | 0.0606 | 0.4200 | 0.3000 | 0.2800 | 0.3061 | 0.2959 | 0.3980 |
| chrysler | 0.6417 | 0.1917 | 0.1667 | 0.3000 | 0.3917 | 0.3083 | 0.2500 | 0.3083 | 0.4417 |
| statue | 0.6364 | 0.2818 | 0.0818 | 0.4909 | 0.2545 | 0.2545 | 0.2110 | 0.3211 | 0.4679 |
| taj | 0.5152 | 0.2525 | 0.2323 | 0.4227 | 0.2784 | 0.2990 | 0.2727 | 0.2727 | 0.4545 |
| sphinx | 0.3737 | 0.3232 | 0.3030 | 0.4286 | 0.3571 | 0.2143 | 0.1579 | 0.2316 | 0.6105 |
| sydney | 0.2828 | 0.2929 | 0.4242 | 0.2900 | 0.2600 | 0.4500 | 0.2800 | 0.3300 | 0.3900 |
| capital | 0.2653 | 0.1735 | 0.5612 | 0.1684 | 0.1474 | 0.6842 | 0.1250 | 0.1354 | 0.7396 |
| empire | 0.1700 | 0.3300 | 0.5000 | 0.2300 | 0.2600 | 0.5100 | 0.1400 | 0.2800 | 0.5800 |
| average | 0.4889 | 0.2500 | 0.2611 | 0.3528 | 0.2903 | 0.3569 | 0.2225 | 0.2672 | 0.5102 |

Table 5.1: Results of our user study. Users were asked to rate randomly sampled images the top 100 images for each type of ranking as to how well they represented each category where 1 corresponded to "Very Well", 2 "Moderately Well", 3 "Poorly", and 4 "Don't know". The above numbers correspond to the percentage of each rating by the users for our ranking with segmentation (**1st 3 columns**), ranking without segmentations (**2nd 3 columns**), ranking randomly (**3rd 3 columns**). As can be seen from the random results, almost half the images collected from Flickr are judged to be poor representations of the category. So, being able to select the good images from among these is an important task. Our ranking that incorporates segmentation information performs better than both a random ranking and the ranking without segmentation on 6 of the 9 categories and does quite well on several of the categories (`pyramid`, `lincoln memorial`, `chrysler building`, `statue of liberty` and `taj mahal`). For example, 79% of the top 100 rated `pyramid` images received ratings that they represented the category "Very Well" and 73% of the top 100 `lincoln memorial` pictures were rated "Very Well". From these numbers we can see that segmentation makes a clear, obviously useful difference for our system. Other categories such as the `sydney opera house` and the `empire state building` are more challenging because the object is often presented only in cluttered scenes where a segmentation into figure/ground is quite difficult. None of the rankings perform very well on these images.

subject of the photograph. Doing this allows us to compute similarity between images using the appearance of only those parts of the image that correspond to the object of interest which will be used in our ranking task, section 5.3.2. The segmentation also gives us an idea of the support of the object which is used to find objects with similar shapes.

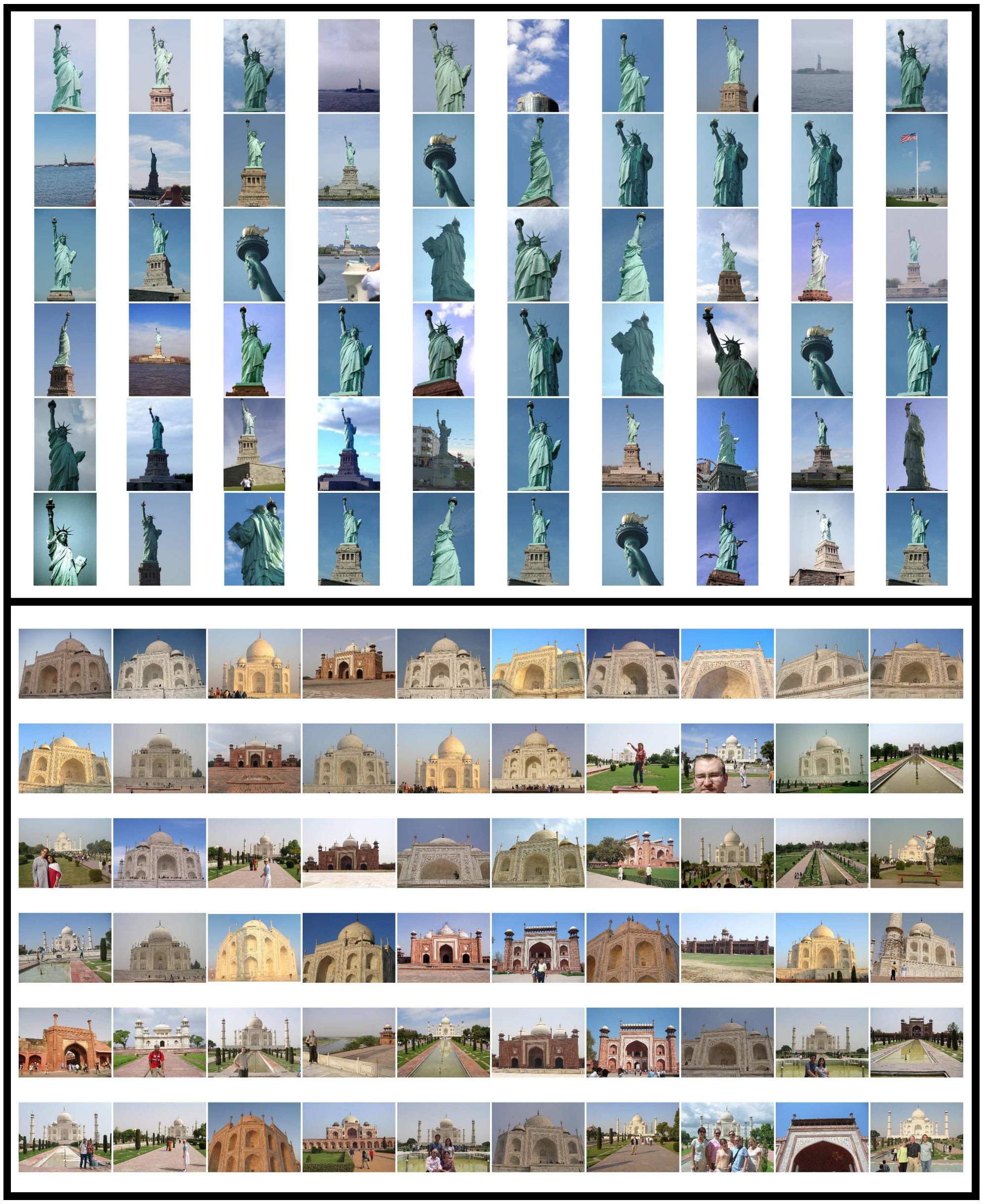


Figure 5.2: The top 60 ranked images (ranked left to right) for the `statue of liberty` and `taj mahal` categories. Several iconic representations of the statue of liberty are highly ranked including the iconic torch. Images of the taj mahal are highly ranked despite color variations. Some of the highly ranked buildings are incorrect, consisting of pictures of another (red) building on the `taj mahal` grounds because this building is similar in appearance and shape. Errors like these might be difficult for non-domain experts to spot.

## 5.3 Ranking Images

For each test category we select 5 iconic ground truth images as training. We compute rankings against the training images using three alternative methods and compare their results. As a baseline computation, the first ranking that we compute is a random ranking of the images. The second ranking uses similarity in appearance to the ground truth images for the appropriate category. The last ranking that we compute uses our figure/ground segmentations to compute similarity based on appearance and shape.

### 5.3.1 Ranking Without Segmentations

To rank the images we use the same 7 dimensional feature vectors as used for segmentation. These vectors have some idea of color, location, focus and texture. For each training and test image we compute the average over all pixels in the image of these feature vectors. The test images are then compared to all training images using the normalized correlation of their average feature vectors. The test images are ranked according to their maximum correlation value to any training image.

### 5.3.2 Ranking With Segmentations

For our ranking with segmentation information we compare test images to training images using similarity in shape and appearance. First the segmentation is run on all of the training and test images.

Shape similarity between test and training images is computed as the normalized correlation between their binary segmentation masks. This should give larger values to shapes that are more similar, though it is a somewhat rough measure of shape similarity.

Appearance vectors are calculated by taking the average feature vector within the region marked as object. Appearance similarity between two images is then computed as the normalized correlation between average feature vectors. Because the appearance is computed only over the region marked as object, this measure is more robust to changes in background than the similarity computed for the ranking without segmentation.

Test images are then ranked according to their maximum correlation to any training image where correlation to a training image is computed as the sum of their appearance and shape correlations.

## 5.4 Results

We have produced ranked results for 9 test categories. We judge our rankings qualitatively by showing some highly ranked photos for our three methods. More results of this nature can be viewed in the supplementary material associated with our paper. We also judge our results quantitatively according to the results of a user study which compares the goodness of our top ranked images to top ranked images ranked using the two alternative methods.

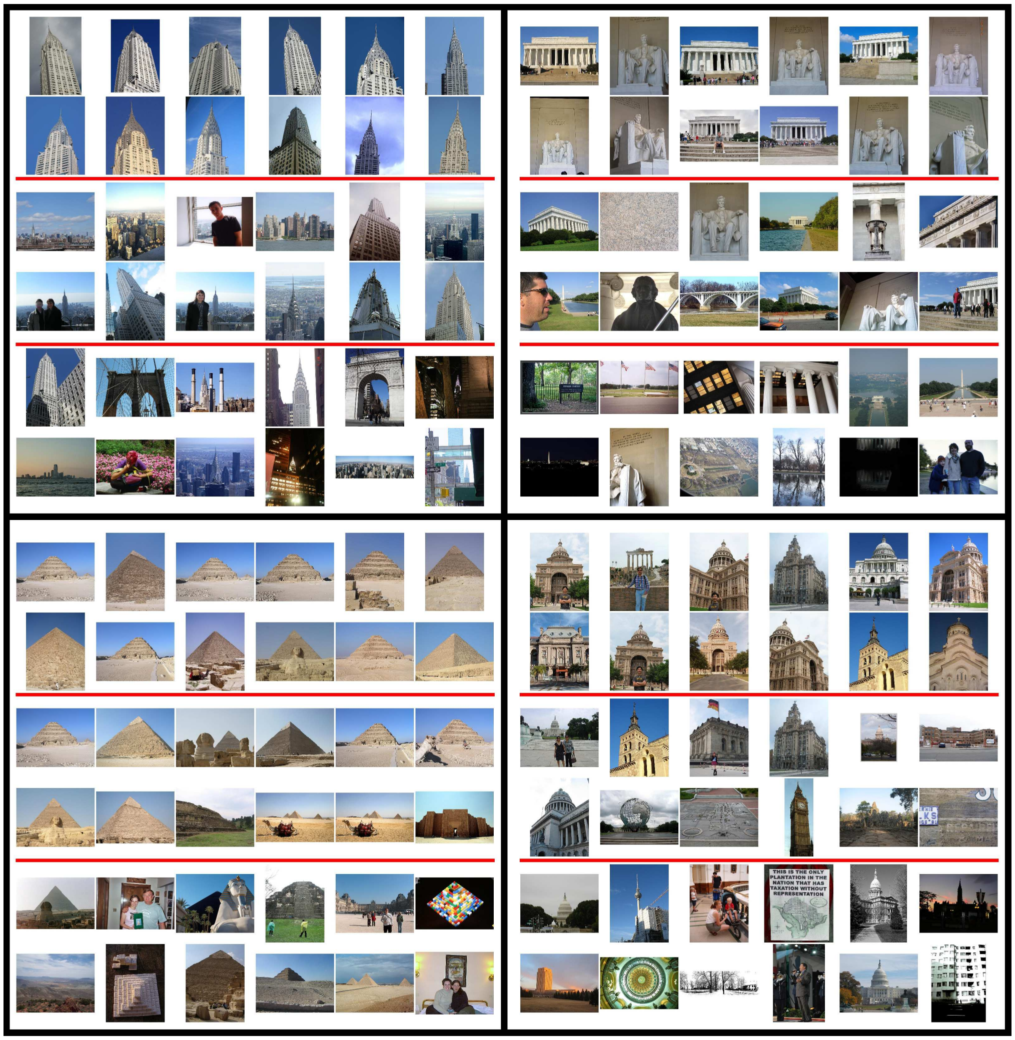

THIS IS THE ONLY PLANTATION IN THE NATION THAT HAS TAXATION WITHOUT REPRESENTATION

Figure 5.3: Each quadrant contains images from a category of objects ranked in three ways (separated by red lines): using appearance and shape of segmented objects, using appearance similarity across the whole image, randomly. The **upper left** quadrant contains images from the `chrysler building` category, the **upper right** the `lincoln memorial`, the **lower left** the `pyramid`, and the **lower right** the `capital building`. Notice that our system performs quite favorably compared to the appearance and random based rankings. For some categories (`chrysler building`, `pyramid`, `lincoln memorial`) it does quite well. Notice that for the `lincoln memorial` class we are able to rank multiple characteristic aspects (both the outdoor view of the Memorial and Lincoln's statue).

### 5.4.1 Ranked Images

In figure 5.2 we show the top 60 ranked images (ranked left to right) for the `statue of liberty` and `taj mahal` categories. These images have been ordered using our method of ranking which includes figure/ground segmentation information. Many of the top ranked images from these categories correspond to good representations of the category. Several of the highly characteristic aspects are represented in these images including the highly iconic torch. Images of the Taj Mahal are highly ranked by our system despite color variations depending on time of day. A few of the highly ranked buildings are incorrect, showing images of another (red) building on the Taj Mahal grounds. This building is highly ranked because it has a very similar appearance and shape. Errors like these might be difficult for non-domain experts to spot.

In figure 5.3 we show the top ranked images using segmentation, the top ranked images without using segmentation, and the top images for a random ranking of the images (separated by red lines). The four quadrants each show a different test category where the upper left contains images from the `chrysler building` category, the upper right the `lincoln memorial`, the lower left the `pyramid` category and the lower right the `capital building` category.

For the `chrysler building` category the difference between our ranking including segmentation (top), the rankings without segmentation (middle), and the random ranking (bottom) is startling. Our method is able to extract images containing iconic photographs of the building whereas the two other rankings show views where even if the building is present, it is present in a much less iconic context. The ranking without segmentation seems to select images that have approximately the right overall make-up (when judged based on color for example), but since it is considering the whole image equally it is not able to make the distinction between skyline images and iconic close up images containing only the Chrysler building.

Our rankings for the `lincoln memorial` and the `pyramid` category are also significantly better than those of the random ranking and the ranking without segmentation. For the `lincoln memorial` category, we are able to rank multiple characteristic aspects (both the outdoor view of the memorial building and the inside view of Lincoln's statue). Even though the method of ranking without segmentation was presented with the same training images it still produces a much less compelling ranking. This is true for the `pyramid` category as well.

`Capital building` was our most muddled category. This was partially due to the fact that during collection we had in mind images depicting the U.S. **Capitol** building in Washington D.C., but incorrectly spelled the query as **capital** building. The term capital building can be used to refer to any state (etc) capital building. Therefore, the images collected tend to depict different capitals from around the globe including the Wisconsin, and Texas capital buildings. Many of these buildings actually have similar shapes to the U.S. Capitol building and so are hard to distinguish. As can be seen in figure 5.3 the top images ranked for this category don't all depict the U.S. Capitol building, but do tend to be photographs of quite similar looking domed buildings.

### 5.4.2 User Ranking

We also judge our performance based on user ratings. Twenty-three volunteers (mostly graduate and undergraduate students) with no idea of the purpose of the experiment were asked to label a random selection of images sampled from the top 100 images from each type of ranking. For each image, the user was asked to label it according to how well it represented a category, where 1 corresponds to a rating of “Very Well”, 2 to “Moderately Well”, 3 to “Poorly”, and 4 to “Don’t Know”. Besides the written instructions we also provided a visual aid of several example images from a training category, eiffel tower, labeled as 1, 2 or 3.

We show the tallied results for each of the three rankings in table 5.1. For each ranking method and for each category, the table shows the percentage 1s, 2s, and 3s assigned to the top 100 images from that ranking.

According to the numbers for the random ranking, about 50% of the images that we collected from Flickr are judged to be poor examples of the category name. Being able to automatically select the high quality images from this noisy set is an important and nontrivial task.

If we measure performance as the percentage of the 100 top-ranked images that received a rating of 1, then we see that our ranking with incorporated segmentation information performs better than both a random ranking and the ranking without segmentation on 6 of the 9 test categories. We do quite well on several of the categories (`pyramid`, `lincoln memorial`, `chrysler building`, `statue of liberty` and `taj mahal`). For example, 79% of our 100 top-ranked `pyramid` images receive ratings indicating that they represent the category “Very Well” and 73% of our 100 top-ranked `lincoln memorial` pictures are rated “Very Well”. From these figures we can see that segmentation makes a clear, obviously useful difference for our system.

Other categories such as the sydney opera house and the empire state building are more challenging because the object is often presented only in cluttered scenes where a segmentation into figure/ground is quite difficult. None of the rankings perform very well on these images.

We use a t-test to determine whether the difference in sample means is significant for the three different ranking methods. The t-test is calculated as the ratio of the difference between the sample means to the variability of the values. We compute this for the average percentage of images ranked as representing the category "Very Well" (labeled as 1s). For our ranking including segmentation versus the ranking without segmentation, t is calculated to be 1.6429, giving about a 7% chance of observing these results if the means were actually equal. For the our ranking with segmentation versus the random ranking, t is calculated to be 3.05 or about a 3% chance of observing these results given equal means. This suggests that the difference between our ranking and the two alternative methods is a statistically significant difference.

Some comments that the users had were related to the confusion in exactly what the definition of a category is. They were presented with just the category name and so some were unsure how to rate images showing other versions of the category than the standard meaning (*e.g.* photographs of a sphinx house cat in the sphinx category). There was also much confusion about the capital building category mostly because of the capitol, capital problem mentioned previously. Most users labeled images with the U.S. capitol building in mind rather than the broader definition of capital building.

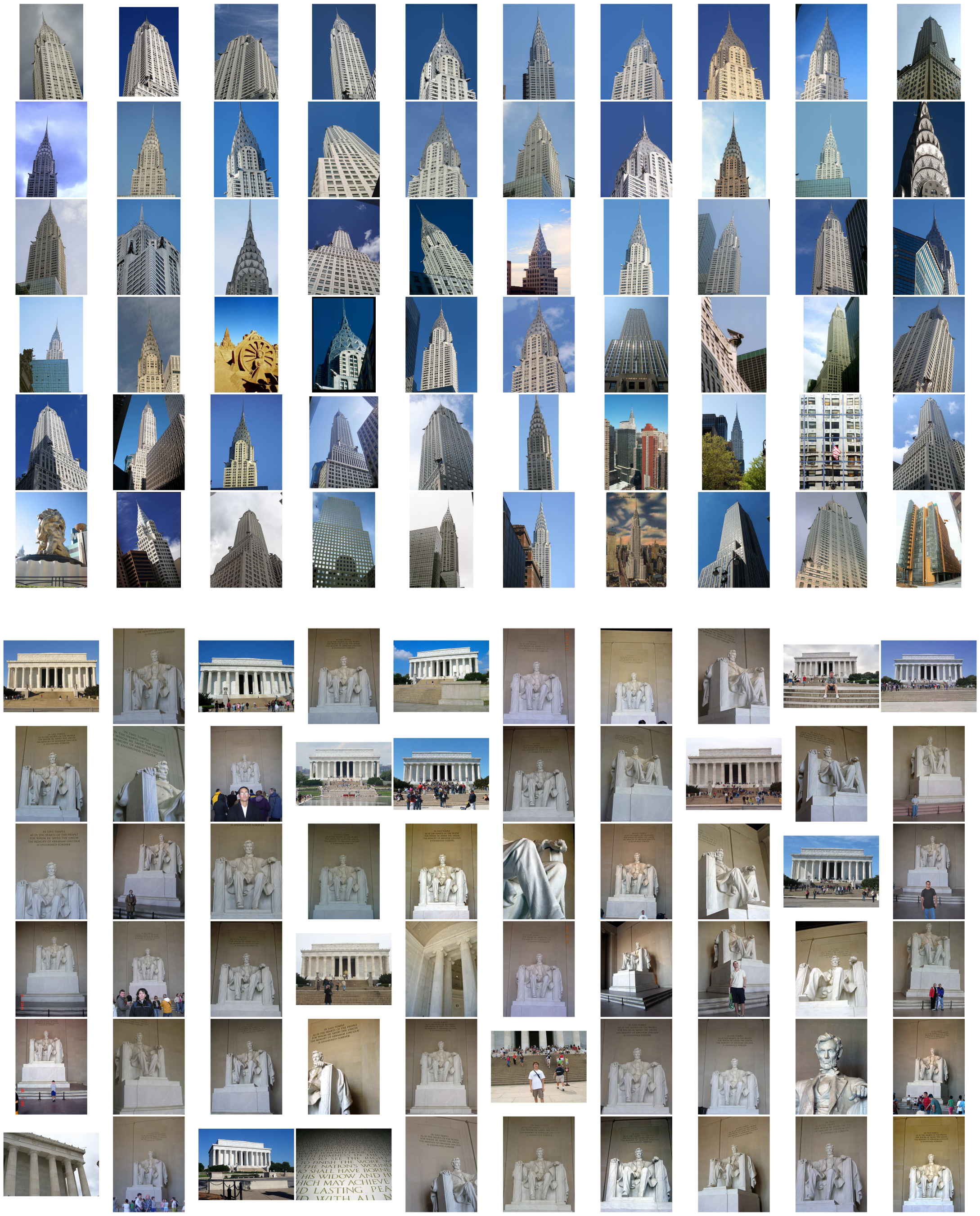


Figure 5.4: The top 60 ranked images (ranked left to right) for the `chrysler building` (top) and `lincoln memorial` (bottom) categories.

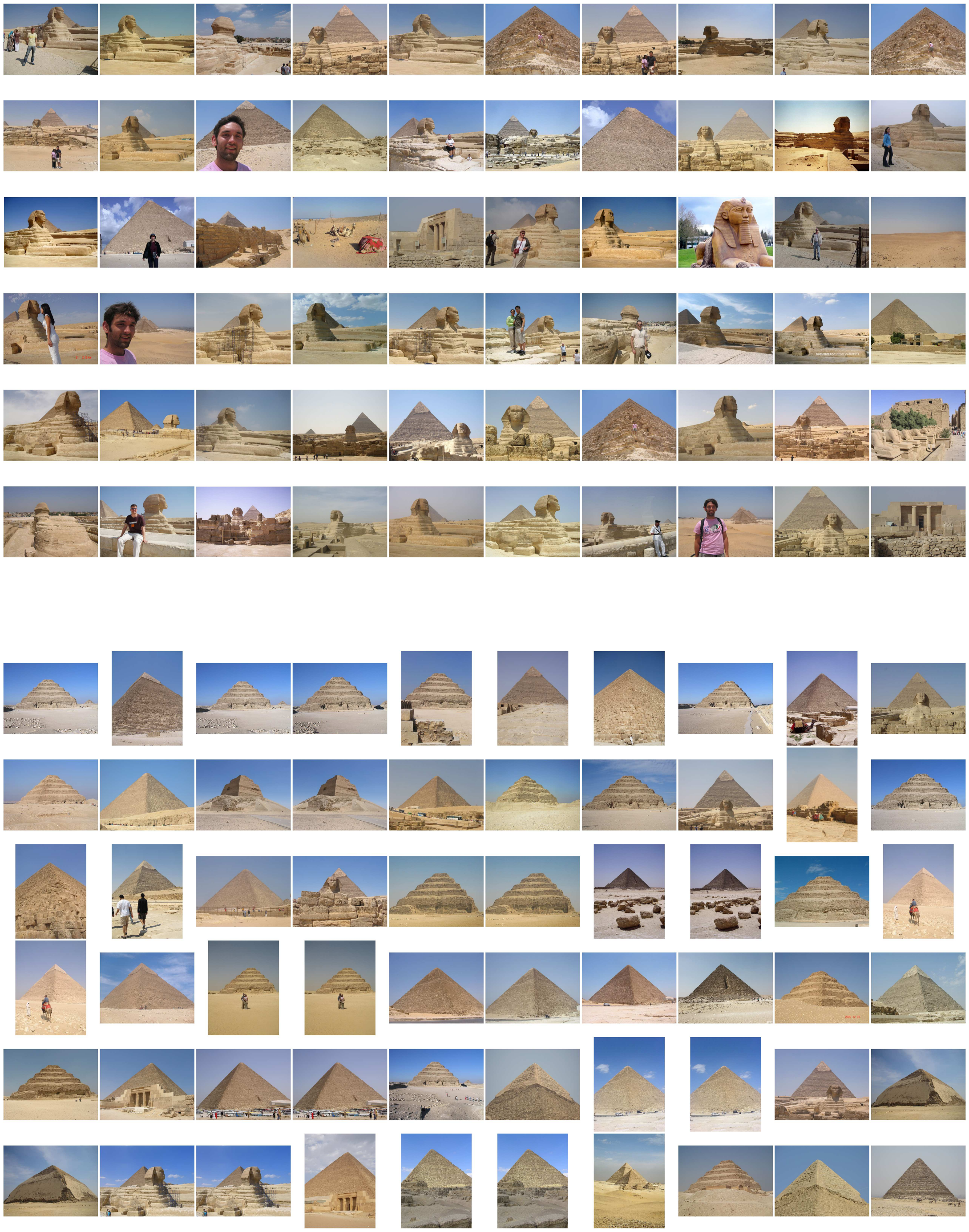


Figure 5.5: The top 60 ranked images (ranked left to right) for the `sphinx` (top) and `pyramid` (bottom) categories.

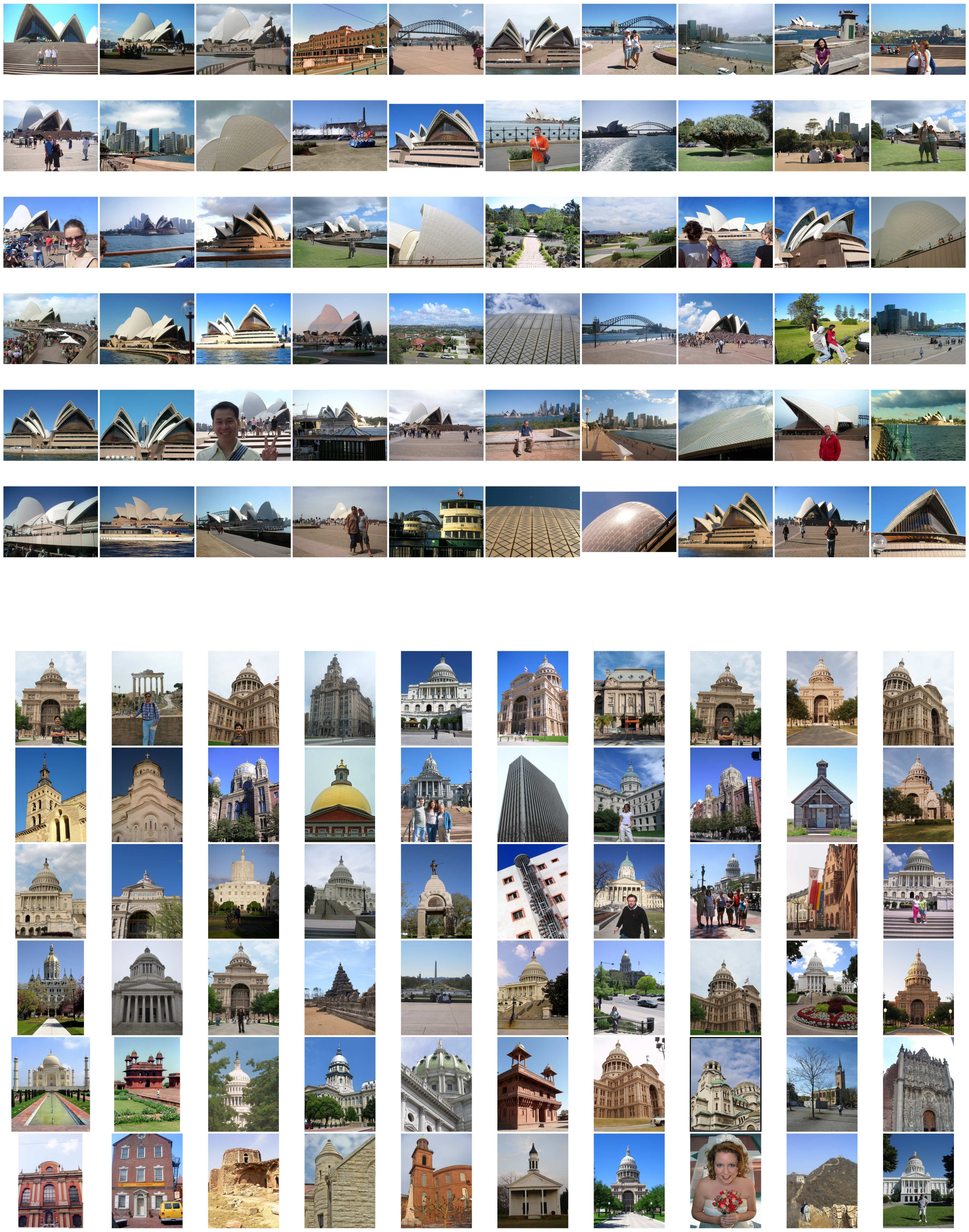


Figure 5.6: The top 60 ranked images (ranked left to right) for the sydney opera house (top) and capital building (bottom) categories.

# Chapter 6

# Conclusion

## 6.1 Names and Faces

In Chapter 3 we showed that it is possible to automatically produce a very large and realistic face data set. Our data set consists of 30,281 faces with roughly 3,000 different individuals extracted from news photographs with associated captions. This data set can be used for further exploration of face recognition algorithms. Using simple models for images and text, we were able to create a fairly good assignment of names to faces in our data set. By incorporating contextual information, this labeling was substantially improved, demonstrating that words and pictures can be used in tandem to produce results that are better than using either medium alone.

Another product of our system was a web interface that organizes the news in a novel way, according to individuals present in news photographs. Users are able to browse the news according to individual (Figure 3.5.4), bring up multiple photographs of a person and view the original news photographs and associated captions featuring that person.

We can use the language and appearance models learned by our system to label novel images or text in isolation. By learning these models in concert, we boost the amount of information available from either the images and text alone. This increases the performance power of our learned models. We have conclusively shown that by incorporating language information we can improve a vision task, namely automatic labeling of faces in images.

### 6.1.1 Data Set

**Baseline Experiments:** We have performed several baseline recognition tests to measure the difficulty of the face recognition data set produced by the system describe in Chapter 3. To do this, we select a ground truth subset of our rectified face images consisting of 3,076 faces (241 individuals with 5 or more face images per individual). These faces were hand cleaned to remove erroneously labeled faces. Half of the individuals were used for training, and half for testing. Two common baselines for face recognition data sets are PCA and PCA followed by LDA. On the test portion of this set, using the first 100 basis vectors found by PCA on the cropped face region with a 1-Nearest Neighbor Classifier gives recognition rates: of 9.4% ± 1.1% using a gallery set of one face per individual, 12.4% ± 0.6% using a gallery of two faces per individual, and 15.4% ± 1.1% using a gallery set of three faces per individual.

Using the first 50 basis vectors of LDA computed on the PCA vectors increases the accuracy to: 17% ± 2.4% for a gallery of one face per individual, 23% ± 1.9%

for a gallery of two faces per individual and 27.4% ± 2.6% for a gallery of 3 faces per individual. These numbers are quite a bit lower than the 80-90% baseline recognition rates quoted for most data sets, suggesting that our face images are in fact quite challenging and that they will be a useful data set for training future face recognition systems.

**Data set in use:** While perfect automatic labeling is not yet possible, this data set has already proved useful, because it is large, because it contains challenging phenomena, and because correcting labels for a subset is relatively straightforward. For example, Ozkan and Duygulu [Ozkan and Duygulu, 2006] used the most frequent 23 people in the database (each of whom occurred over 200 times) in additional clustering experiments. The assumption made in that work is that the clusters were more than 50 percent correctly labeled, so the current data set easily met this requirement.

The database was also used recently in work by Ferencz *et al.* on face recognition [Ferencz *et al.*, 2005; Jain *et al.*, 2006]. In this case, a subset of images with correct labels were used for training and testing in a supervised learning framework. A simple interface was used in which database examples could quickly be manually classified as correct or incorrect. As a result, it took just a couple of hours to produce a database of more than 1000 correctly labeled set of "faces in the wild" for that work.

## 6.2 Animals on the Web

In Chapter 4 we showed that it is possible to build very effective classifiers for web images depicting various animal categories. Our method outperforms the original

Google text search ranking and provides a first step toward a method for improving the accuracy of image search. We also showed that visual information makes a substantial contribution to search. Our classifier uses a combination of textual and visual cues, including features describing image color, texture and shape. The combined performance of these visual and textual features is significantly better than that of rankings computed using any one single cue.

Our method uses an unusual form of (very light) supervisory input to deal with the occurrence of multiple senses within the search results for any query. Instead of labeling each training image, we simply identify which of a set of 10 clusters of example images are relevant. Currently there is no known method to deal with this polysemy problem, but in the future we would like to be able to do this automatically.

**Dataset:** We also produced an extremely good data set of 10 animal categories containing pictures of animals in a wide variety of aspects, depictions, appearances, poses and species. For the “monkey” class we demonstrated that it was possible to use the same system on a much larger set of images by using multiple related queries. Having this much data allowed us to build an extremely powerful classifier using the same procedure as for the initial 10 categories. The “monkey” data set that we produce is incredibly rich and varied. It is also extremely accurate, with 405 images in the 500 top-ranked images portraying monkeys of various species, including lemurs, chimps and gibbons. Our results suggest that it should be possible to build enormous, clean sets of labeled animal images for many semantic categories.

## 6.3 Ranking Iconic Images

In Chapter 5 we described a method to rank images according to how well they represent a given category. This method exploited the fact that iconic representations of a category should appear with high frequency and similar appearance in a set of images linked by the fact that they have all been associated with a common label (Flickr tag). We also demonstrated that incorporating a rough idea of where the object is located in the image can improve our performance significantly.

The user comments we received reinforce the fact that notion of a category is a confusing and slippery thing. More study should be put into determining what is meant by a category.

In the future we would like to be able to rank the images in a completely unsupervised manner. We tried various methods of ranking including clustering and ways to select ground truth images according to how iconic they seemed or how similar they were to the bulk of images. None of our attempts were successful and seemed to indicate that this is a harder problem than it might seem. One last thing we would like to work on is some functional definition of iconicness according to perceptual cues of figure/ground like surroundedness and above/below.

# Bibliography

[Andrews *et al.*, 2003] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Proc. NIPS 15*, pages 561–568. MIT Press, 2003.

[Arandjelovic and Zisserman, 2005] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2005.

[Bach *et al.*, 1996] J.R. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, C.F. Shu, and Inc. Virage. Virage image search engine: an open framework for image management. In *SPIE*, 1996.

[Banerjee and Evans, 2004] S. Banerjee and B. Evans. Unsupervised automation of photographic composition rules in digital still cameras. In *Conf on Sensors, Color, Cameras and Systems for Digital Photography*, 2004.

[Barnard and Forsyth, 2001] K. Barnard and D.A. Forsyth. Learning the semantics of words and pictures. In *Int. Conf. on Computer Vision*, pages 408–15, 2001.

[Barnard and Johnson, 2005] Kobus Barnard and Matthew Johnson. Word sense disambiguation with pictures. *Artif. Intell.*, 167(1-2):13–30, 2005.

[Barnard *et al.*, 2001] K. Barnard, P. Duygulu, and D.A. Forsyth. Clustering art. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II:434–441, 2001.

[Barnard *et al.*, 2003a] Kobus Barnard, Pinar Duygulu, Nando de Freitas, David Forsyth, David Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[Barnard *et al.*, 2003b] Kobus Barnard, Pinar Duygulu, Raghavendra Guru, Prasad Gabbur, , and David Forsyth. The effects of segmentation and feature choice in a translation model of object recognition. In *CVPR*, 2003.

[Belhumeur *et al.*, 1997] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class-specific linear projection. *IEEE T. Pattern Analysis and Machine Intelligence*, 19(7):711–720, July 1997.

[Berg and Forsyth, 2006] T. L. Berg and D.A. Forsyth. Animals on the web. In *CVPR*, June 2006.

[Berg and Forsyth, 2007] T. L. Berg and D.A. Forsyth. Automatic ranking of iconic images. In *Berkeley Technical Report*, January 2007.

[Berg and Malik, 2001a] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, June 2001.

[Berg and Malik, 2001b] A.C. Berg and J. Malik. Geometric blur and template matching. In *CVPR01*, pages I:607–614, 2001.

[Berg *et al.*, 2004a] T. L. Berg, A. C. Berg, J. Edwards, and D.A. Forsyth. Who's in the picture? In *NIPS*, December 2004.

[Berg *et al.*, 2004b] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.W. Teh, E. Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *CVPR*, June 2004.

[Berg *et al.*, 2005] A. C. Berg, T. L Berg, and J. Malik. Shape matching and object recognition using low distortion correspondence. In *CVPR*, June 2005.

[Berg *et al.*, In Submission] T. L. Berg, A. C. Berg, J. Edwards, M. Maire, R. White, Y.W. Teh, E. Learned-Miller, and D.A. Forsyth. Names and faces in the news. In *International Journal of Computer Vision*, In Submission.

[Berg, 2005] A. C. Berg. Phd thesis. 2005.

[Berger *et al.*, 1996] A. Berger, S.D. Pietra, and V. D. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1), 1996.

[Blanz and Vetter, 2003] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE T. Pattern Analysis and Machine Intelligence*, 25(9), 2003.

[Blei and Jordan, 2003] David M. Blei and Michael I. Jordan. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 127–134, New York, NY, USA, 2003. ACM Press.

[Blei *et al.*, 2003] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *JMLR*, 3:993–1022, January 2003.

[Boykov and Kolmogorov, 2004] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, September 2004.

[Brown *et al.*, 1993] P. Brown, S.A. Della Pietra, V.J. Della Pietra, and R.L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 32(2):263–311, 1993.

[Carson *et al.*, 2002] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld – image segmentation using expectationmaximization and its application to image querying. *IEEE T. Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[Chen and Wang, 2004] Yixin Chen and James Z. Wang. Image categorization by learning and reasoning with regions. *J. Mach. Learn. Res.*, 5:913–939, 2004.

[Cunningham *et al.*, 2002] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: A framework and graphical development environment for robust nlp tools and applications. In *40th Anniversary Meeting of the Association for Computational Linguistics*, 2002.

[Datta *et al.*, 2005] R. Datta, J. Li, and J.Z. Wang. Content-based image retrieval - approaches and trends of the new age. In *ACM Multimedia*, 2005.

[Dietterich *et al.*, 1997] Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artif. Intell.*, 89(1-2):31–71, 1997.

[Duda *et al.*, 2001] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* 2001.

[Duygulu *et al.*, 2002] P. Duygulu, K. Barnard, N. de Freitas, and D.A. Forsyth. Object recognition as machine translation. In *ECCV*, pages IV: 97–112, 2002.

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

[Fei-Fei *et al.*, In Press] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models for 101 object categories. *CVIU*, In Press.

[Ferencz *et al.*, 2005] Andras Ferencz, Erik Learned-Miller, and Jitendra Malik. Learning hyper-features for visual identification. In *Advances in Neural Information Processing*, volume 18, 2005.

[Fergus *et al.*, 2003] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, June 2003.

[Fergus *et al.*, 2004] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for google images. In *ECCV*, May 2004.

[Fergus *et al.*, 2005] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *ICCV*, October 2005.

[Fitzgibbon and Zisserman, 2002] A.W. Fitzgibbon and A. Zisserman. On affine invariant clustering and automatic cast listing in movies. In *Proc. 7th European Conference on Computer Vision*. Springer-Verlag, 2002.

[Flickner *et al.*, 1995] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, pages 23–32, September 1995.

[Fowlkes *et al.*, 2004] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nyström method. *IEEE T. Pattern Analysis and Machine Intelligence*, 25(2), 2004.

[Frome *et al.*, 2006] A. Frome, Y. Singer, and J. Malik. Image retrieval and recognition using local distance functions. In *NIPS*, 2006.

[Gao *et al.*, 2005] B. Gao, T.Y. Liu, T. Qin, X. Zheng, Q.S. Cheng, and W.Y. Ma. Web image clustering by consistent utilization of visual features and surrounding texts. In *ACM Multimedia*, 2005.

[Geman and Geman, 1984] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *PAMI*, 6:721–741, 1984.

[Gevers and Smeulders, 2000] T. Gevers and A.W.M. Smeulders. Pictoseek: combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, pages 102–119, January 2000.

[Golub and Loan, 1996] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins Unversity Press, Baltimore, third edition, 1996.

[Govindaraju *et al.*, 1989] V. Govindaraju, D.B. Sher, R.K. Srihari, and S.N. Srihari. Locating human faces in newspaper photographs. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 549–554, 1989.

[Grauman and Darrell, 2005] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, 2005.

[Gross *et al.*, 2001] R. Gross, J. Shi, and J. Cohn. Quo vadis face recognition? In *Third Workshop on Empirical Evaluation Methods in Computer Vision*, 2001.

[Gross *et al.*, 2004] R. Gross, I. Matthews, and S. Baker. Appearance-based face recognition and light-fields. *IEEE T. Pattern Analysis and Machine Intelligence*, 26(4):449– 465, 2004.

[Harmandas *et al.*, 1997] V. Harmandas, M. Sanderson, and M.D. Dunlop. Image retrieval by hypertext links. In *ACM SIGIR*, 1997.

[Hastie *et al.*, 2001] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction.* Springer Verlag, 2001.

[Hofmann and Puzicha, 1998] Thomas Hofmann and Jan Puzicha. Statistical models for co-occurrence data. A.I. Memo 1635, Massachusetts Institute of Technology, 1998.

[Houghton, 1999] Ricky Houghton. Named faces: Putting names to faces. *IEEE Intelligent Systems*, 14(5):45–50, 1999.

[Ioffe and Forsyth, 2001] S. Ioffe and D. Forsyth. Mixtures of trees for object recognition. In *CVPR*, 2001.

[Iqbal and Aggarwal, 2002] Q Iqbal and J.K. Aggarwal. Cires: a system for content-based retrieval in digital image libraries. In *Control, Automation, Robotics and Vision*, pages 205–210, December 2002.

[Jain *et al.*, 2006] V. Jain, A. Ferencz, and E. Learned-Miller. Discriminative training of hyper-feature models for object identification. In *British Machine Vision Conference*, pages 357–366, 2006.

[Jeon *et al.*, 2003] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using crossmedia relevance models. In *SIGIR*, pages 119–126, 2003.

[Joshi *et al.*, 2004] Dhiraj Joshi, James Z. Wang, and Jia Li. The story picturing engine: finding elite images to illustrate a story using mutual reinforcement. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 119–126, New York, NY, USA, 2004. ACM Press.

[Kim *et al.*, 2002] K. I. Kim, K. Jung, and H. J. Kim. Face recognition using kernel principal component analysis. *Signal Processing Letters*, 9(2):40–42, 2002.

[La Cascia *et al.*, 1998] M La Cascia, S. Sethi, and S. Sclaroff. Combining textual and visual cues for content-based image retrieval on the world wide web. In *Workshop on Content-Based Access of Image and Video Libraries*, pages 24–28, June 1998.

[Lavrenko *et al.*, 2003] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Neural Information Processing Systems*, 2003.

[Lazebnik *et al.*, 2006] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[Leibe *et al.*, 2006] B. Leibe, K. Mikolajczyk, and B. Schiele. Segmentation based multi-cue integration for object detection. In *BMVC*, 2006.

[Lempel and Soffer, 2001] R. Lempel and A. Soffer. Picashow: pictorial authority search by hyperlinks on the web. In *WWW*, 2001.

[Li and Wang, 2003] Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10), 2003.

[Li *et al.*, 1999] J. Li, J. Wang, R. Gray, and G. Wiederhold. Multiresolution object-of-interest detection of images with low depth of field. In *CIAP*, 1999.

[Liu and Chen, 1999] J.S. Liu and R. Chen. Sequential monte-carlo methods for dynamic systems. Technical report, Stanford University, 1999. preprint.

[Lu *et al.*, 2003] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Networks*, 14(1):117– 126, 2003.

[Luo *et al.*, 2001] J. Luo, S.P. Etz, A. Singhal, and R.T. Gray. Performance-scalable computational approach to main subject detection in photographs. In *HVEI VI*, 2001.

[Ma and Manjunath, 1999] W.Y. Ma and B.S. Manjunath. Netra: a toolbox for navigating large image databases. *International Journal of Computer Vision*, pages 184–198, May 1999.

[Maron and Lozano-Pérez, 1998] Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 570–576, Cambridge, MA, USA, 1998. MIT Press.

[Maron and Ratan, 1998] O. Maron and A.L. Ratan. Multiple-instance learning for natural scene classification. In *The Fifteenth International Conference on Machine Learning*, 1998.

[McLachlan and Krishnan, 1996] G.J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley and Sons, 1996.

[Mikolajczyk, 2002] K. Mikolajczyk. Face detector. 2002. Ph.D report.

[Naaman *et al.*, 2005] Mor Naaman, Ron B. Yeh, Hector Garcia-Molina, and Andreas Paepcke. Leveraging context to resolve identity in photo albums. In *JCDL '05: Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 178–187, New York, NY, USA, 2005. ACM Press.

[Ozkan and Duygulu, 2006] D. Ozkan and P. Duygulu. A graph based approach for naming faces in news photos. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages II: 1477–1482, 2006.

[Pentland *et al.*, 1996] A. Pentland, R.W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *International Journal of Computer Vision*, pages 233–254, June 1996.

[Phillips and Newton, 2002] P. J. Phillips and E. Newton. Meta-analysis of face recognition algorithms. In *Proceeedings of the Int. Conf. on Automatic Face and Gesture Recognition*, 2002.

[Phillips *et al.*, 2002] P.J. Phillips, P. Grother, R.J Micheals, D.M. Blackburn, E Tabassi, and J.M. Bone. Frvt 2002: Evaluation report. Technical report, Face Recognition Vendor Test, 2002.

[Poggio and Sung, 1995] T. Poggio and Kah-Kay Sung. Finding human faces with a gaussian mixture distribution-based face model. In *Asian Conf. on Computer Vision*, pages 435–440, 1995.

[Puzicha *et al.*, 1999] J. Puzicha, Y. Rubner, C. Tomasi, and J. M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *ICCV*, 1999.

[Quack *et al.*, 2004] T. Quack, U. Monich, L. Thiele, and B.S. Manjunath. Cortina: a system for large-scale, content-based web image retrieval. In *International conference on multimedia*, pages 508–511, June 2004.

[Ramanan *et al.*, 2005] D. Ramanan, D.A. Forsyth, and K. Barnard. Detecting, localizing and recovering kinematics of textured animals. In *CVPR*, June 2005.

[Rasmussen, 1997] E. Rasmussen. Indexing images. *Annual Review of Information Science and Technology*, 1997.

[Ray and Craven, 2005] Soumya Ray and Mark Craven. Supervised versus multiple instance learning: an empirical comparison. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 697–704, New York, NY, USA, 2005. ACM Press.

[Rowley *et al.*, 1996a] H.A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing 8*, pages 875–881, 1996.

[Rowley *et al.*, 1996b] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 203–8, 1996.

[Rowley *et al.*, 1998a] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE T. Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[Rowley *et al.*, 1998b] H.A. Rowley, S. Baluja, and T. Kanade. Rotation invariant neural network-based face detection. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 38–44, 1998.

[Rui *et al.*, 1997] Y. Rui, T.S. Huang, and S.F. Chang. Image retrieval: Past, present, and future. In *International Symposium on Multimedia Information Processing*, 1997.

[Satoh and Kanade, 1997] Shin'ichi Satoh and Takeo Kanade. Name-it: Association of face and name in video. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, page 368, Washington, DC, USA, 1997. IEEE Computer Society.

[Satoh *et al.*, 1999] S. Satoh, Y. Nakamura, and T. Kanade. Name-it: naming and detecting faces in news videos. *IEEE Multimedia*, 6(1):22–35, 1999.

[Scheeres, 2002] J. Scheeres. Airport face scanner failed. *Wired News*, 2002. http://www.wired.com/news/privacy/0,1848,52563,00.html.

[Schmid, 2001] C. Schmid. Constructing models for content-based image retrieval. In *CVPR*, June 2001.

[Schneiderman and Kanade, 2000] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *CVPR00*, pages I: 746–751, 2000.

[Schölkopf *et al.*, 1998] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[Sirovich and Kirby, 1987] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, 4(3):519–524, 1987.

[Sivic *et al.*, 2005] J. Sivic, B. Russell, A.A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *ICCV*, October 2005.

[Smeulders *et al.*, 2000] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *PAMI*, pages 1349–1380, December 2000.

[Smith and Chang, 1997] J.R. Smith and S.F. Chang. Searching for images and videos on the world wide web. In *IEEE MultiMedia*, 1997.

[Song *et al.*, 2004] Xiaodan Song, Ching-Yung Lin, and Ming-Ting Sun. Cross-modality automatic face model training from large video databases. In *CVPRW '04: Proceedings of the 2004 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'04) Volume 5*, page 91, Washington, DC, USA, 2004. IEEE Computer Society.

[Srihari, 1995] R.K. Srihari. Automatic indexing and content based retrieval of captioned images. *Computer*, 28(9):49–56, 1995.

[Sudderth *et al.*, 2005] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Learning hierarchical models of scenes, objects, and parts. In *ICCV*, October 2005.

[Sung and Poggio, 1998] K-K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE T. Pattern Analysis and Machine Intelligence*, 20:39–51, 1998.

[Swain *et al.*, 1997] M.J. Swain, C. Frankel, and V. Athitsos. Webseer: an image search engine for the world wide web. In *CVPR*, 1997.

[Tao *et al.*, 2004] Qingping Tao, Stephen Scott, N. V. Vinodchandran, and Thomas Takeo Osugi. Svm-based generalized multiple-instance learning via approximate box counting. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 101, New York, NY, USA, 2004. ACM Press.

[Tong and Chang, 2001] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *ACM multimedia*, 2001.

[Torr and Murray, 1997] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. Computer Vision*, 24:271–300, 1997.

[Torr and Zisserman, 1998] P. Torr and A. Zisserman. Robust computation and parametrization of multiple view relations. In *Int. Conf. on Computer Vision*, pages 485–491, 1998.

[Turk and Pentland, 1991] M. Turk and A. Pentland. Eigen faces for recognition. *J. of Cognitive Neuroscience*, 1991.

[Viola and Jones, 2004] P. Viola and M.J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, May 2004.

[Wang *et al.*, 2001] J.Z. Wang, J. Li, and G. Wiederhold. Simplicity: semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence*, pages 947–963, September 2001.

[Williams and Seeger, 2001] C. Williams and M. Seeger. Using the nyström method to speed up kernel machines. In *Proc. NIPS*, volume 13, pages 682–688, 2001.

[Yanai and Barnard, 2005] Keiji Yanai and Kobus Barnard. Image region entropy: a measure of "visualness" of web images associated with one concept. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 419–422, New York, NY, USA, 2005. ACM Press.

[Yang and Hauptmann, 2004] Jun Yang and Alexander G. Hauptmann. Naming every individual in news video monologues. In *MULTIMEDIA '04: Proceedings of* *the 12th annual ACM international conference on Multimedia*, pages 580–587, New York, NY, USA, 2004. ACM Press.

[Yang *et al.*, 2000] M.-H. Yang, N. Ahuja, and D. J. Kriegman. Face recognition using kernel eigenfaces. In *IEEE Int. Conf. Image Processing*, volume 1, pages 37–40, 2000.

[Yang *et al.*, 2002] M.H. Yang, D.J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *PAMI*, 24(1):34–58, January 2002.

[Yang *et al.*, 2005a] J. Yang, A.F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin. Kpca plus lda: a complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE T. Pattern Analysis and Machine Intelligence*, 27(2):230–244, 2005.

[Yang *et al.*, 2005b] Jun Yang, Rong Yan, and Alexander G. Hauptmann. Multiple instance learning for labeling faces in broadcasting news video. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 31–40, New York, NY, USA, 2005. ACM Press.

[Yang, 2002] Ming-Hsuan Yang. Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *FGR '02: Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, page 215, Washington, DC, USA, 2002. IEEE Computer Society.

[Zhang and Goldman, 2001] Q. Zhang and S. Goldman. Em-dd: An improved multiple-instance learning technique. In *Proc NIPS*, pages 1073–1080, 2001.

[Zhang *et al.*, 1999] Z. Zhang, R.K. Srihari, and A. Rao. Face detection and its applications in intelligent and focused image retrieval. In *11'th IEEE Int. Conf. Tools with Artificial Intelligence*, pages 121–128, 1999.

[Zhang *et al.*, 2003] Lei Zhang, Longbin Chen, Mingjing Li, and Hongjiang Zhang. Automated annotation of human faces in family albums. In *MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia*, pages 355–358, New York, NY, USA, 2003. ACM Press.

[Zhang *et al.*, 2004] Lei Zhang, Yuxiao Hu, Mingjing Li, Weiying Ma, and Hongjiang Zhang. Efficient propagation for face annotation in family albums. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 716–723, New York, NY, USA, 2004. ACM Press.

[Zhang *et al.*, 2006] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR*, 2006.

[Zhao *et al.*, 2000] L. Zhao, K. Kipper, W. Schuler, C. Vogler, N. Badler, and M. Palmer. Machine translation system from english to american sign language. In *Association for Machine Translation in the Americas*, 2000.

[Zhao *et al.*, 2003] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Comput. Surv.*, 35(4):399–458, 2003.