University of Massachusetts Amherst
ScholarWorks@UMass Amherst

Open Access Dissertations

9-2010

Using Context to Enhance the Understanding of Face Images

Vidit Jain
University of Massachusetts Amherst, vidit.jain@gmail.com

Recommended Citation

Jain, Vidit, "Using Context to Enhance the Understanding of Face Images" (2010). *Open Access Dissertations*. 287.
https://scholarworks.umass.edu/open_access_dissertations/287

USING CONTEXT TO ENHANCE THE UNDERSTANDING OF FACE IMAGES

A Dissertation Presented

by

VIDIT JAIN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2010

Department of Computer Science

USING CONTEXT TO ENHANCE THE UNDERSTANDING OF FACE IMAGES

A Dissertation Presented

by

VIDIT JAIN

Approved as to style and content by:

Erik G. Learned-Miller, Chair

Allen R. Hanson, Member

James Allan, Member

Andrew K. McCallum, Member

Ramgopal R. Mettu, Member

Andrew G. Barto, Department Chair
Department of Computer Science

To my amazing parents.

ACKNOWLEDGMENTS

Just like any other journey, my trip to the completion of this dissertation had several moments when I was lost and confused. Thanks to my excellent professors, caring friends, and a loving family – I no longer recall any of these times anymore. What I do recall are the numerous joyful days from the past few years in the beautiful pioneer valley.

I would like to thank three professors: Prof. Erik Learned-Miller, Prof. Allen (Al) Hanson, and Prof. James Allan. Their invaluable advice played a very important role in the development of this dissertation. Erik played the role of a perfect advisor by being a source of inspiration and funding. I greatly appreciate his patient support to my unbaked ideas and his enthusiastic participation in our everlasting debate on generative vs. discriminative models. I would like to thank Al for being a great mentor. His insightful questions grounded in the practicalities of implementing theoretical models helped me develop simple and generalizable solutions. I would like to thank James for his guidance in a short independent study and the graduate life in general. It was a great pleasure working with him and I learned tons of things about careful experimental evaluation from him.

My graduate studies would have been a disaster if not for the fun workplace, the UMass Vision Lab – thanks to all the oldies (Frank, Dima, Pla, et al.) and the newbies (Gary, Andrew Kae, et al.). Thanks to Gary and Jackie for being super-critical while proof-reading this document and my other manuscripts. Special thanks to Marwan "Moe" Mattar for everything "*on* the top of my head" – for sharing our joy when our approaches "beat others pants down" and when we could utter "cry me a river." Outside the vision lab, I am thankful to my roommates: Rajeev, Sripati, Thahir, and Basith; and friends: Manju, Kedar, Shiraj, and Sameer; for making the A-town bearable. Thanks to the maker of the awesome cranberry cookies that fed me in the odd hours while writing this dissertation.

Finally, I would like to thank the most important people in my life, my family. Thanks to Dad for inspiring me to achieve great things in life. Thanks to Mom for being the pillar of support that I can always rely on. Thanks to the big brother for taking all the bullets for me – both on the personal and the professional front. Thanks to my sister for being super loving and caring. Thanks a lot!

ABSTRACT

USING CONTEXT TO ENHANCE THE UNDERSTANDING OF FACE IMAGES

SEPTEMBER 2010

VIDIT JAIN

B.Tech., INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Erik G. Learned-Miller

Faces are special objects of interest. Developing automated systems for detecting and recognizing faces is useful in a variety of application domains including providing aid to visually-impaired people and managing large-scale collections of images. Humans have a remarkable ability to detect and identify faces in an image, but related automated systems perform poorly in real-world scenarios, particularly on faces that are difficult to detect and recognize. *Why are humans so good?* There is general agreement in the cognitive science community that the human brain uses the context of the scene shown in an image to solve the difficult cases of detection and recognition. This dissertation focuses on emulating this approach by using different kinds of contextual information for improving the performance of various approaches for face detection and face recognition.

For the face detection problem, we describe an algorithm that employs the easy-to-detect faces in an image to find the difficult-to-detect faces in the same image. For the face recognition problem, we present a joint probabilistic model for image-caption pairs. This model solves the difficult cases of face recognition in an image by using the context generated from the caption associated with the same image. Finally, we present an effective solution for classifying the scene shown in an image, which provides useful context for both of the face detection and recognition problems.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

1.1 Motivation

Faces are special objects of interest. Not only are they useful for social interactions, but they also appear to be different from other objects in the processing of their visual stimuli by the human brain [31, 98, 115, 116]. In particular, the existence of cases of congenital prosopagnosia[1] [7] supports the hypothesis that humans are genetically wired to recognize human faces. Nevertheless, it is also evident that the ability to recognize faces is fine-tuned to the types of faces to which the person is exposed especially during infancy and youth [55]. As a result of this interplay between nature and nurture, an average adult human brain has highly developed machinery for recognizing faces.

Developing automated systems for detecting and recognizing faces is very important. In addition to providing help to people with disabilities such as impaired vision and prosopagnosia, these systems have been found to be useful in a variety of application domains such as personal access control [28], human-computer interfaces [3], and indexing and organization of large photo collections. Perhaps due to this wide-spread applicability, several face detection [88, 91, 108] and recognition [33, 64, 83] systems have been developed. Despite high performance reported by these systems on several face data sets [94, 100], automatic face recognition has not achieved acceptable performance in real-world scenarios. For instance, the face recognition system deployed

[1] Prosopagnosia refers to a disorder where the ability to recognize faces is impaired, while the ability to recognize other objects remains unaffected and no other neurological defects are observed.

at the Logan International Airport failed to match the identities of a control group in 38 percent of the cases [78].

Why are humans so good at face detection and recognition? There is substantial evidence [12, 81] that in addition to an effective part-based encoding of faces in the human brain, context plays a significant role in recognizing objects and faces (although it is unclear how and at what stage the context is processed in human vision). Bar [4] hypothesized that the human brain uses a different representation of an object for each of the different contexts in which this object could be observed. While analyzing the utility of contextual cues for object detection, Wolf and Bileschi [111] concluded that the context is a useful cue only when the appearance information is weak.

In brief, there is a general agreement that the human brain uses context to solve difficult cases of detection and recognition. The automated systems, on the other hand, typically focus only on a given image region, ignoring any kind of information external to the appearance of this image region. As a result, these systems are trying to solve face recognition and detection problems without using any scene context. In other words, they are solving a problem that may be more difficult than necessary.

In this dissertation, we focus on using different kinds of contextual information for enhancing the understanding of face images. In particular, we develop different probabilistic models to infer the context from other parts of the image, co-occurring faces, and the associated caption (if available) to solve the difficult cases of face detection and recognition.

1.2 What is context?

Context refers to the circumstances that form the setting for an observation, perception, or event. The context for an image region is defined as any information external to the appearance of the given image region. The types of context commonly found in the computer vision literature are:

- *Spatial context.* This context refers to the use of spatial constraints such as modeling the dependencies between neighboring image regions. For instance, Carbonetto et al. [18] use the spatial relationships between the objects in an image to annotate different parts of an image.

- *Geometric context.* This context refers to the reasoning about the geometric properties (e.g., the surface orientation) of the image regions. For instance, Hoiem et al. [41] used the surface orientation of different image regions as the context to boost (or reduce) the probability of observing different objects in an image region; they argued that it is more likely to see a car on a horizontal surface than a vertical surface.

- *Scene context.* This context refers to the use of statistics of the low-level image features to characterize the scene shown in a given image. For instance, Torralba et al. used the scene statistics as context for object detection [102, 103, 105] and depth estimation [104].

In this dissertation, we use two other types of context:

- *Collective context.* This context refers to performing joint inference for all of the image regions, or performing a joint inference for different, yet related, classification problems. We use this type of context to solve the face detection problem for difficult-to-detect faces in an image.

- *Multi-modal context.* This context refers to using the information from other sources (e.g., a textual description of an image), if available. We use this type of context to solve the face recognition problem for difficult-to-recognize faces in an image.

1.3 Outline of this document

The rest of this document is organized as follows. In Chapter 2, the design of a face detector that uses the easy-to-detect faces in an image to solve the difficult-to-detect faces in this image is described. Next, in Chapter 3, a joint probabilistic model is presented, which uses the caption associated with an image to generate the context about the scene shown in the image to help in the recognition of the difficult-to-recognize faces in the given image. Then, in Chapter 4, we describe a model that simultaneously performs the segmentation of an image and generates a classification label for the scene shown in the image; the scene classification label obtained from this model can potentially be used as context for either of the face detection and face recognition problems. Finally, our conclusions and discussions of some possible extensions to this work are presented in Chapter 5.

1.4 Contributions in this dissertation

There are six major contributions in this dissertation:

1. We creation of a competitive benchmark and a rigorous evaluation scheme for evaluating the performance of different face detection algorithms. This benchmark is referred to as *FDDB*, or *Face detection data set and benchmark* [48].

2. We develop an algorithm for online domain-adaptation of a pre-trained cascade of classifiers for contextual face detection [49, 52].

3. We improve the face identification performance of hyper-features based models through a direct, discriminative training of these models [47].

4. We present a joint probabilistic model that uses the coherence of face images and their captions to obtain clusters of faces and distributions of words that are closely related to a single person. We refer to this model as *People-LDA* [50].

5. We create a data set of images of five popular sports (known as the *FlickrSports-5* data set) to evaluate different approaches for scene classification.

6. We develop a probabilistic framework for simultaneously solving the segmentation and scene classification problems for a given image. This framework is referred to as *selective hidden random fields* [51].

CHAPTER 2

CONTEXTUAL FACE DETECTION

2.1 Introduction

Face detection refers to the problem of determining the location and extent of *all* the faces present in an image. A restricted formulation of this problem is *face localization*, which assumes the presence of exactly one face in the given image and the goal is to determine the location and extent of this face. In this chapter, we focus on a more general version of this problem where we do not make any assumption about a priori knowledge of the number of faces present in the given image. Furthermore, we assume that a face detection algorithm takes a *single* image as input, and this image is *not* a part of a sequence of images or a video. Note that because of the continuity in the location, size, and appearance of face regions in consecutive images in a sequence, a solution for face detection in image sequences could employ tracking-based approaches [72]. In the absence of such structured information, detecting faces in single images is likely to be more difficult than detecting faces in a sequence of images. Hereafter, we focus only on this more difficult case of face detection in single images.

Detecting faces in single images has been found useful in a variety of real-world applications. For instance, face detection systems are used for automatically controlling the exposure in digital cameras [113], and for categorizing and filtering retrieved images by commercial image search engines such as Google Images[1] and Bing.[2] De-

[1] http://images.google.com

[2] http://www.bing.com/images/

spite this widespread interest and significant research effort over the last few decades, face detection still remains an area of active research, as people try to improve automatic systems. As we show later in this chapter, the performance of standard face detection systems is very low on images acquired in unconstrained environments.

One reason for the low performance of most of these face detectors is because they perform the face vs. non-face classification independently for each candidate image region. In unconstrained environments, learning a precise model of discrimination between face and non-face regions becomes very difficult because of the presence of large variations in face appearances due to factors such as facial expressions, occlusions, and illumination patterns. We argue that the faces appearing in a single image, however, have similar range of such variations because they share the common geometric structure, illumination sources, and similar properties of the scene. For instance, in each of the two images shown in Figure 2.1 all of the face regions have similar shadow patterns. Also note that because of the difference in the pose and position of these faces relative to the camera, some faces have weaker shadows compared to the other faces. We postulate that these faces with weaker shadows (and hence stronger appearance signal) should be relatively easy to detect. In this chapter, we present an adaptive face detection algorithm that uses such easy-to-detect faces to develop a local model of discrimination between face and non-face regions that is specific to the given image.

Before we describe the design of our contextual face detector, we first present a competitive benchmark and a precise scheme for evaluating the performance of different face detection algorithms in Section 2.2. Next, in Section 2.3, we describe a scientific setup for conducting experiments to compare different face detection approaches. Then, in Section 2.4, we describe a typical approach for implementing a face detection system and, in Section 2.5, we present the details of our contextual face

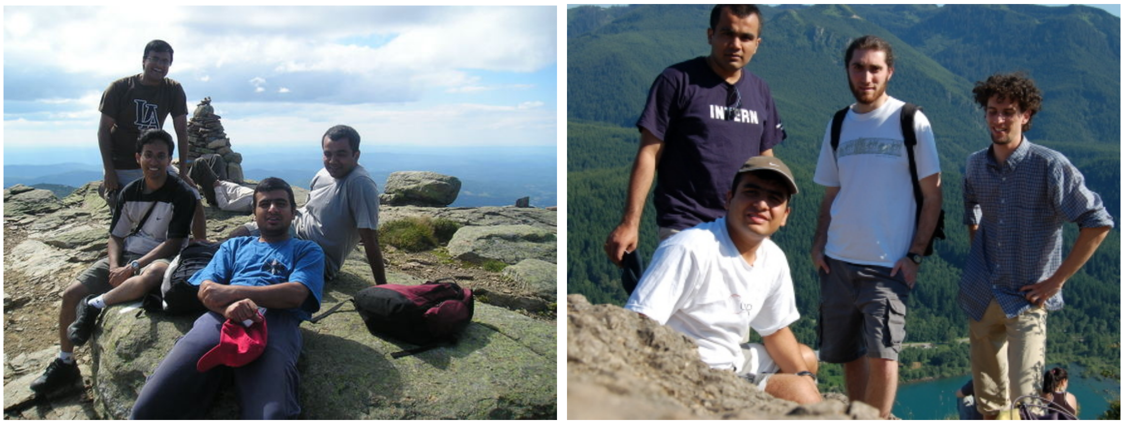


Figure 2.1. Easy-to-detect faces could help identify difficult-to-detect faces.

All of the faces in each of these images have similar shadow patterns – shadow on the right half of the faces in the left image, and shadow on the left half of the faces in the right image. Note that because of the difference in the pose and position of these faces relative to the camera, some faces have weaker shadows compared to the other faces. We postulate that these faces with weaker shadows (and hence stronger appearance signal) should be relatively easy to detect. Furthermore, these easy-to-detect faces could be used to develop a local model of discrimination between face and non-face regions that is specific to the given image.

detector. Finally, in Section 2.6, we report significant improvements in face detection results using this contextual face detector.

2.2 Evaluation of detection algorithms

Despite the maturity of face detection research, it remains difficult to compare different algorithms for face detection. This is partly due to the lack of common evaluation schemes. Also, existing data sets for evaluating face detection algorithms do not capture some aspects of face appearances that are manifested in real-world scenarios. Here, we address both of these issues. We present a new data set of face images with more faces and more accurate annotations for face regions than in previous data sets. We also propose two rigorous and precise methods for evaluating the performance of face detection algorithms.

2.2.1 Data sets

For a data set to be useful for evaluating face detection, the locations of all faces in these images need to be annotated. Sung et al. [100] built one such data set. Although this data set included images from a wide range of sources including scanned newspapers, all of the faces appearing in these images were upright and frontal. Later, Rowley et al. [88] created a similar data set with images that included faces with in-plane rotation. Schneiderman et al. [90, 91] combined these two data sets with an additional collection of profile face images, which is commonly known as the MIT+CMU data set. Figure 2.2 shows some samples from this data set.

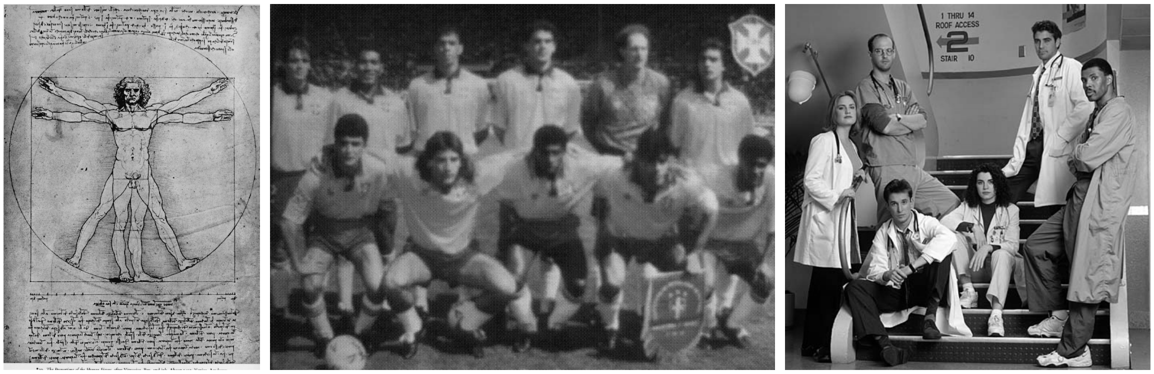


Figure 2.2. Example images from the *MIT+CMU* face data set.

This data set includes images from a wide range of sources including scanned newspapers, and includes frontal and profile images. However, since this collection contains only gray-scale images, it is not applicable for evaluating face detection systems that employ color information as well.

Since this resulting collection contains only gray-scale images, it is not applicable for evaluating face detection systems that employ color information as well [42]. Some of the subsequent face detection data sets included color images, but they also had several shortcomings. For instance, the GENKI data set [107] includes color images that show a range of head poses (yaw, pitch ±45°, roll ±20°), but every image in this collection contains exactly one face. Similarly, the Kodak [66], UCD [94] and VT-AAST [1] data sets included images of faces with occlusions, but the small sizes of these data sets limit their utility in creating effective benchmarks for face detection algorithms.

To address the above-mentioned issues, we present a new data set *FDDB* that includes

- 2845 images with a total of 5171 faces;
- a wide range of difficulties including occlusions, difficult poses, and low resolution and out-of-focus faces;
- the specification of face regions as elliptical regions; and
- both gray-scale and color images.

Next we discuss the origin and creation of this data set.

2.2.2 FDDB: Face detection data set and benchmark

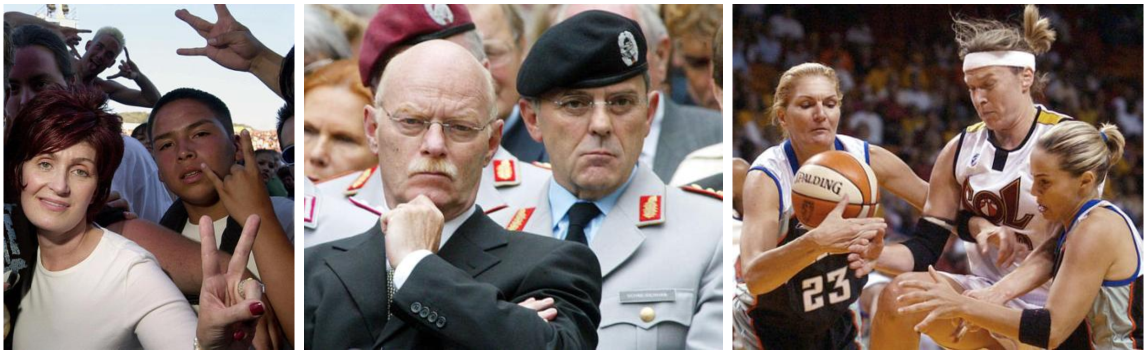


Figure 2.3. Example images from Berg et al.'s data set.

The images in this collection were collected from news articles and display large variation in pose, lighting, background, and appearance.

Berg et al. [10] created a data set that contains images and associated captions extracted from news articles (see Figure 2.3). The images in this collection display large variation in pose, lighting, background and appearance. Some of these variations in face appearance are due to factors such as motion, occlusions, and facial expressions, which are characteristic of the unconstrained setting for image acquisition. Berg et al. used Mikolajczyk's face detector [75] to extract many of the face regions in these images. This set of extracted face images is commonly known as the *Faces in the Wild* data set.[3] Since the faces in this data set were selected based on the output of an automatic face detector, an evaluation of face detection algorithms on the existing set of annotated faces would favor the approaches with outputs highly correlated with Mikolajczyk's face detector. This bias makes this data set unsuitable for evaluating different approaches for face detection. However, the richness of the images included in this collection motivated us to build an index of *all* of the faces present in a subset of images from this collection. We believe that benchmarking face detection algorithms on this data set will provide good estimates of their expected performance in unconstrained settings.

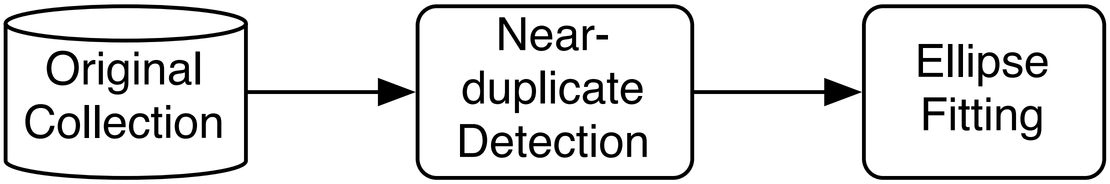


Figure 2.4. Outline of the face labeling process. To create our data set of annotated face regions, first the near-duplicate images are identified and removed, and then the remaining images are presented to manual annotators to draw ellipses around the face regions.

The images in Berg et al.'s data set were collected from the Yahoo! news website,[4] which accumulates news articles from different sources. Although different news organizations may cover a news event independently of each other, they often share

[3] http://www.tamaraberg.com/faceDataset/index.html

[4] http://news.yahoo.com

photographs from common sources such as the Associated Press[5] and Reuters.[6] The published photographs, however, may not be digitally identical to each other because they are often modified (e.g., cropped or contrast-corrected) before publication. We refer to all of the images that are derived from a single original photograph as *near-duplicates* of each other. Note that the above-mentioned processing of the original photograph to obtain an image to be published in a news article has led to the presence of several near-duplicate images in Berg et al.'s data set. Also note that the presence of such near-duplicate images is limited to a few data collection domains such as news photos and those on the Internet, and is *not* a characteristic of most practical face detection application scenarios. For example, it is uncommon to find such near-duplicate images in a personal photo collection. Thus, an evaluation of face detection algorithms on a data set with multiple copies of near-duplicate images may not generalize well across domains. For this reason, we decided to identify and remove as many near-duplicates from our collection as possible. We now present the details of the duplicate detection.

2.2.2.1 Near-duplicate detection

We selected a total of 3527 images (based on the chronological ordering) from the image-caption pairs of Berg et al. [10]. Examining pairs for possible duplicates in this collection in the naïve fashion would require approximately 12.5 million annotations. An alternative arrangement would be to display a set of images and manually identify groups of images in this set, where images in a single group are near-duplicates of each other. Due to the large number of images in our collection, it is unclear how to display all the images simultaneously to enable this manual identification of near-duplicates in this fashion.

[5] http://www.ap.org/

[6] http://www.reuters.com/

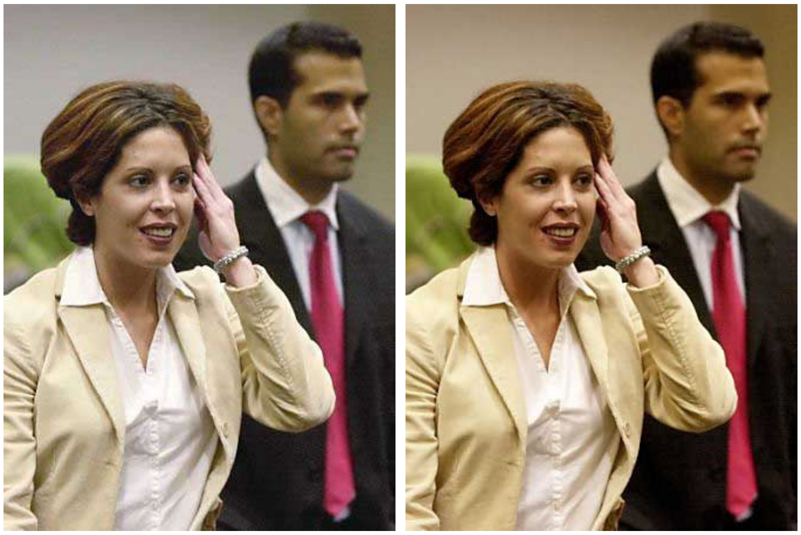


Figure 2.5. An example of a pair of near-duplicate images.

These two images differ from each other slightly in the resolution and the color and intensity distributions, but the pose and expression of the faces are identical, suggesting that they were derived from a single photograph.

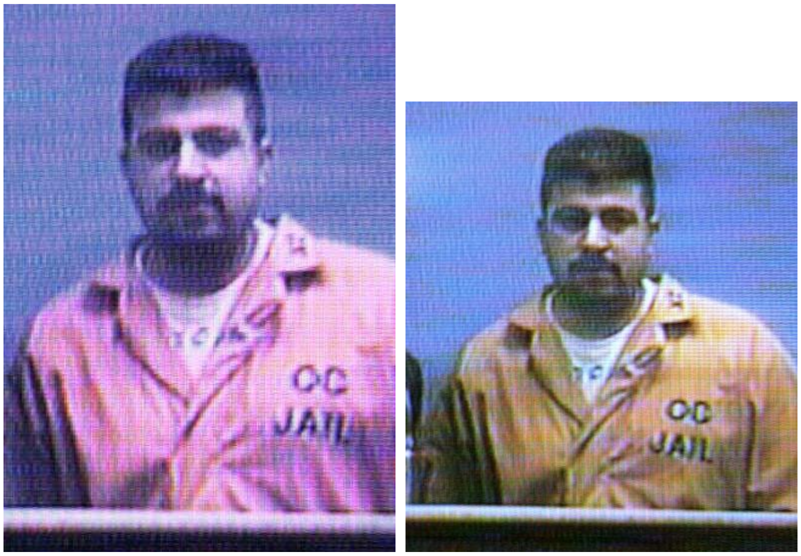


Figure 2.6. An example of a pair of similar but *not* near-duplicate images.

Although these two images are very similar in appearance, there is a slight difference in the head pose. This difference in pose suggests that these images were derived from two separate photographs, and hence are *not* considered near-identical images.

Identification of near-duplicate images has been studied for web search [20, 21, 30]. However, in the web search domain, scalability issues are often more important than the detection of all near-duplicate images in the collection. Since we are interested in discovering *all* of the near-duplicates in our data set, these approaches are not directly applicable to our task. Zhang et al. [117] presented a more computationally intensive approach based on stochastic attribute relational graph (ARG) matching. Their approach was shown to perform well on a related problem of detecting near-identical frames in news video databases. These ARGs represent the compositional parts and part-relations of image scenes over several interest points detected in an image. To compute a matching score between the ARGs constructed for two different images, a generative model for the graph transformation process is employed. This approach has been observed to achieve high recall of near-duplicates, which makes it appropriate for detecting similar images in our data set.

As with most automatic approaches for duplicate detection, this approach has a trade-off among false positives and false negatives. To restrict the number of false positives, while maintaining a high true positive rate, we follow an iterative approach (outlined in Algorithm 1) that alternates between clustering and manual inspection of the clusters. We cluster (steps 3-5 of Algorithm 1) using a spectral graph-clustering approach [80]. Then, we manually label each non-singleton cluster from the preceding step as either *uniform*, meaning that it contains images that are all near-duplicates of each other, or *non-uniform*, meaning that at least one pair of images in the cluster are not near-duplicates of each other. Finally, we replace each uniform cluster with one of the images belonging to it.

For the clustering step, in particular, we construct a fully-connected undirected graph G over all the images in the collection, where the ARG-matching scores are used as weights for the edges between each pair of images. Following the spectral graph-clustering approach [80], we compute the (unnormalized) Laplacian L_G of graph G

with n nodes as

$$L_G = D_G - W_G, \tag{2.1}$$

where D_G is a $n \times n$ diagonal matrix with elements $D_G(i,i)$ equal to the degree of the i^{th} node in G, respectively, and W_G is the adjacency matrix of G. A projection of the graph G into a subspace spanned by the top few eigenvectors of L_G provides an effective distance metric between all pairs of nodes (images, in our case). We perform mean-shift clustering [22] with a narrow kernel in this projected space to obtain clusters of images.

Algorithm 1 Identifying near-duplicate images in a collection.

```
Construct a graph $G = \{V, E\}$, where $V$ is the set of images, and $E$ are all pairwise edges with weights as the ARG matching scores.
repeat
    Compute the Laplacian of $G$, $L_G$.
    Use the top $m$ eigenvectors of $L_G$ to project each image onto $\mathbb{R}^m$.
    Cluster the projected data points using mean-shift clustering with a small-width kernel.
    Manually label each cluster as either uniform or non-uniform.
    Collapse the uniform clusters onto their centroids, and update $G$.
until none of the clusters can be collapsed.
```

Using this procedure, we were able to arrange the images according to their mutual similarities. Annotators were asked to identify clusters in which all images were derived from the same source. Each of these clusters was replaced by a single exemplar from the cluster. In this process we manually discovered 103 uniform clusters over seven iterations, with 682 images that were near-duplicates. Additional manual inspections were performed to find an additional three cases of duplication. Note that a quantitative evaluation of the above procedure for near-duplicates is not done because a labeled data set for evaluating the problem of near-duplicate detection does not exist. Although we tried our best to remove all the near-duplicates in our data set, it is conceivable that some of them still remain.

Next we describe our annotation of face regions.

2.2.2.2 Annotating face regions

As a preliminary annotation, we drew bounding boxes around all the faces in 2845 images. From this set of annotations, all of the face regions with height or width less than 20 pixels were excluded, resulting in a total of 5171 face annotations in our collection.

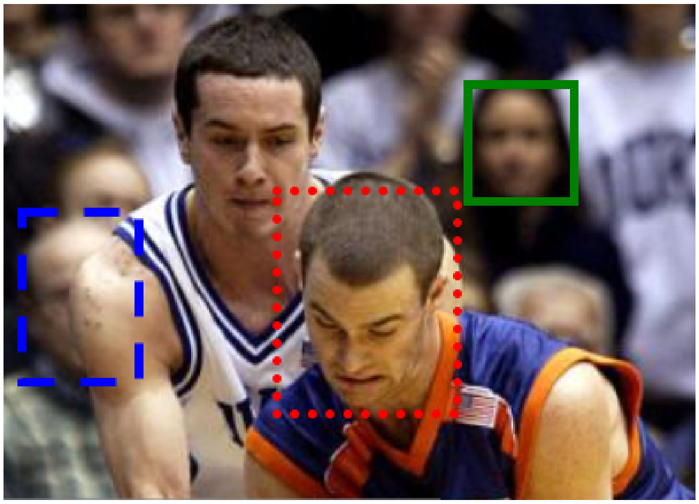


Figure 2.7. Challenges in face labeling.

For some image regions, deciding whether or not it represents a "face" can be challenging. Several factors such as low resolution (green, solid), occlusion (blue, dashed), and pose of the head (red, dotted) may make this determination ambiguous.

For several image regions, the decision of labeling them as face regions or non-face regions remains ambiguous due to factors such as low resolution, occlusion, and head-pose (e.g., see Figure 2.7). One possible approach for handling these ambiguities would be to compute a quantitative measure of the "quality" of the face regions, and reject the image regions with the value below a pre-determined threshold. We were not able, however, to construct a satisfactory set of objective criteria for making this determination. For example, it is difficult to characterize the spatial resolution needed to characterize an image patch as a face. Similarly, for occluded face regions, while a threshold based on the fraction of the face pixels visible could be used as a criterion, it can be argued that some parts of the face (e.g., eyes) are more informative than other parts. Also, note that for the current set of images, all of the regions with faces looking away from the camera have been labeled as non-face regions. In other words, the faces with the angle between the nose (specified as radially outward perpendicular to the head) and the ray from the camera to the person's head is less than 90 degrees are *not* considered face regions. This angle is denoted as θ in the illustration shown in Figure 2.8. Note that estimating this angle precisely from an image is difficult.

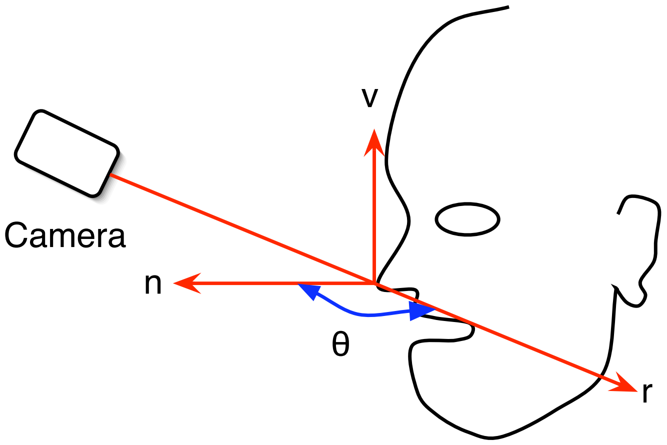


Figure 2.8. Illustration of the pose of a head relative to the orientation of the camera.

Here, the vectors specifying the orientation of the nose and the camera are denoted by $\mathbf{n}$ and $\mathbf{r}$, respectively. Both of these vectors are assumed to be perpendicular to the vertical orientation of the head, which is denoted by $\mathbf{v}$. The pose of a head relative to the orientation of the camera is represented by the angle θ between $\mathbf{n}$ and $\mathbf{r}$.

Due to the lack of an objective criterion for including (or excluding) a face region in our data set, we resort to human judgments for this decision. Since a single human decision for determining the label for some image regions is likely to be inconsistent, we used an approach based on the agreement statistics among multiple human annotators. All of these face regions were presented to different people through a web interface to obtain multiple independent decisions about the validity of these image regions as face regions. The annotators were instructed to reject the face regions for which neither of the two eyes (or glasses) were visible in the image. They were also requested to reject a face region if they were unable to (qualitatively) estimate its position, size, or orientation. The guidelines provided to the annotators are described in Appendix A.

2.2.2.3 Elliptical face regions

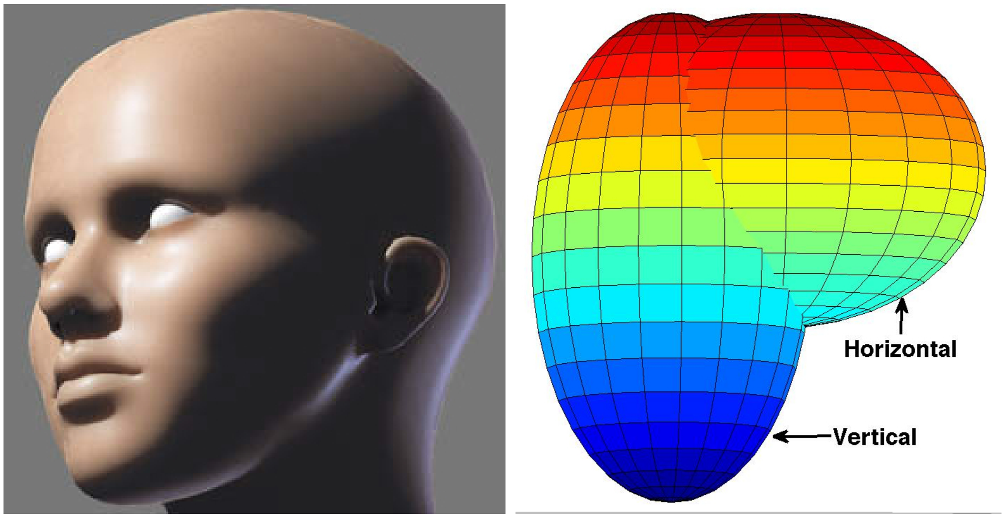


Figure 2.9. An approximation of the shape of a human head.

We approximate the shape of a human head (**left**) as the union of two ellipsoids (**right**). We refer to these ellipses as vertical and horizontal ellipsoids.

As shown in Figure 2.9,[7] the shape of a human head can be approximated using two three-dimensional ellipsoids. We call these ellipsoids the *vertical* and *horizontal* ellipsoids. Since the horizontal ellipsoid provides little information about the features of the face region, we estimate a 2D ellipse for the orthographic projection of the hypothesized vertical ellipsoid in the image plane. We believe that representing a

[7] Reproduced with permission from Dimitar Nikolov, Lead Animator, Haemimont Games.

face region by this 2D ellipse provides a more accurate specification than a bounding box without introducing any additional parameters.

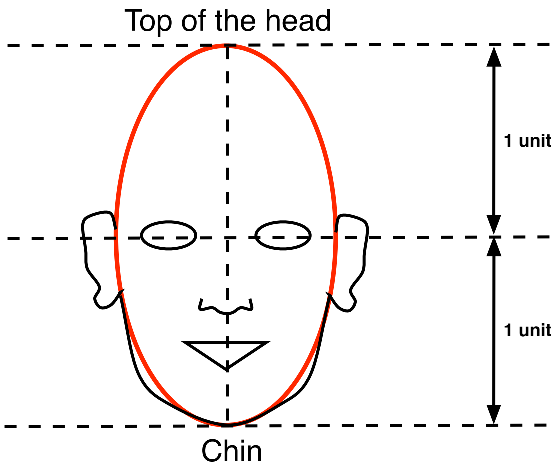


Figure 2.10. Guidelines for drawing ellipses around face regions.

The extreme points of the major axis of the ellipse are respectively matched to the chin and the topmost point of the hypothetical vertical ellipsoid used for approximating the human head (see Figure 2.9). Note that this ellipse does not include the ears. Also, for a non-frontal face, at least one of the lateral extremes (left or right) of this ellipse are matched to the boundary between the face region and the corresponding (left or right) ear. The details of our specifications are included in Appendix A.

We specified each face region using an ellipse parameterized by the location of its center, the lengths of its major and minor axes, and its orientation. Since a 2D orthographic projection of the human face is often not elliptical, fitting an ellipse around the face regions in an image is challenging. To make consistent annotations for all the faces in our data set, the human annotators are instructed to follow the guidelines shown in Figure 2.10. In Figure 2.11, we illustrate the annotations obtained by following these guidelines to the faces in an actual image. Some difficult example annotations in our data set are shown in Figure 2.12.

The next step is to produce a consistent and reasonable evaluation criterion for comparing different detection algorithms.

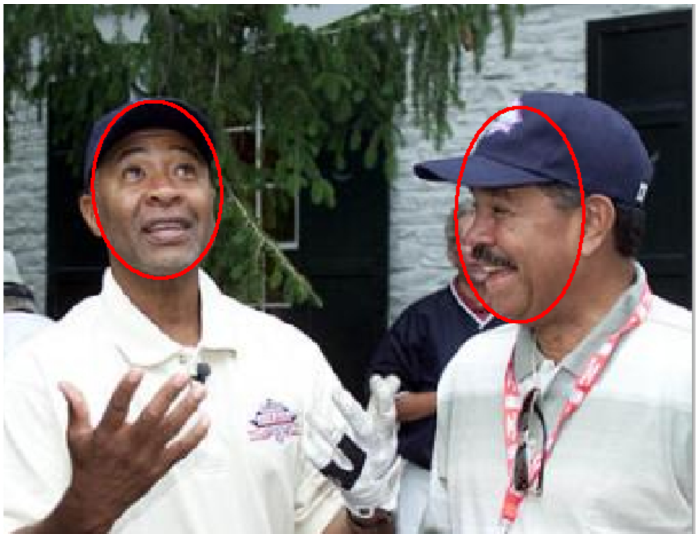


Figure 2.11. An illustration of annotations for all the faces in an image.

The two red ellipses specify the location of the two faces present in this image. Note that for a non-frontal face (**right**), the ellipse traces the boundary between the face and the visible ear. As a result, the elliptical region includes pixels that are not a part of the face.

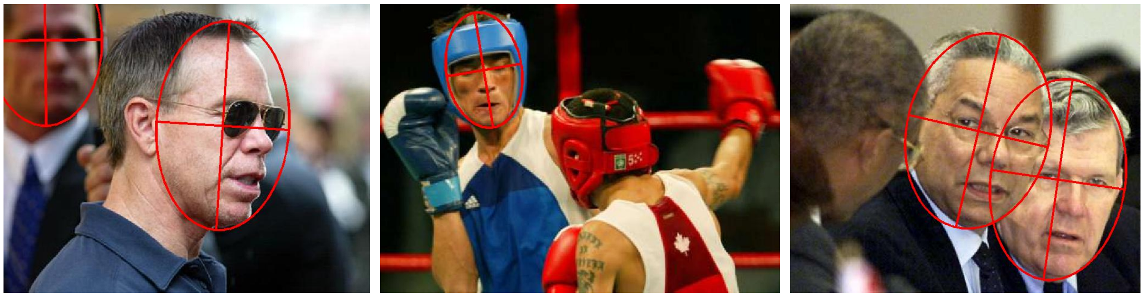


Figure 2.12. Some difficult examples of face annotations.

2.2.3 Evaluation scheme

To establish an evaluation criterion for detection algorithms, we first specify some assumptions we make about their outputs. We assume that

- A detection corresponds to a contiguous image region.
- Any post-processing required to merge overlapping or similar detections has already been done.

- Each detection corresponds to exactly one entire face, no more, no less. In other words, a detection cannot be considered to detect two faces at once, and two detections cannot be used together to detect a single face. We further argue that if an algorithm detects multiple disjoint parts of a face as separate detections, only one of them should contribute towards a positive detection and the remaining detections should be considered as false positives.

To represent the degree of match between a detection d_i and an annotated region l_j, we employ the commonly used ratio of intersected areas to joined areas:

$$S(d_i, l_j) = \frac{area(d_i) \cap area(l_j)}{area(d_i) \cup area(l_j)}. \tag{2.2}$$

2.2.3.1 Matching detections and annotations

A major remaining question is how to establish a correspondence between a set of detections and a set of annotations. While this problem is easy for very good results on a given image, it can be subtle and tricky for large numbers of false positives or multiple overlapping detections (see Figure 2.13 for an example). Below, we formulate this problem of matching annotations and detections as finding a maximum weighted matching in a bipartite graph (as shown in Figure 2.14).

Let L be the set of annotated face regions (or labels) and D be the set of detections. We construct a graph G with the set of nodes $V = L \cup D$. Each node d_i is connected to each label $l_j \in L$ with an edge weight w_{ij} as the score computed in Equation 2.2. For each detection $d_i \in D$, we further introduce a node n_i to correspond to the case when this detection d_i has no matching face region in L.

A matching of detections to face regions in this graph corresponds to the selection of a set of edges $M \subseteq E$. In the desired matching of nodes, we want every detection to be matched to at most one labeled face region, and every labeled face region to be matched to at most one detection. Note that the nodes n_k have a degree equal to one,

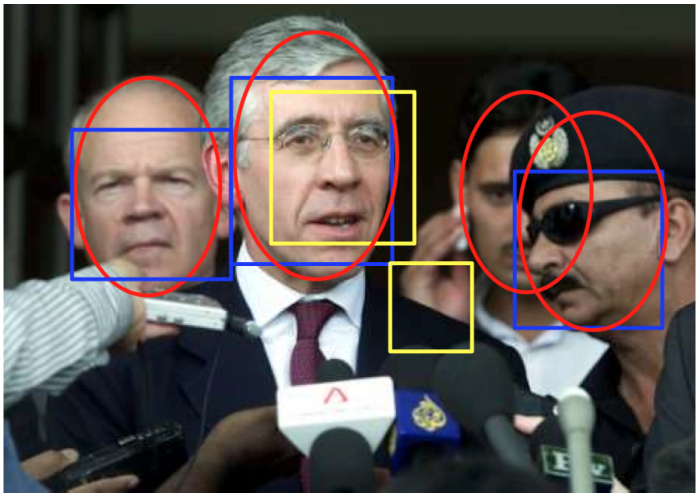


Figure 2.13. Example scenario for matching detections and annotations. In this image, the ellipses specify the face annotations and the five rectangles denote a face detector's output. Note that the second face from left has two detections overlapping with it. We require a valid matching to accept only one of these detections as the true match, and to consider the other detection as a false positive. Also, note that the third face from the left has no detection overlapping with it, so no detection should be matched with this face. The blue rectangles denote the true positives and yellow rectangles denote the false positives in the desired matching.

so they can be connected to at most one detection through M as well. Mathematically, the desired matching M maximizes the cumulative matching score while satisfying the following constraints:

$$\forall d \in D, \exists l \in \{L \cup N\}, \qquad d \xrightarrow{M} l; \tag{2.3}$$

$$\forall l \in L, \nexists d, d' \in D, d \neq d', \qquad d \xrightarrow{M} l \wedge d' \xrightarrow{M} l. \tag{2.4}$$

The determination of the maximum weight matching in a weighted bipartite graph has an equivalent dual formulation as finding the solution of the minimum weighted (vertex) cover problem on a related graph. This dual formulation is exploited by the Hungarian algorithm [58] to obtain the solution for the former problem. For a given image, we employ this method to determine the match between detections and

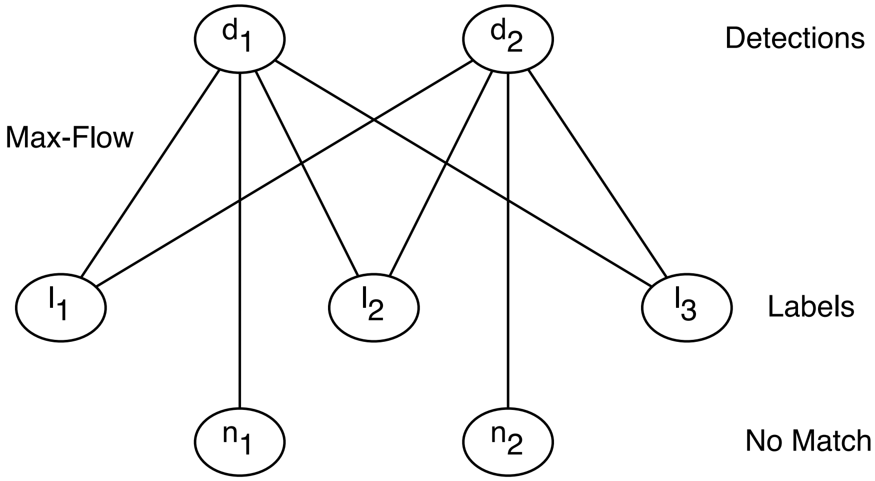


Figure 2.14. Maximum weight matching in a bipartite graph.

We make an injective (one-to-one) mapping from the set of detected image regions d_i to the set of image regions l_i annotated as face regions. The property of the resulting mapping is that it maximizes the cumulative similarity score for all the detected image regions.

ground-truth annotations.[8] The resulting similarity score is used for evaluating the performance of the detection algorithm on this image.

2.2.3.2 Evaluation metrics

Let d_i and v_i denote the i^{th} detection and the corresponding matching node in the matching M obtained by the algorithm described in Section 2.2.3.1, respectively. We propose the following two metrics for specifying the score y_i for this detection:

- **Discrete score (DS)** :

$$y_i = \begin{cases} 1, & \text{if } S(d_i, v_i) > 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

[8] Our implementation of the matching process is available as part of the FDDB evaluation toolkit at http://vis-www.cs.umass.edu/fddb/evaluation.tgz.

- **Continuous score (CS)**:

$$y_i \quad = \quad S(d_i, v_i). \tag{2.6}$$

For this discussion, we assume a region-based output from a face detector, and ignore any additional inference about properties such as head pose [44, 53, 63, 82, 93] and the location of facial landmarks [109]. The scope of this work is limited to the evaluation of a region-based output alone.

2.3 Experimental Setup

For an accurate and useful comparison of different face detection algorithms, we recommend a distinction based on the training data used for estimating their parameters. To this end, the FDDB data set is split into ten mutually exclusive sets, or *folds*, such that the number of faces appearing in the images in each of these folds is approximately equal across all of these ten folds.[9] We use these folds to propose the following experiments:

EXP-1: 10-fold cross-validation

This experiment specifies that only the images from nine (out of ten) folds are used for estimating the parameters of the face detector. The trained detector is then used to detect faces in the images in the remaining one fold. This process of training on nine folds and testing on one fold is performed for each of the ten choices for the test fold.

[9] The ten folds used in our experiments are available at `http://vis-www.cs.umass.edu/fddb/FDDB-folds.tgz`.

EXP-2: Unrestricted training

For this experiment, data outside the FDDB data set is permitted to be included in the training set. The above-mentioned ten folds are used separately as test sets.

Note that, in both of these experiments, the lists of detected faces are reported separately for each of these ten folds. Each of the detected faces is expected to have a real-valued score associated with it, which denotes the detector's confidence in this detection. A threshold over this confidence score is varied to generate different points on a receiver operating characteristic (ROC) curve to report the cumulative performance of the given face detector.[10] A software implementation of this procedure of generating the performance curves for a list of detected faces in the FDDB data set is available at `http://vis-www.cs.umass.edu/fddb/evaluation.tgz`. All of the performance curves included in this chapter are generated using this software.

Having established a benchmark for evaluating different face detectors, we are now prepared to discuss the details of some approaches for face detection. In the next section, we describe the steps followed by a typical face detection system and discuss the details of the standard Viola-Jones [108] face detector. We will use this detector to develop a contextual face detection algorithm, which will be presented in Section 2.5.

2.4 A typical face detection approach

Given an image, a typical face detection system follows the following three steps (illustrated in Figure 2.15):

1. Sample candidate image regions,

[10] Typically, an ROC curve is a plot of true positive rate vs. false positive rate. Since there are millions of non-face regions in an image collection, the false positive rate is expected to be very low for most of the detection algorithms. For better visualization of the performance of different face detectors, our performance curves include a plot of true positive rate vs. false positives instead.

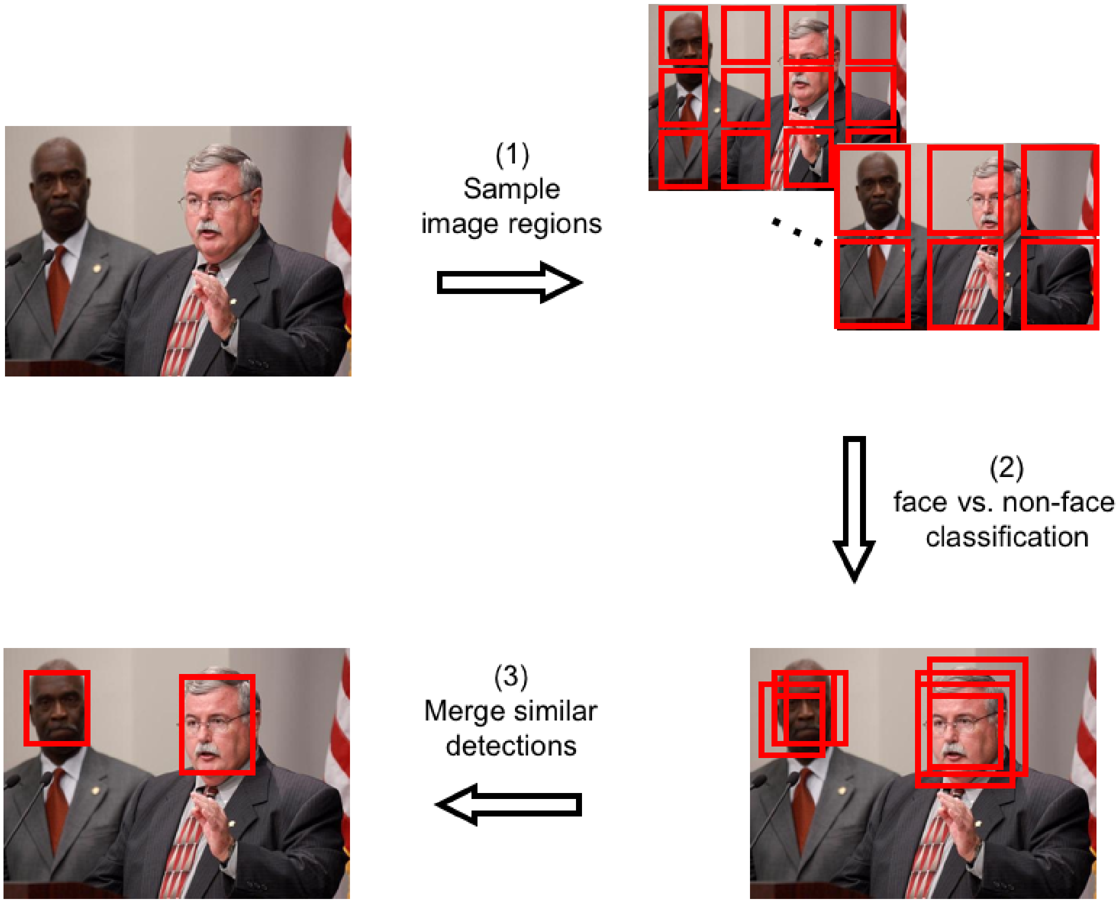


Figure 2.15. Steps followed by a typical face detection system.

2. Classify each of these image regions as a face or non-face region, and

3. Post-process the detected face regions to merge overlapping detections and remove spurious detections.

While modeling dependencies among these three steps may improve the performance of a face detector, the inference in such a joint model is likely to be computationally expensive. Similar to most of the existing face detection systems, we also do not model these dependencies and focus on developing better alternatives for the second step of face classification. For the other two steps, we follow standard approaches, the details of which are discussed later in Section 2.6.

2.4.1 Classifiers for face detection

A variety of statistical techniques, including naïve Bayes, neural-networks, and multi-layer perceptrons, have been explored for the binary classification task of determining whether a given image region is a face region. Here, we summarize a few face detectors that represent key progress in this area of research. The reader is referred to Yang et al. [114], and Hjelmas and Low [37] for detailed surveys of face detection research. In particular, we discuss three approaches that achieved very impressive results, and provide useful baselines for evaluating other face detection algorithms.

- **Neural-networks**. Rowley et al. [88] trained several multi-layer perceptrons with different receptive fields to predict the classification label for a given image region. An arbitration perceptron is trained to combine the output of all of these trained neural networks for generating the final prediction. While they achieved impressive results on the image from MIT+CMU data set (discussed in Section 2.2.1), it remains unclear how to determine a good set of parameters (e.g., number of layers and hidden units) for their neural network to perform well on a new image collection.

- **Parts-based model**. Schneiderman and Kanade [90] formulated the problem of face detection as the determination of the maximum a posteriori estimate of the presence of a face given the observed image patch. In their model, an image region is represented as a an ordered set of multiple sub-regions. Intuitively, these sub-regions correspond to different parts of a face such as the eyes and the nose. This model inspired much of the later work on parts-based representation for general object detection as well. They also showed impressive face detection results on MIT+CMU data set, but this approach is computationally expensive.

- **A cascade of AdaBoost classifiers**. The Viola-Jones detector [108] is considered a significant advancement in face detection not only because it achieved a high level of true positive rate with very low number of false positives, but also, through clever engineering, it was one of the first real-time face detector with a processing speed of 15 frames per second. Their detector was able to achieve such high level of performance in an efficient manner because of: (1) the use of an effective data structure called integral image representation for evaluating responses of Haar-like features; (2) an AdaBoost-based learning framework for obtaining strong classifiers from multiple weak classifiers; and (3) the use of a cascade of these classifiers for early rejection of non-face regions.[11]

All of the above approaches are supervised learning techniques. Supervised learning relies on the assumption of similarity between the distribution of training and test instances. However, in practice there are often significant differences between these distributions. These differences arise due to the cost of collecting large training data sets and also to the difficulties in obtaining training instances from a particular target test domain.[12] In face detection, it may be infeasible to collect training data for the enormous variety of domains in which face detection is useful. In realistic applications, then, we can expect to encounter domains at test time for which we have seen little training data. Furthermore, even when doing face detection in domains for which we do have significant training data, we may be able to perform significantly better classification by conditioning our classifier's output on the specifics of

[11] Further details of these three steps of the Viola-Jones' detector are not relevant to our discussion, and can be found in their original paper [108].

[12] We use *domains* to denote distinct modes in the data distribution. Note that in the limit of infinite data with samples from all the possible modes of the data distribution, there is no distinction between the source and target domain, but in practice, this distinction is manifested due to the limitations of the process of data collection.

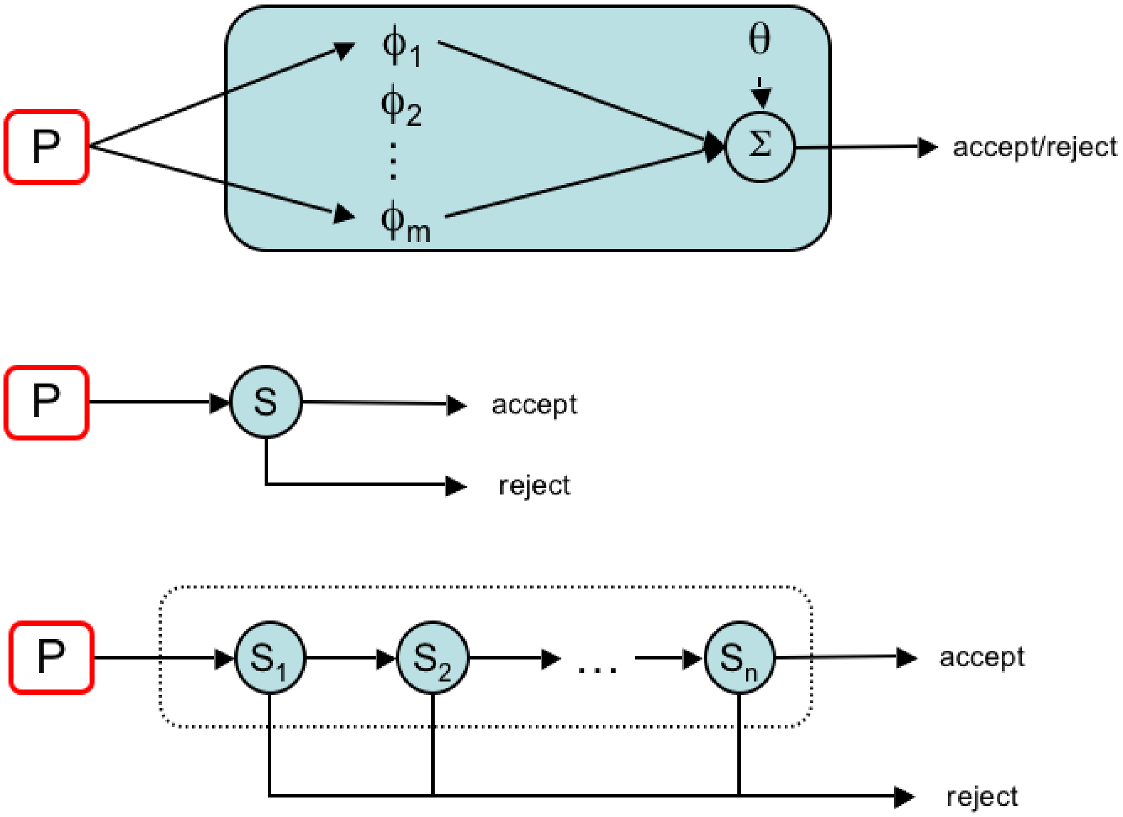


Figure 2.16. The classification cascade of the Viola-Jones face detector.

Given an image patch P, a set of features are Φ are computed and fed into a binary classifier (**top**; shown succinctly in **middle**). The Viola-Jones face detector (**bottom**) uses n such classifiers to define a cascade that instantaneously rejects a patch that is rejected by any of the n classifier. As a result, an image region is classified as a face region if and only if it is accepted by all the classifiers in the cascade.

the domain. That is, a generic classifier is unlikely to define the same classification boundary as one that has been adapted to a specific domain.

In the next section, we present a method for adapting pre-trained classifiers to improve performance in a new test domain. While this method could be applied to any choice of face classifier, we only present the details for adapting the classification cascade of the Viola-Jones detector (shown in Figure 2.16). We omit other details of the Viola-Jones detector such as the Haar-like features and the AdaBoost-based learning framework because they are not necessary for this discussion.

2.5 An online, adaptive cascade of classifiers

Most of the work on domain-adaptation [17, 23, 24, 61, 121] addresses the case in which a small number of labeled examples are available from the target domain. We defer a discussion of these approaches to Section 2.7. Here, we focus on the extreme case in which no labeled data is available for the new domain. We also assume, as described below, that we do not have access to the original training data from which the original classifier was derived. Furthermore, we assume that there is no known relationship among our *test* images. That is, we assume that each new test image may represent a new domain for the face detection problem. This means that there is only limited information to share across images. Hence our method re-adapts a pre-existing classifier to each new image it encounters. We demonstrate a dramatic increase in the state-of-the-art performance on the FDDB benchmark (Section 2.2.2), which shows that there is a surprisingly large amount to be gained by adapting a classifier, even using the information in just a single image.

Our domain-adaptation approach exploits the structure in the appearance of the faces co-occurring in an image to predict the detection label collectively for all the candidate regions in an image. This differs from the typical approach (as described in the previous section) of applying a classifier to each of these regions *independently* [88, 90, 100, 108].

Consider the image shown in Figure 2.17. A detector is likely to fail on the face of the person sitting in the left-bottom corner because of the shadow on the left half of this face. Since the shadow is relatively small on the two faces in the right half of the image, a good detector may successfully detect these faces. These two "easy-to-detect" faces could subsequently be used to infer common structure in the appearance of all the faces in this image, allowing us to normalize the "harder-to-detect" candidate face regions and ultimately classify them correctly. This same

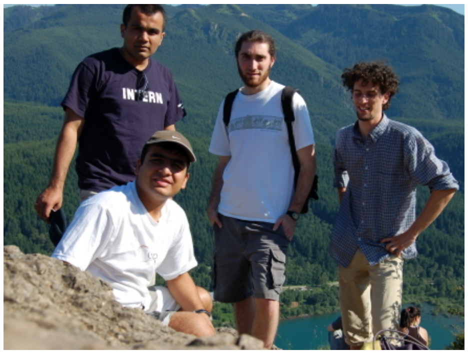


Figure 2.17. Easy-to-detect faces could help identify hard-to-detect faces. There is a shadow in the right half of all the four faces. The shadow is stronger on the two faces on the left, making them more difficult to detect than the other two faces. A detector that could learn the shadow pattern from the easy-to-detect faces could normalize the other faces to reduce their difficulty of detection.

reasoning can be applied to background patches, reducing both false negatives and false positives.

One naïve way to implement the above intuition is to scan the image for high-confidence faces, and then adapt the detection model according to the high-confidence face and non-face regions. This two-stage process has two problems. First, it can lead to over-fitting to the first stage predictions. Second, it represents a substantial increase in computation. We avoid the issue of over-fitting to new observations by using a Bayesian model with a strong prior. Furthermore, to minimize the increase in computation, we choose Gaussian process regression as the Bayesian model because its solution can be analytically computed. We discuss the details of this model in Section 2.5.3. But, first we present our formulation of face detection as regression, followed by a brief introduction of Gaussian process regression in Section 2.5.2.

2.5.1 Face detection as regression

In Section 2.4.1, we discussed the formulation of the problem of face detection as a classification task. Here, we argue that regression – or the problem of making real-valued predictions – is a more suitable formulation for developing a contextual face detector.

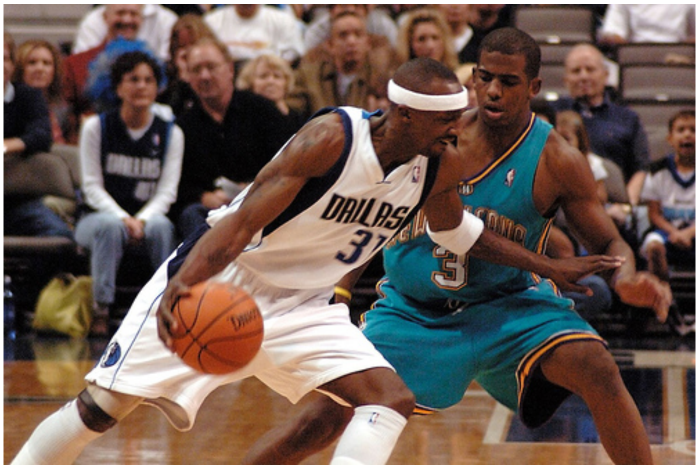


Figure 2.18. Multiple modes in appearance quality of face regions.

The resolution of the faces of people in the audience is much lower than the resolution of the faces of the players. Thus, it is likely that there is little commonality in the structure of appearance between these two classes of faces in this image.

The quality of appearance of a face region in an image depends on several factors including the pose of the person, the distance of the face from the camera, and the occlusion of the face from other objects present in the scene. For instance, in Figure 2.18, the resolution of the faces of people in the audience is much lower than the resolution of the faces of the players. Thus, it is likely that there is little commonality in the structure of appearance between these two classes of faces in this image. Similarly, in a different image, there might be more than two such modes in the distribution of face appearances present in a single scene.

As described in the previous section, our approach exploits the common appearance structure among the face regions in an image. Due to the potential presence of multiple modes in a single image, we formulate face detection as solving a regression problem rather than a classification problem. In this formulation, our approach predicts similar detection scores for image regions that are similar in appearance. In other words, the detection scores for the faces in the audience are encouraged to be similar to each other but may be different from the detection scores for the face regions corresponding to the players. Although this effect can be achieved using an ensemble of classification models, specifying a Bayesian model[13] with an appropriate family of priors for such an ensemble is non-trivial. For regression, on the other hand, Gaussian process prior provides a straightforward way to implement the corresponding Bayesian formulation.

2.5.2 Gaussian process regression

A Gaussian process refers to a stochastic process for which every finite set of samples is jointly Gaussian. When a Gaussian process prior is used in a Bayesian regression model for inferring continuous valued output, the resulting regression is called Gaussian process regression (GPR) (see Rasmussen and Williams [87] for further details). Consider the standard regression model with Gaussian noise:

$$y = \mathbf{x}^T \mathbf{w} + \eta, \tag{2.7}$$

where y is the target variable, $\mathbf{x}$ is the input vector, and $\eta \sim \mathcal{N}(0, \sigma_n^2)$. Let us assume a zero mean Gaussian prior on $\mathbf{w}$, i.e., $\mathbf{w} \sim \mathcal{N}(0, \Sigma_p)$. The conditional likelihood and the posterior distribution are respectively given by

$$\begin{aligned} p(\mathbf{y}|\mathbf{X}, \mathbf{w}) &= \mathcal{N}(\mathbf{X}^T \mathbf{w}, \sigma^2 I), && (2.8) \\ p(\mathbf{w}|\mathbf{X}, \mathbf{y}) &= \mathcal{N}(\frac{1}{\sigma_n^2} A^{-1} \mathbf{X}\mathbf{y}, A^{-1}), && (2.9) \end{aligned}$$

where $A = \sigma_n^{-2} \mathbf{X}\mathbf{X}^T + \Sigma_p^{-1}$. Subsequently, the prediction for a new example $\mathbf{x}_*$ is given by

[13] We need the Bayesian formulation to introduce a strong prior to avoid over-fitting to the new observations from the easy-to-detect faces.

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\frac{1}{\sigma_n^2}\mathbf{x}_*^T A^{-1}\mathbf{X}\mathbf{y}, \mathbf{x}_*^T A^{-1}\mathbf{x}_*). \tag{2.10}$$

Instead of using the original representation for data $\mathbf{x}$, if a function $\phi(\cdot)$ is used to project $\mathbf{x}$ into a (potentially) higher-dimensional space, then the resulting prediction for a new example follows the distribution

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\frac{1}{\sigma_n^2}\phi(\mathbf{x}_*)^T B^{-1}\Phi\mathbf{y}, \phi(\mathbf{x}_*)^T B^{-1}\phi(\mathbf{x}_*)), \tag{2.11}$$

where the terms $\Phi = \Phi(\mathbf{X})$ and $B = \sigma_n^{-2}\Phi(\mathbf{X})\Phi(\mathbf{X})^T + \Sigma_p^{-1}$ are used for notational convenience. Rearranging a few terms, this equation is equivalent to

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\phi_*^T\Sigma_p\Phi(K + \sigma_n^2 I)^{-1}\mathbf{y}, \phi_*^T\Sigma_p\phi_* - \phi_*^T\Sigma_p\Phi(K + \sigma_n^2 I)^{-1}\Phi^T\Sigma_p\phi_*). \tag{2.12}$$

where $\phi_* = \phi(\mathbf{x}_*)$, and $K = \Phi^T\Sigma_p\Phi$. Since all the terms involving the projection function ϕ occur in the form $\phi(\mathbf{x})^T\Sigma_p\phi(\mathbf{x}')$, an appropriate covariance function or *kernel* $k(\mathbf{x}, \mathbf{x}')$ can be used to avoid an explicit representation of the feature space. Thus, the resulting predictive distribution becomes

$$p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = \mathcal{N}(\mu(\mathbf{x}_*, \mathbf{X}, \mathbf{y}), \sigma(\mathbf{x}_*, \mathbf{X})), \tag{2.13}$$

where

$$\mu(\mathbf{x}_*, \mathbf{X}, \mathbf{y}) = K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I)^{-1}\mathbf{y}, \tag{2.14}$$

$$\sigma(\mathbf{x}_*, \mathbf{X}) = K(\mathbf{x}_*, \mathbf{x}_*) - K(\mathbf{x}_*, \mathbf{X})(K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I)^{-1}K(\mathbf{x}_*, \mathbf{X})^T. \tag{2.15}$$

These mean and variance terms are used in the next section to re-compute the prediction scores for instances near the classification boundary.

2.5.3 Online domain-adaptation

Let S denote a classifier based on the sign of the prediction value from a function $f(\cdot)$, i.e.,

$$S(\mathbf{x}|f) \triangleq \operatorname{sgn}(f(\mathbf{x})). \tag{2.16}$$

Let us assume that the probability of error of f is monotonically non-increasing with $|f(\mathbf{x})|$. In other words, the large prediction values (both positive and negative) are more likely to be correct than the small prediction values. This assumption further suggests that the classification label obtained by the classifier S for the points near the classification boundary (i.e., 0) may not be reliable. For the face detection problem, this assumption suggests that the pre-trained classifier can confidently accept unoccluded, in-focus, or "easy-to-detect" faces, and reject several non-face regions from a given image. This classifier is assumed to generate high prediction values for these easy acceptances and rejections. Consequently, the decision for the faces with low prediction values is assumed to be more difficult as compared to the decision for the regions with high prediction values.

Here, we propose to update the scores for the data instances with low prediction values from the pre-trained classifier by encouraging consistency in the final prediction values. In other words, if two data points $\mathbf{x}_1$ and $\mathbf{x}_2$ are similar to each other, then the corresponding predictions $f'(\mathbf{x}_1)$ and $f'(\mathbf{x}_1)$ should also be similar to each other. To this end, we define a small margin around the classification boundary. The data points with prediction values outside this margin are used to learn a Gaussian process regression model, which is used to update the prediction values of the data points with prediction values lying inside the margin. Figure 2.19 illustrates the intuition for this classifier adaptation, and the formal description is included below.

Given $\epsilon > 0$, define the *in-margin set* $\mathbf{X}_m \subseteq \mathbf{X}$ as

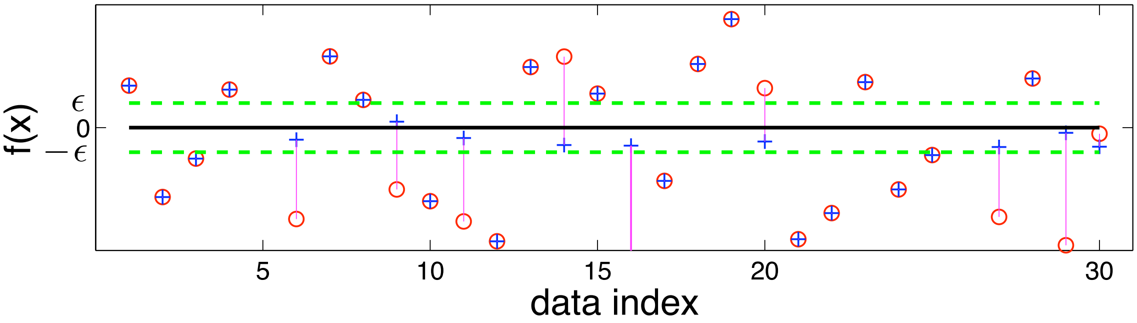


Figure 2.19. An illustration of online domain-adaptation.

Let $f(x)$ denote the output of a classifier on a data point x. Consider an ϵ margin (green dotted line) around the classification boundary (black solid line). For points lying in the margin the classifier is not very certain about the predictive label. The proposed method updates the scores for the points in this margin based on their similarity to the other points for which the classifier is relatively more confident about the classification label. The original classification output is shown using blue '+,' whereas the updated output (obtained using Equation 2.19) is shown using red 'o.'

$$\mathbf{X}_m \triangleq \{\mathbf{x} \in \mathbf{X} \text{ s.t. } |f(\mathbf{x})| < \epsilon\}. \tag{2.17}$$

Similarly, define the *out-of-margin set* $\mathbf{X}_o \subseteq \mathbf{X}$ as

$$\mathbf{X}_o \triangleq \mathbf{X} \setminus \mathbf{X}_m. \tag{2.18}$$

Using the mean (Equation 2.14) and variance (Equation 2.15) terms of the predictive distribution

$$f'(\mathbf{x}) \;=\; \begin{cases} f(\mathbf{x}) & \text{if } |f(\mathbf{x})| > \epsilon \\ \mu(\mathbf{x}, \mathbf{X}_o, \Phi(\mathbf{X}_o)) - \sigma(\mathbf{x}, \mathbf{X}_o) & \text{otherwise.} \end{cases} \tag{2.19}$$

Finally, the classifier is defined as

$$S'(\mathbf{x}|f, \epsilon) \;\triangleq\; \operatorname{sgn}(f'(x)). \tag{2.20}$$

In our face detection experiments, we use the noisy squared-exponential function as the covariance function K_θ, i.e.,

$$K_\theta(\mathbf{x}_i, \mathbf{x}) = \nu^2 \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2l^2}\right) + \sigma_{gpn}^2 \delta_{\mathbf{x}_i,\mathbf{x}}, \tag{2.21}$$

where ν and l refer to the weight and scale-length parameters of the squared-exponential function, and σ_{gpn} is the variance of the added noise when $\mathbf{x}_i$ and $\mathbf{x}_j$ are identical. Also, $\delta_{\mathbf{x}_i,\mathbf{x}}$ is a Kronecker delta. Hereafter, we use $\theta^T = [\nu, l, \sigma]^T$ to refer to the set of all of these three parameters of the covariance function K.

Algorithm 2 Cascade of adaptive classifiers.

Require: input X, classifier cascade $\{S\}_{1\ldots n}$, margin $\epsilon \geq 0$, covariance function $k(\cdot,\cdot)$

```
1: **for** $n = 1$ to $N$ **do**
2:     Let the stage classifier $S_n := \text{sgn}(f_n(x))$
3:     $X_m \leftarrow \{x \in X | \, |f_n(x)| < \epsilon\}$
4:     $X_o \leftarrow X \setminus X_m$
5:     $Y_o \leftarrow f_n(X_o)$
6:     $\theta^* \leftarrow \underset{\theta}{\text{argmax}} \, \log p(Y_o | X_o, \theta, k)$, where $\theta$ are the parameters of $k$
7:     $\forall x \in X_o$, $f'_n(x) \leftarrow f_n(x)$
8:     $\forall x \in X_m$, compute $f'_n(x)$ using Equation 2.19.
9:     $X \leftarrow \{x \in X | f'_n(x) > 0 \}$
10: **end for**
```

The steps for our method for domain-adaptation for a pre-trained cascade of classifiers are shown in Algorithm 2. We refer to the new classifier as *cascade of adaptive classifiers*. Note that in this classifier, for each test image, the parameters θ of the covariance function (Equation 2.21) are estimated by maximizing the likelihood of observing the data and prediction values in the out-of-margin set X_o (Step 6 of Algorithm 2). In our experiments, we observed little dependency on the initialization of this maximization step. The other parameter ϵ is chosen by maximizing the true positive rate at the observed number of false detections equal to 10% of the total number of true faces on a held-out data set.

2.6 Experiments

Here, we discuss our observations for the EXP-2 experimental setup (see Section 2.3 for details) on the FDDB benchmark. For our experiments, we use the OpenCV[14] implementation of the Viola-Jones face detector. In particular, we use a pre-trained cascade of classifiers for frontal face detection, which is included in the OpenCV version 1.0 distribution as `haarcascade_frontalface_default.xml`. This cascade evaluates image regions of width (w_r) and height (h_r) equal to 24 pixels. In the following experiments, we set the parameter specifying the minimum number of neighbors required for a valid detection to 0, and use the `CV_HAAR_DO_CANNY_PRUNING` flag.

We first present the observations related to the sampling of candidate image regions and the merging of overlapping detections. A discussion of these steps was deferred to here in Section 2.4.

2.6.1 Sampling candidate image regions

Given an image I and a real-valued scale-factor parameter s, the OpenCV implementation of Viola-Jones detector samples the candidate image regions as following:

1. Add all of the image regions $I(x, y, w_r, h_r)$[15] that lie entirely within the limits of image I, to the set of candidate image regions.

2. If at least one candidate image region is found in the previous step, down-scale the image I by a factor equal to s, and go to Step 1.

Denoting the width and height of the image I by w_I and h_I respectively, the number of scales at which the candidate image regions are sampled is given by

[14] http://sourceforge.net/projects/opencvlibrary/

[15] We use $I(x, y, w_r, h_r)$ to denote a rectangular region in the image I with (x, y) as the coordinates of the left-top corner, and w_r and h_r as its width and height respectively. Also note that for our experiments $w_r = h_r = 24$ pixels.

$$n_s \quad = \quad \log_s \min\left(\frac{w_I}{w_r}, \frac{h_I}{h_r}\right). \tag{2.22}$$

From Equation 2.22, it is clear that for a lower scale factor, a larger number of regions are evaluated for the given image. Figures 2.20 and 2.21 show the effect of the choice of scale-factor on the performance of the Viola-Jones detector using the evaluation metrics in terms of discrete and continuous scores (described in Section 2.2.3.2) respectively.

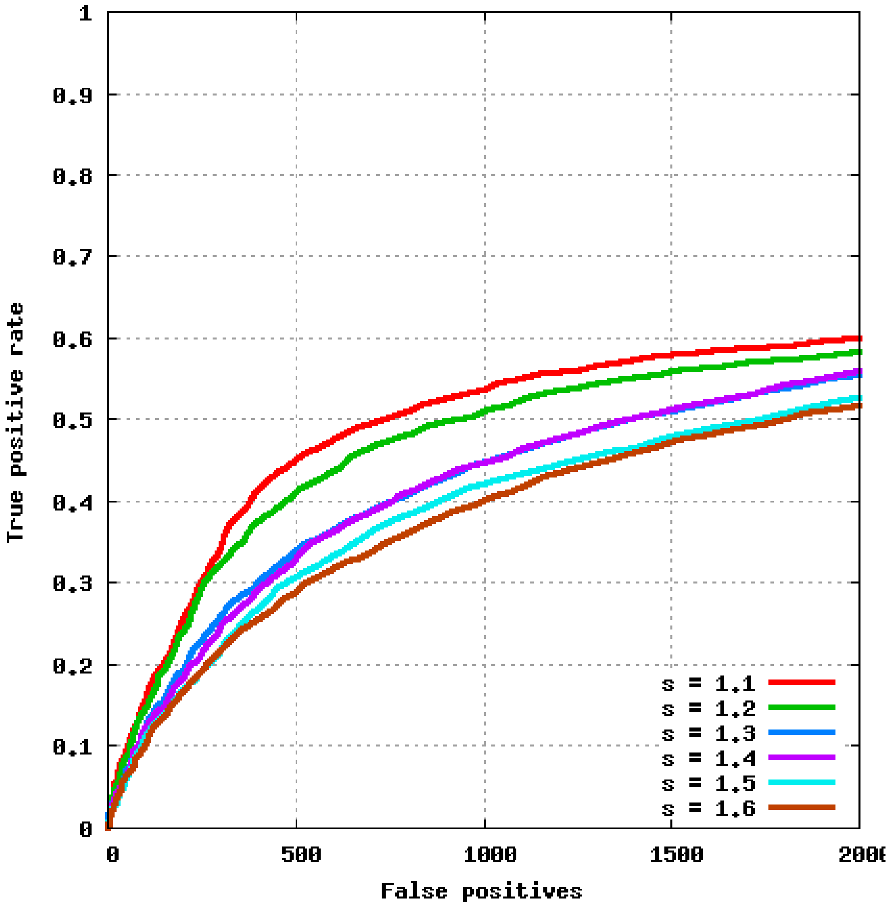


Figure 2.20. The effect of change in sampling scales using discrete score.

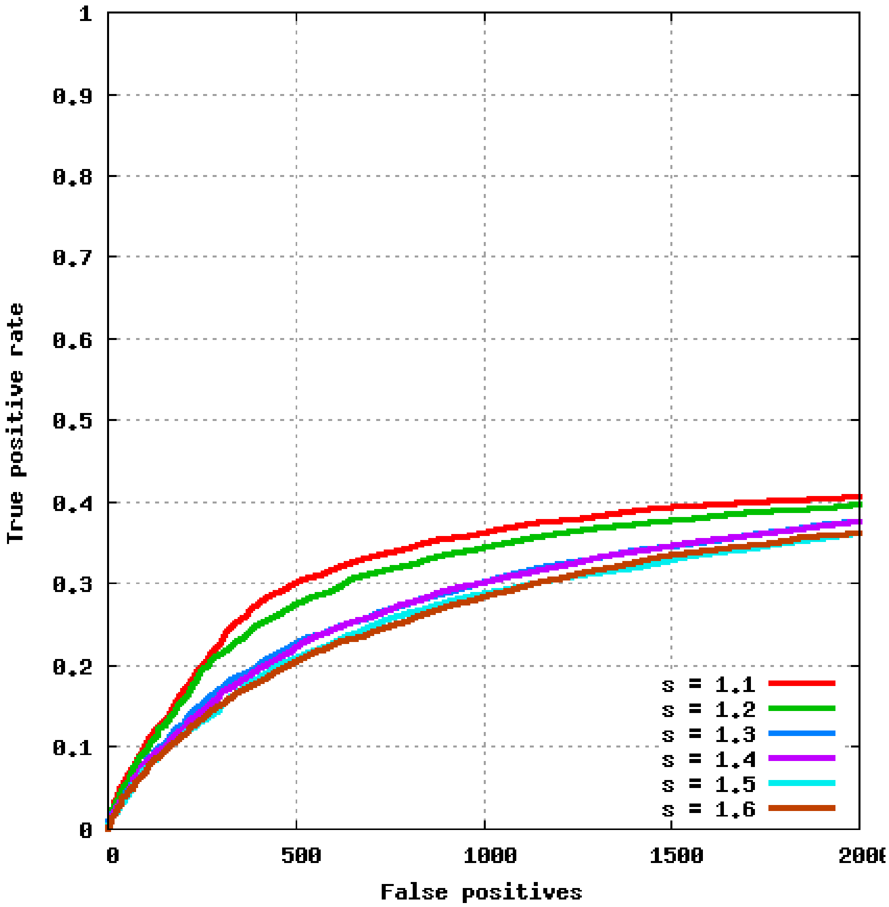


Figure 2.21. The effect of change in sampling scales using continuous score.

Among the two best performance curves, the performance for scale-factor $s = 1.1$ is slightly better than the performance for $s = 1.2$. The number of evaluations for the former choice of s is much larger than the number of evaluations for the latter choice of s. Considering this trade-off, we choose $s = 1.2$ in all of our further experiments.

2.6.2 Merging similar detections

As a result of the sampling approach described in the previous section, several overlapping regions can be considered candidates for being classified as face regions. Thus, it is likely that the presence of a face in an image would lead to the classification of a set of overlapping image regions as face regions. Note that according to the matching algorithm described in Section 2.2.3.1, each of these repeated detections correspond to an additional false positive. Hence, to remove unnecessary repeated detections, we only report a single detection for the above set of detections. The average location and extent over the set of detections are used to specify this representative detection.

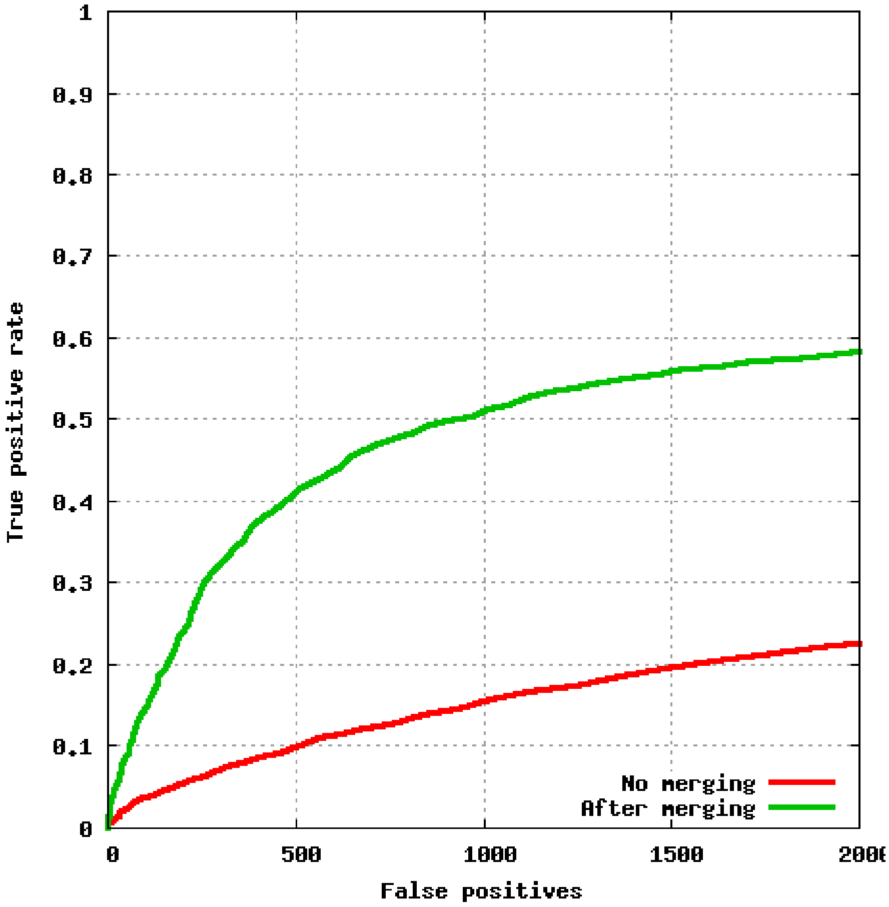


Figure 2.22. The effect of merging similar detections using discrete score.

The observations in Figures 2.22 and 2.23 confirm the intuition for achieving an improved performance by merging the overlapping detections. For all of the following experiments, we merge overlapping detections as a post-processing of a face detector's output.

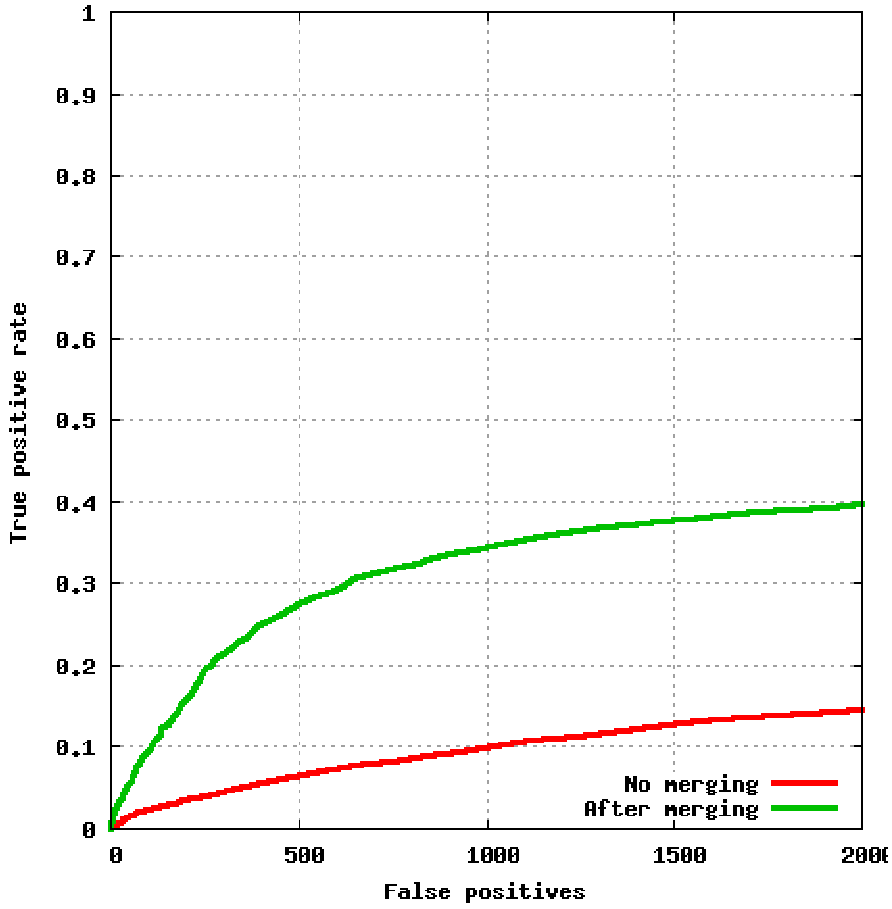


Figure 2.23. The effect of merging similar detections using continuous score.

2.6.3 Comparing different face detection systems

To the best of our knowledge, only the following implementations of face detection systems are available for public use:

- The OpenCV implementation of the Viola-Jones detector [108],
- Mikolajczyk's variant [75] of Schneiderman et al.'s approach [90],[16]
- Kienzle et al.'s [56] face detector.[17]

[16] http://www.robots.ox.ac.uk/~vgg/research/affine/face_detectors.html

[17] http://www.kyb.mpg.de/bs/people/kienzle/fdlib/fdlib.htm

The performance of Kienzle et al.'s face detector was very poor on our benchmark. Note that neither the source code of this system is available, nor did we have an access to sufficient parameters to optimize to obtain acceptable performance curves for our benchmark. Hence, we exclude this detector from our comparison of face detection systems.

In Figures 2.24 and 2.25, we present the performance curves for Viola-Jones detector, Mikolajczyk's face detector and our face detector. Since our detector uses the Viola-Jones detector and the base detection algorithm and Gaussian process regression for re-computation of detection scores, we refer to our detector as *VJ GPR*.

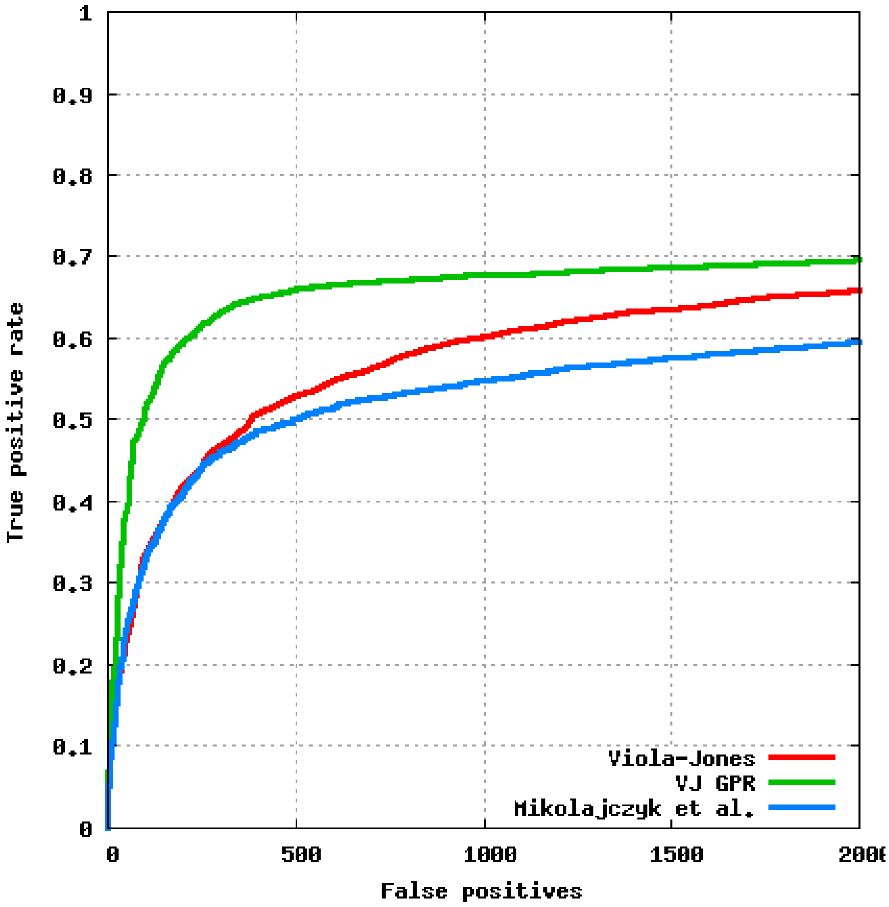


Figure 2.24. A comparison of different face detectors based on discrete score.

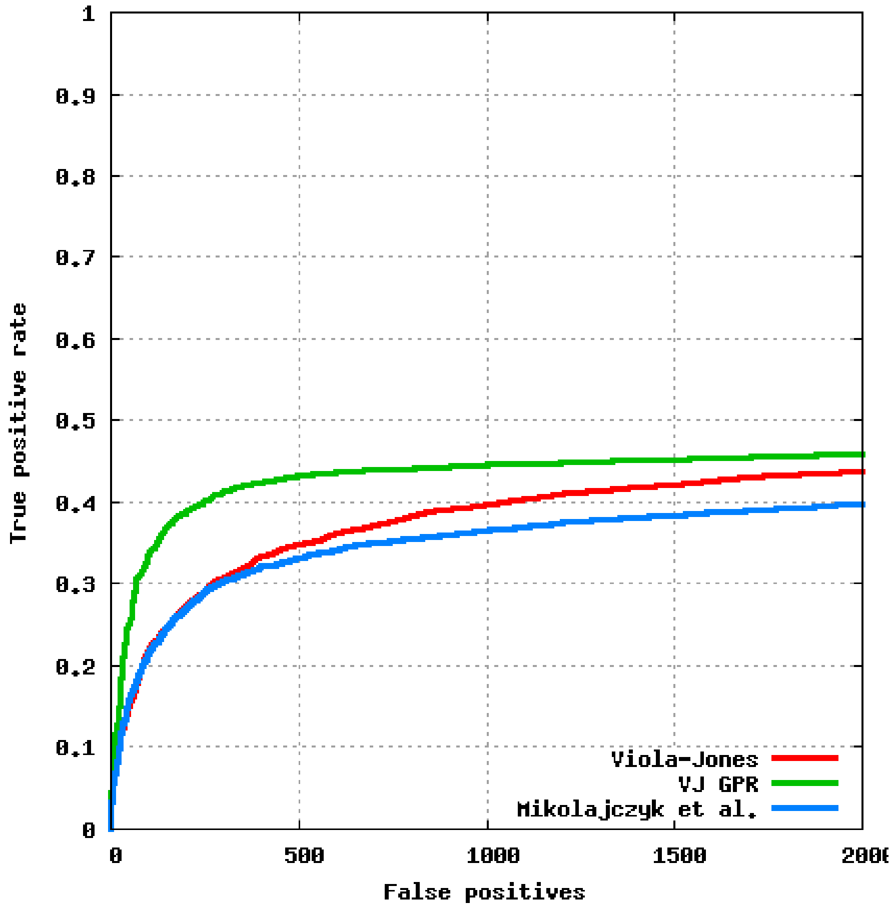
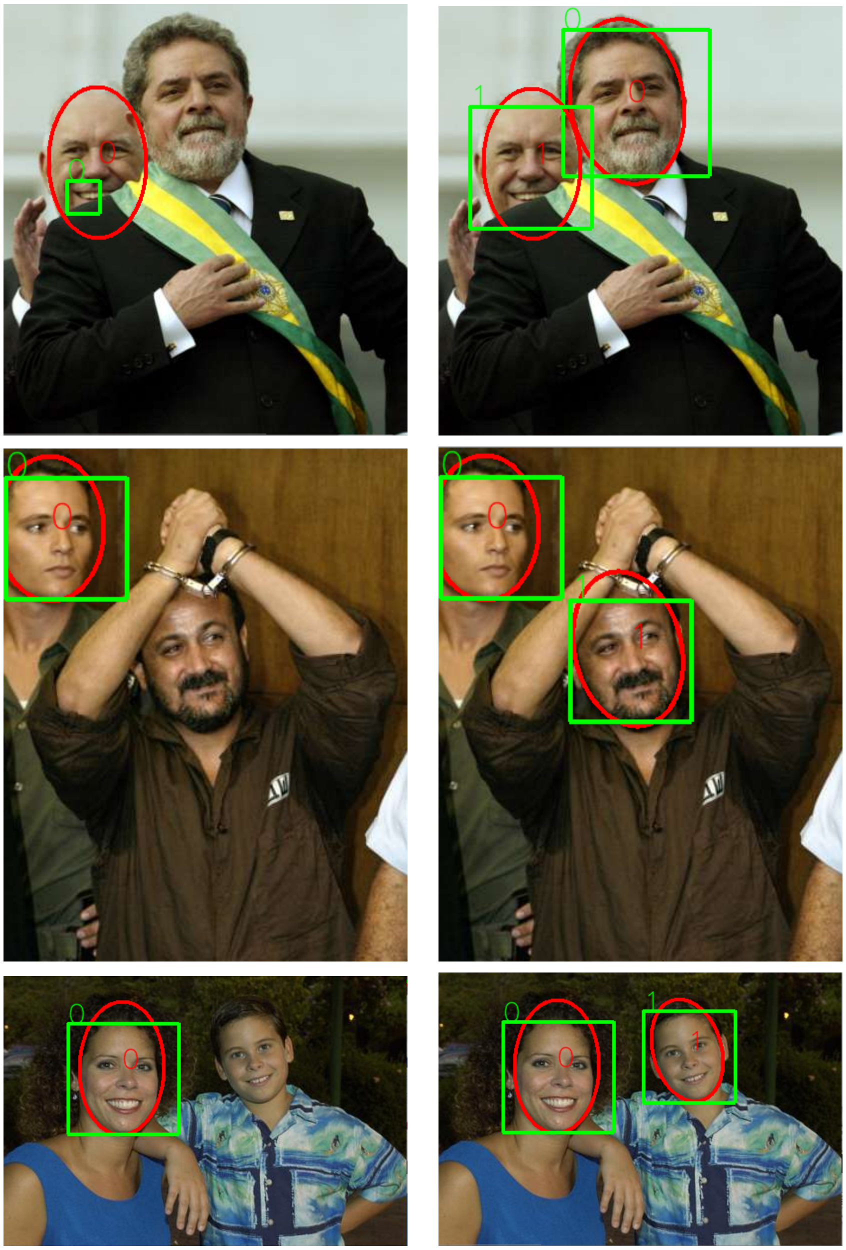


Figure 2.25. A comparison of different face detectors based on continuous score.

As seen in these performance curves, the number of false positives obtained from all of these face detection systems increases rapidly as the true positive rate increases. Note that the performance of all of these systems on the new benchmark are much worse than those on the previous benchmarks, where they obtain less than 100 false positives at a true positive rate of 0.9. Figure 2.26 shows the comparison of face detection results obtained by the Viola-Jones face detector and our adaptive face detector with parameter settings such that they obtain same false positive rates on the FDDB benchmark.

Note that we did not alter the extent of the detected regions from these face detection systems. It is likely to obtain better performance curves by changing the



Viola-Jones detector

Our adaptive detector

Figure 2.26. Detections obtained by two face detectors on some images from the FDDB data set.

The detections are denoted by green rectangles, whereas the matched ground truth face annotation is denoted by red ellipses. These face detection results are obtained using systems with identical false positive rate.

height or width systematically for all of the detected face regions to obtain more overlap between the detected regions and the annotated faces. Here, we focus on the fundamental improvements in face detection approaches, and hence an exploration of such post-processing techniques for improvements in performance are beyond the scope of this work.

Furthermore, the above experiments are limited to the approaches that were developed for frontal face detection, whereas our data set includes images of both frontal and non-frontal faces. This limitation is due to the unavailability of a public implementation of a multi-pose or pose-invariant face detection system. Nevertheless, the new benchmark includes more challenging examples of face appearances than the previous benchmarks. We hope that our benchmark will further prompt researchers to explore new research directions in face detection.

2.6.4 Evaluating the parameters of our online domain-adaptation method

Our method for online domain-adaptation uses the parameter ϵ to define a margin around the classification boundary. This parameter is chosen using cross-validation in the experiments in the previous section. Here, we study the effect of the choice of this parameter in more detail. Intuitively, when the margin around the classification boundary is too tight, only a few instances will lie in this margin. Hence, we expect our approach to show little improvement over the base detection algorithm (Viola-Jones detector, in these experiments). On the other hand, if the margin is too large, only a few instances will lie outside the margin, and hence our model is likely to generate the updated scores according to the generic prior model. Note that the performance in the latter case could be worse than the performance of the base detection algorithm. To summarize, as we increase the value of ϵ (starting from $\epsilon = 0$), we expect the performance to first improve, and then get worse. The above intuition is validated through the performance curves shown in Figures 2.27 and 2.28.

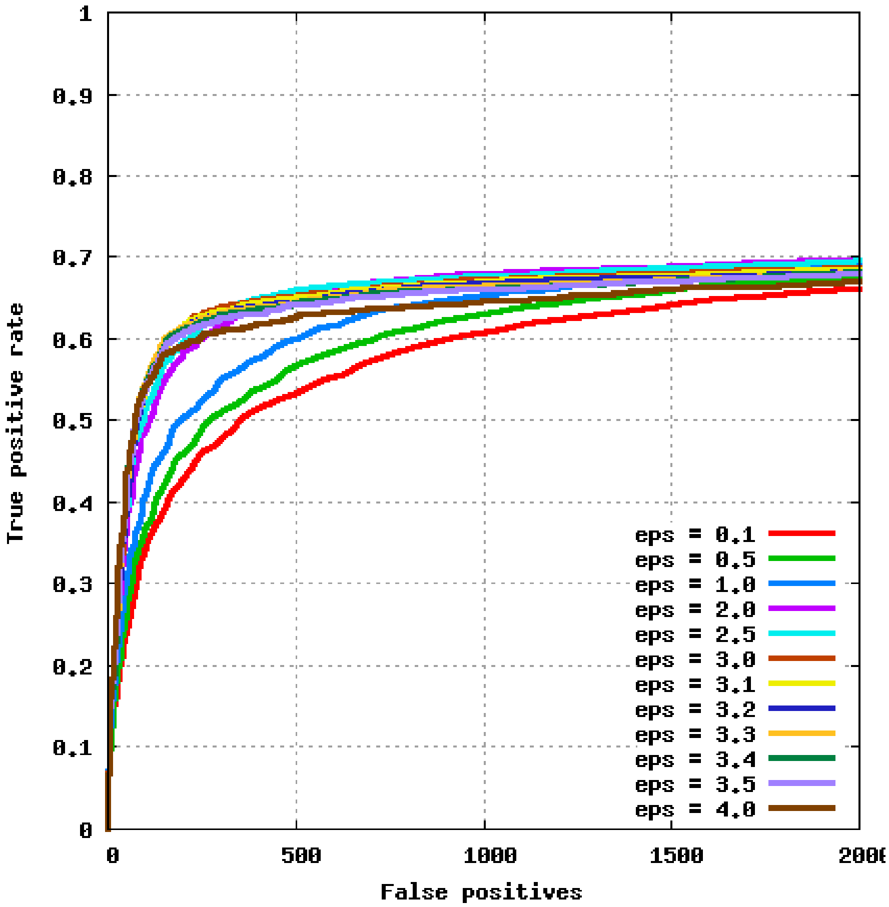


Figure 2.27. The effect of the choice of the margin parameter ϵ using discrete score.

In our next experiment, we evaluate the possibility of bootstrapping our approach to obtain further improvements in performance. Note that our approach treats the base detection algorithm as an oracle that provides a real-valued confidence score for all of the candidate image regions. It is conceivable to design an iterative approach, where the detection scores computed in one iteration are treated as the input for the re-computation of the detection scores in the next iteration. We implemented this iterative approach and observed that only a few iterations were required for the predictions to converge. Moreover, even after a single iteration, the performance curves were very similar to the performance curves obtained by using multiple iterations till the convergence of predictions. Also, since performing multiple iterations of score

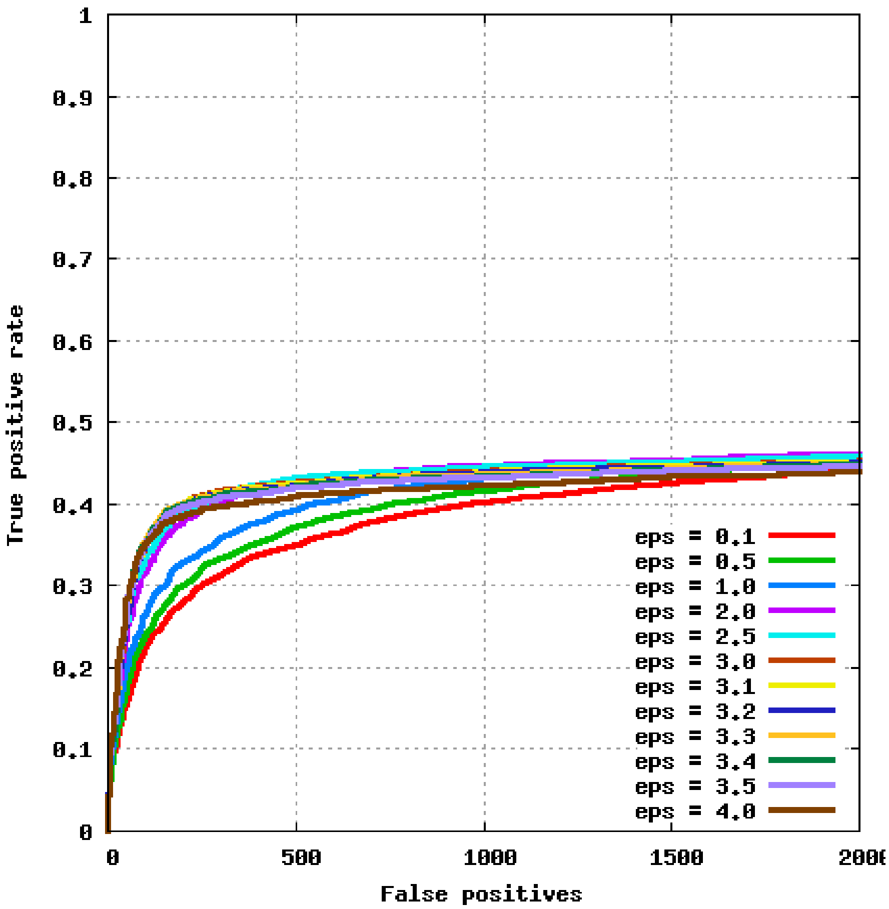


Figure 2.28. The effect of the choice of the margin parameter ϵ using continuous score.

computations significantly increases the computational cost of our approach, we used only a single iteration in all of the other experiments.

2.7 Other related work

Our proposal for domain-adaptation is similar to work in a variety of fields. Here, we discuss these related approaches and distinguish our method from the previous work.

Semi-supervised learning refers to the problem of learning from both labeled and unlabeled data. One common approach for handling unlabeled data is to construct a graph using pairwise similarities between both labeled and unlabeled training instances. The nodes corresponding to the labeled instances are annotated with the original labels, and *label propagation* [121] is performed to estimate the labels for unlabeled instances. Another related line of research uses the unlabeled data to estimate the underlying data density and move the classification boundary out of regions with high data density. For example, Lawrence and Jordan [61] presented a null-category noise model to push the unlabeled data out of the margin; and Szummer et al. [101] included a regularization term based on a local estimate of the mutual information between the data and the label distributions to move the classification boundary out of the high density data regions. The latter approach, also referred to as *information regularization*, was extended by Corduneanu et al. [23] for semi-supervised learning.

Our approach uses the similarity between the data points to update the detection score of the data points for which the predicted score from the pre-trained detector is near the classification boundary. This update effectively sparsifies the distribution of data around the original classification boundary. While the intuition behind this procedure is similar to those of the above-mentioned methods, these semi-supervised learning approaches assume that both of the labeled and unlabeled data are sampled from an identical underlying distribution. As described in the previous section, this assumption does not hold true for our problem setting.

The problem formulation used in this paper is very similar to the work in *domain-adaptation*. In domain adaptation, labeled data from one or multiple "source" domains is used to train models to perform well on a different yet related "target" domain. Daumé and Marcu [24] approach this problem by modeling the data distribution for each of these domains as a mixture of a global and a domain-specific component. This global component is inferred from the data of the source domain(s) and applied to the data of the target domain. Another approach to the domain-adaptation problem is to use models trained on the data from the source domain to label a subset of the unlabeled data from the unlabeled target domain and re-train the classifier on the combined labeled data set [112].

Most of the work in domain-adaptation suggests minimizing a convex combination of source and target empirical risk [17]. Thus, the classifier needs to be re-trained (repeatedly) from scratch for every new domain. For face detection, we argue that the distribution of face appearances varies significantly from one image to another. Hence, in the domain-adaptation setting, every image represents a new domain. Applying existing techniques for domain-adaptation would therefore be prohibitively slow for our problem formulation.

Our work could also be interpreted as regularizing the output of a face detection algorithm on the data manifold. Belkin et al. [9] proposed smoothing the discriminative function by controlling the complexity of the learned classifier through the norm of the desired function in the corresponding reproducing kernel Hilbert spaces. They further showed that this *manifold regularization* framework generalizes a large set of learning algorithms including ridge regression and support vector machines. Although this framework provides useful insights into the relation between the hypotheses for the original detector and the adapted detector, the infeasibility of re-training the classifier for a new test image prevents us from building on this work as well. A similar argument holds true for the relevance of the previous work related to the analysis of *covariate shift* [11].

To the best of our knowledge, our work is the first to approach domain-adaptation in a completely unsupervised and on-line setting. In other words, instead of training a new classifier from scratch for a new target domain, we adapt a classifier trained on a different source domain by encouraging smoothness of the output function. We present a simple, yet effective, method to perform this adaptation, and report state-of-the-art results in face detection using this approach.

2.8 Conclusions

We have shown that simply by adapting a black-box classifier so that its outputs are smooth with respect to a new test set, we can substantially improve its performance. The performance gain we have achieved on the FDDB face detection benchmark is dramatic, especially in view of the fact that the Viola-Jones classifier has remained a top contender in face detection accuracy since its introduction in 2004. (We note that FDDB is particularly difficult, including many profile views and other faces which the best detectors currently miss.)

While it is certainly worth asking whether semi-supervised methods, despite their greater computational burden, could be applied in this scenario, several problems have kept us from pursuing this question. First, the original training data for the Viola-Jones classifier is proprietary and unavailable for us to use. Thus, we must already accept an alternative training set than the one used to train the original published classifier. Second, the details of training the original Viola-Jones classifier are not completely specified in the literature. We have had difficulty reproducing the original results from the classifier, even after contacting one of the original authors of the paper.

Without the original training data and without a clearly specified training algorithm, the task of applying a semi-supervised method is especially daunting. This perhaps makes our approach even more appealing, since there is no need for either the original data or the original algorithms. In future work, we hope to characterize necessary and/or sufficient conditions for our approach to lead to an expected reduction in classification error.

CHAPTER 3
CONTEXTUAL FACE RECOGNITION

3.1 Introduction

Face recognition refers to the problem of determining the identity of persons from the appearance of their faces. We assume that the input to a face recognition algorithm is a *single image region* showing a person's face.[1] In a typical setting for face recognition, a few example images are selected for each individual to specify the appearance of his or her face. The set of all of these example images is referred to as the *gallery images*. For a new test face image (also known as a *probe image*), the identity of the person is determined as the best match in the gallery images.

One common approach for face recognition is to learn a model of the appearances of faces from the gallery images. Under large variations of parameters such as pose, lighting, and expression, modeling such distributions of appearances of face images is very difficult [120]. For the above-mentioned uncontrolled settings, it is relatively easy to build models for the *difference in appearance* of two face images – one model each for the *same* and *different* classes of image pairs. This formulation corresponds to the problem of determining if two face images are of the same person, which is commonly known as *face verification* or *face identification*. Note that the problem of face recognition can be formulated as solving multiple face identification problems as follows. First, the probe image is evaluated for the possibility of a match with each of the gallery images using a face identifier. Then, the gallery image with the

[1] The reader is referred to Zhou et al.'s work [119] for a discussion of face recognition in video sequences.

best identification result is selected to specify the identity of the probe image. This formulation is illustrated in Figure 3.1. In this chapter, we follow this approach for face recognition.

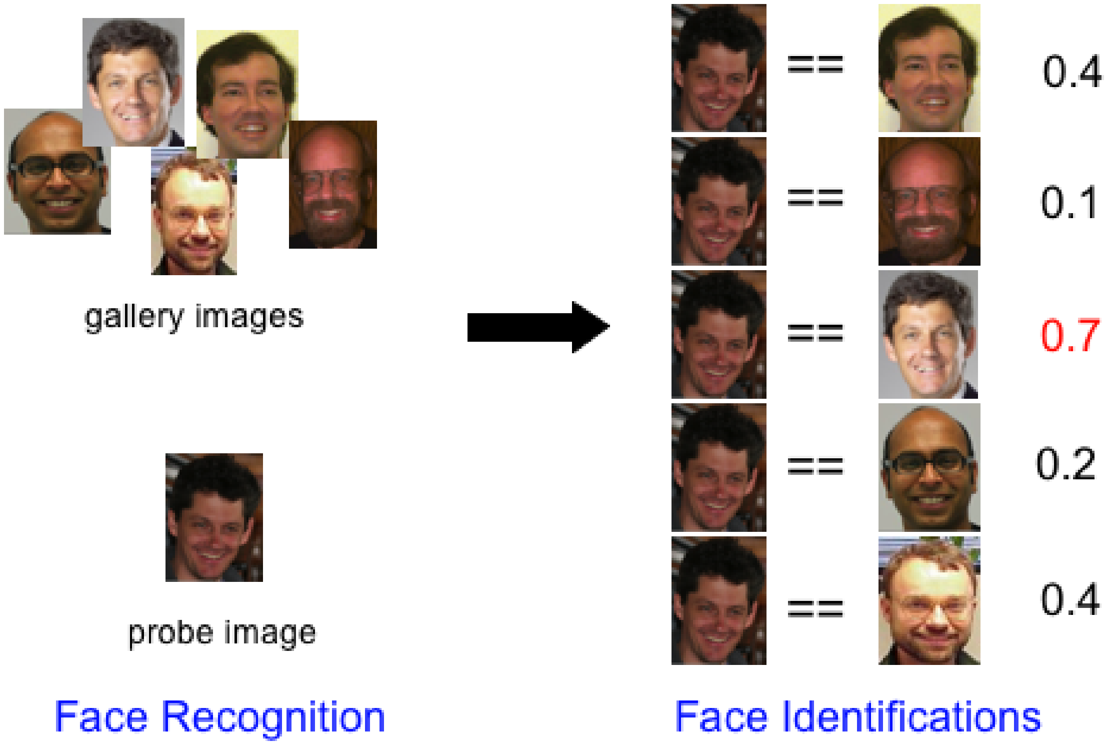


Figure 3.1. Face recognition formulated as solving multiple face identification problems. Given a set of gallery images and a probe image, we compute the face identification scores for the probe image with each of the gallery images. The gallery image with the maximum identification (indicated in red) is then used to specify the identity of the person in the probe image. The scores shown in this illustration are not the output of an actual face identifier.

While models for face identification are easier to learn than for face recognition, the following two challenges remain in face identification as well: first, faces of different individuals may have similar appearances; and second, the face of a single person may appear very different in two images due to variations in factors such as pose, makeup, and emotion.

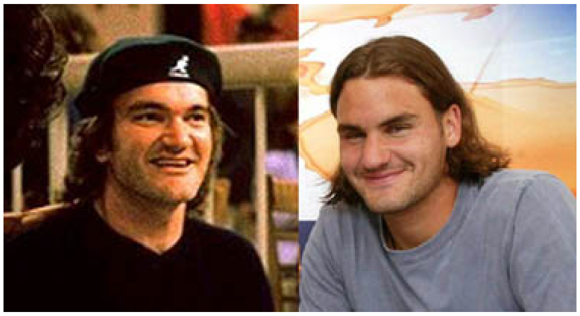
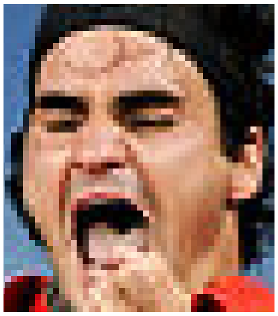


Figure 3.2. An example of a pair of different individuals with similar face appearances.

Film director Quentin Tarantino (**left**) and tennis player Roger Federer (**right**) have similar looking faces. Hence, disambiguating between these two identities based only on their facial features is difficult.



```
Roger Federer of Switzerland
celebrates after winning the
men's title against Novak
Djokovic of Serbia at the
U.S. Open tennis tournament
in New York.
```

Figure 3.3. An illustration where context helps in face recognition.

Is this a picture of Roger Federer after winning the U.S. Open tennis championship, or of Quentin Tarantino after winning an Academy Award for the movie Pulp Fiction? Since both of these individuals have similar-looking faces (as shown in Figure 3.2), it is very difficult to answer this question from the given image alone. This distinction, however, becomes trivial after looking at the caption for this image (shown on the right of the image).

For instance, as shown in Figure 3.2, tennis player Roger Federer and film director Quentin Tarantino have similar-looking faces. Now consider the image shown in Figure 3.3. Even after assuming that this image shows the face of either Federer or Tarantino, the disambiguation of the identity of the person shown in this image is challenging due to an extreme display of emotion in this face image. However, if a caption, e.g., the one shown to the right of the image in Figure 3.3, is also provided, an analysis of this caption would make it easier to identify the person shown in the above image as Federer.

In this chapter, we study the problem of face recognition in the above setting where a caption is provided for all images, which is typical for images appearing in news articles. We argue that the context generated from the analysis of the caption significantly reduces the number of competing identities for faces appearing in an image. To this end, we present a multi-modal probabilistic model *People-LDA* for the image-caption pairs. Our model uses the coherence of face images and their captions to obtain clusters of face images and of words that are closely related to a single person. To do this we incorporate a face identifier in a statistical topic-modeling framework.

First, in Section 3.2, we present the details of the face identifier used in our model. Next, in Section 3.3, we discuss the fundamentals of topic-models. In Section 3.4, we present our model for image-caption pairs. Then, we describe the setup used in our experiments in Section 3.5, and discuss our results in Section 3.7. Finally, we conclude this chapter with a discussion of possible future directions in Section 3.8.

3.2 Hyper-features based face identification

In an image, some regions are more useful than others to determine the class of the object shown in the given image. The same patches may not be very helpful in determining the identity of the shown object within the object-class. For instance, the presence of two tires in an image suggests that the image shows an automobile, but may not provide enough information to distinguish between Alice's Toyota Prius and Bob's BMW Z8. In other words, some image regions are useful for classification (i.e., "car"), whereas others are useful for identification (i.e., specific car brand or model). For face identification, regions of the latter type are useful because the object class ("faces") is already known and the goal is to determine the closest matching instance within the face class.

To select the image regions useful for identification, Ferencz et al. [29] proposed to model the *difference in appearances* for a pair of image regions as follows. They estimate two distributions for the expected distance between a pair of image regions, one each for the following two cases:

1. "same" pair, i.e., when the two image regions in the given pair show the same object,

2. "different" pair, i.e., when the two image regions in the given pair show different objects.

Both of these distributions are defined in terms of the appearance of only *one* of these two regions (hereafter referred to as the *left* image region; the other region in this pair is referred to as the *right* image region). In particular, they used easy-to-compute image features (or *hyper-features*) such as the location, intensity values, and directional edge-energies to represent the left image region. Several example pairs of images showing the same object are used to estimate the expected distribution of difference in appearance for the "same" pair of images. A similar distribution is estimated separately for the "different" pair of images. The difference between these two estimated distributions is used to determine the expected utility of the given image region, which is used to select a few regions from each image. This selection of regions is done using a single image, and hence can be done *once* for each of the gallery images, and does not need to be computed again for every probe image.

After the selection of useful image regions, Ferencz et al. used a generative model to combine the differences between the corresponding pairs of left and right image regions for the computation of the final identification score. Here, we present an alternative, more direct, modeling of this computation using a discriminative approach.

We show that our approach improves the identification performance of Ferencz et al.'s approach.

First, in Section 3.2.1, the notation used in the following discussion on hyperfeatures based systems is described. Next, in Section 3.2.2, the process of selecting image regions is described. Then, in Section 3.2.3, the details of Ferencz et al.'s generative model and our discriminative model for the computation of identification score for a pair of images are presented. Next, the performance curves for different face identification approaches are presented in Section 3.2.4. Finally, in Section 3.2.5, the suitability of each of the above two models is evaluated for the contextual face recognizer described in the previous section.

3.2.1 Preliminaries

Let I^L and I^R denote the left and right images respectively. We assume that the faces shown in these images are approximately frontal faces. In other words, any variations in the head pose has already been normalized using a face alignment algorithm.[2] Also, a binary random variable C is used to denote if I^L and I^R are examples of the same object.

For each of these images, a large number of regions are sampled at different scales and locations in the image. The j^{th} region in the left image is referred to as F_j^L. The image region most similar[3] to F_j^L in a small neighborhood of the corresponding location in the right image is referred to as F_j^R. The distance between F_j^L and F_j^R is computed as

$$d_j = 1 - F_j^L \star F_j^R, \tag{3.1}$$

[2] The reader is referred to Learned-Miller et al. [62] and Huang et al. [45] for approaches for image alignment.

[3] The similarity between two image regions is measured as the normalized cross-correlation between them.

where the symbol $\star$ denotes the normalized cross-correlation computation.

For each of these image regions, a set of simple features such as its location in the image, the intensity values, and the edge energies in different directions are computed. The monomials, of degree up to three, of these features are collected to create a pool of candidate hyper-features. This pool has a large number of hyper-features that are correlated with each other. From this pool, a few (20 in our experiments) of these hyper-features are selected using least angle regression [26]. The computed values of these selected hyper-features for the j^{th} image region is denoted by $\mathbf{h}_j$. Hereafter, unless a distinction between two different image regions is needed, we drop the subscript j for notational convenience.

Let us denote the two distributions for the expected distance between two image regions described in Section 3.2 as $P(d|C = 0, \mathbf{h})$ and $P(d|C = 1, \mathbf{h})$, where d is a continuous random variable and the distribution of $d|C$ is likely to be asymmetric around its mean. Based on these observations, the family of gamma distributions (shown in Figure 3.4) is selected to model these two conditional probability density functions as

$$P(d|C = 0, \mathbf{h}) \quad = \quad Gamma(d; \alpha_0(\mathbf{h}), \theta_0(\mathbf{h})), \tag{3.2}$$

$$P(d|C = 1, \mathbf{h}) \quad = \quad Gamma(d; \alpha_1(\mathbf{h}), \theta_1(\mathbf{h})), \tag{3.3}$$

where

$$Gamma(d; \alpha, \theta) = d^{k-1} \frac{e^{-d/\theta}}{\theta^{\alpha}\Gamma(\alpha)}. \tag{3.4}$$

Given the observed hyper-feature values $\mathbf{h}$, the parameters $\alpha_0, \alpha_1, \theta_0, \theta_1$ are obtained using a generalized linear model [71], the parameters of which are learned using the several examples of pairs of face images.

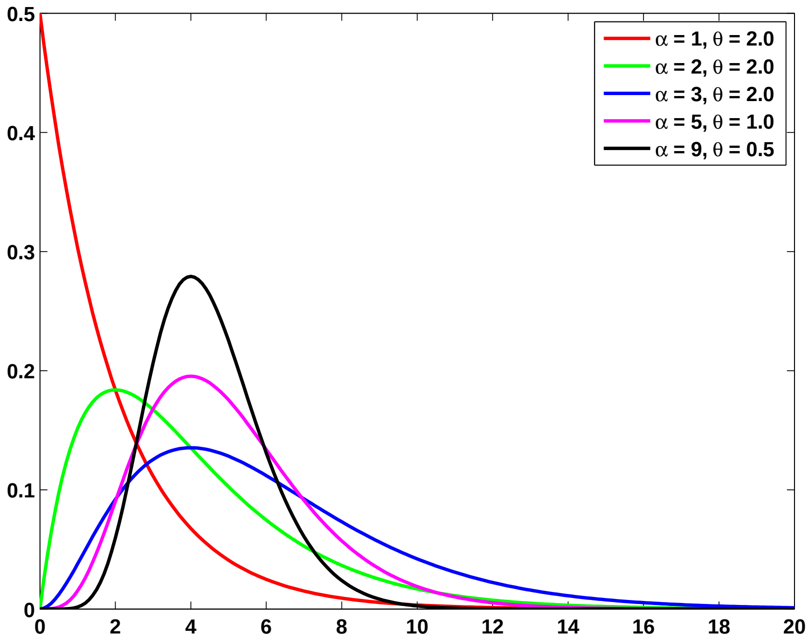


Figure 3.4. Probability density functions for different gamma distributions.

A gamma distribution has a shape parameter α and a scale parameter θ. Note that this family of unimodal distributions are asymmetric around their modes.

Using the notation described here, we now provide further details of the process of selecting image regions useful for the identification task.

3.2.2 Selection of useful image regions

Intuitively, if $P(d|C=0,\mathbf{h})$ and $P(d|C=1,\mathbf{h})$ are similar distributions, we do not expect much useful information about the value of the match-mismatch variable C from an observation of the value of the difference in appearances d. Mathematically, this intuition corresponds to the mutual information $I(d;C|\mathbf{h})$ between random variables C and d given the hyper-feature values $\mathbf{h}$, which is defined as

$$I(d;C|\mathbf{h}) \quad = \quad H(d|\mathbf{h}) - H(d|C,\mathbf{h}), \tag{3.5}$$

where $H(\cdot)$ is the Shannon entropy[4] and $P(d|\mathbf{h})$ is obtained by combining the estimates of $P(d|C=0,\mathbf{h})$ and $P(d|C=1,\mathbf{h})$.

All of the regions in an image are then sorted in a non-increasing order according to the estimate of their expected mutual information, and the top m regions are selected. We assume that the selections of image regions are independent of each other, which is not valid in general. However, for hyper-feature based systems, Ferencz et al. observed that modeling pairwise relationships between image regions does not improve the performance of these systems significantly.

We denote these m regions selected in the left image of the given pair of images as $\{F_1^L, \cdots, F_m^L\}$. We determine the corresponding regions in the right image using the procedure described in Section 3.2.1 and refer to them as $\{F_1^R, \cdots, F_m^R\}$. The distance d_j between the corresponding pairs of image regions F_j^L and F_j^R is computed using Equation 3.1. For ease of notation, we refer to a pair of corresponding image regions (F_j^L, F_j^R) together with the distance d_j between them as the *bi-patch* F_j. In the next section, we present two different models that use a collection of these bi-patches to predict the binary identification label for the given pair of images (I^L, I^R).

3.2.3 Prediction of identification label

The end goal of an identification task is to predict the value of the identification label C, where $C = 1$ implies that the given pair of images is a "same" pair and $C = 0$ implies that it is a "different" pair. In a probabilistic model, this prediction corresponds to determining if $P(C=1|I^L, I^R)$ is greater than $P(C=0|I^L, I^R)$. In other words, we predict that I^L and I^R are of the same object if

[4] For a discrete random variable X with n possible outcomes $\{x_i : i = 1 \cdots n\}$, the Shannon entropy is given by

$$H(X) \quad = \quad -\sum_{i=1}^{n} p(x_i) \log p(x_i), \tag{3.6}$$

where $p(x_i)$ is the probability mass function for the outcome x_i.

$$\frac{P(C=1|I^L, I^R)}{P(C=0|I^L, I^R)} \quad > \quad 1. \tag{3.7}$$

This criterion is also known as the maximum a posteriori (MAP) classification criterion.

In the previous section, we discussed the representation of an image-pair in terms of m bi-patches $F_1, ..., F_m$. Using this representation, we approximate the conditional probabilities as

$$P(C|I^L, I^R) \quad \approx \quad P(C|F_1, \cdots, F_m), \tag{3.8}$$

$$P(I^L, I^R|C) \quad \approx \quad P(F_1, \cdots, F_m|C). \tag{3.9}$$

In the next section, we describe Ferencz et al.'s generative model that optimizes the conditional likelihood of bi-patches given the identification label (Equation 3.9), which are then used to infer the identification label using Bayes' rule.

3.2.3.1 A generative model

Given an image-pair (I^L, I^R), Ferencz et al. [29] develop two separate models for estimating the probabilities $P(I^L, I^R|C = 1)$ and $P(I^L, I^R|C = 0)$ and employed Bayes' rule to compute the ratios of the posterior probabilities as

$$\frac{P(C=1|I^L, I^R)}{P(C=0|I^L, I^R)} \quad = \quad \frac{P(I^L, I^R|C=1)P(C=1)}{P(I^L, I^R|C=0)P(C=0)}. \tag{3.10}$$

Using the right hand side of the above equation, the probabilistic decision (Equation 3.7) is rewritten as

$$\frac{P(I^L, I^R|C=1)P(C=1)}{P(I^L, I^R|C=0)P(C=0)} \quad > \quad 1, \tag{3.11}$$

which by defining $\lambda = \frac{P(C=0)}{P(C=1)}$ becomes

$$\frac{P(I^L, I^R | C=1)}{P(I^L, I^R | C=0)} > \lambda. \tag{3.12}$$

Ferencz et al. further assume that all of the bi-patches in an image-pair are independent of each other when conditioned on the identification label C, i.e.,

$$P(F_1, ..., F_m | C) \;\; = \;\; \prod_{j=1}^{m} P(F_j | C) \tag{3.13}$$

Applying the above assumption and Equation 3.9 in Equation 3.12, they used the identification criterion

$$\prod_{j=1}^{m} \frac{P(F_j | C=1)}{P(F_j | C=0)} \;\; > \;\; \lambda. \tag{3.14}$$

Since the bi-patch F_j is completely specified by the hyper-feature values $\mathbf{h}_j$ and the distance d_j, we have

$$\begin{aligned} P(F_j | C) &= P(d_j, \mathbf{h}_j | C) && (3.15) \\ &= P(d_j | C, \mathbf{h}_j) P(\mathbf{h}_j | C) && (3.16) \\ &\propto P(d_j | C, \mathbf{h}_j), && (3.17) \end{aligned}$$

where Equation 3.17 is obtained by assuming the independence between $\mathbf{h}_j$ and C (which holds almost exactly in practice) and by assuming a uniform distribution for the hyper-feature values $\mathbf{h}_j$.

Using Equation 3.17, the final decision criterion for Ferencz et al.'s model is given by

$$\prod_{j=1}^{m} \frac{P(d_j | C=1, \mathbf{h}_j)}{P(d_j | C=0, \mathbf{h}_j)} \;\; > \;\; \lambda. \tag{3.18}$$

Since both $P(C)$ and $P(F|C)$ are completely specified in this model, it provides a generative process for the difference in appearance. Hence we refer to this model as a generative model.

The objective of developing the above model is to obtain better identification results, and *not* to design a generative process for sampling differences in appearance. While the above model was shown to achieve impressive results on identification of cars and faces, we argue that through a direct modeling of the desired objective, i.e., the distribution of the identification label, the performance of these hyper-feature based approaches could be further improved. In the next section, we present one such approach.

3.2.3.2 A discriminative model

Applying Equation 3.8 to Equation 3.7, the decision criterion becomes

$$\frac{P(C=1|F_1,\cdots,F_m)}{P(C=0|F_1,\cdots,F_m)} > 1. \tag{3.19}$$

Since the bi-patch F_j is completely specified by the hyper-feature values $\mathbf{h}_j$ and the distance d_j, the above equation is rewritten as

$$\frac{P(C=1|d_1,\cdots,d_m,\mathbf{h}_1,\cdots,\mathbf{h}_m)}{P(C=0|d_1,\cdots,d_m,\mathbf{h}_1,\cdots,\mathbf{h}_m)} > 1. \tag{3.20}$$

Since C is a binary random variable, we know that

$$P(C=1|d_{1:m},\mathbf{h}_{1:m}) + P(C=0|d_{1:m},\mathbf{h}_{1:m}) = 1. \tag{3.21}$$

Using this equality and changes of notation $d_{1:m}$ to represent $d_1,\cdots,d_m$ and $\mathbf{h}_{1:m}$ to denote $\mathbf{h}_1,\cdots,\mathbf{h}_m$, the decision criterion is given by

$$\frac{P(C=1|d_{1:m},\mathbf{h}_{1:m})}{1-P(C=1|d_{1:m},\mathbf{h}_{1:m})} > 1. \tag{3.22}$$

Next, we present a discriminative model that directly estimates the value of $P(C = 1|d_{1:m}, \mathbf{h}_{1:m})$ for a given image region. This estimated value is then used in the above decision criterion for an identification task.

To predict the probability values for a binary random variable, logistic regression is a commonly used approach. Here C is a binary random variable that depends on $(d_{1:m}, \mathbf{h}_{1:m})$. This dependency is captured by using the model

$$P(C = 1|d_{1:m}, \mathbf{h}_{1:m}) = \frac{1}{1 + e^{-X\beta}}, \tag{3.23}$$

where X is a vector representation computed using $(d_{1:m}, \mathbf{h}_{1:m})$ and β represents the parameters of the logistic regression model

$$\log\left(\frac{P(C = 1|d_{1:m}, \mathbf{h}_{1:m})}{1 - P(C = 1|d_{1:m}, \mathbf{h}_{1:m})}\right) = X\beta + \epsilon. \tag{3.24}$$

Here a binomial distribution is assumed for the noise term ϵ.

An alternate specification of a logistic curve $f(x)$ is given by two parameters: α_1 such that $f(\alpha_1) = 0.5$; and α_2 equal to the derivative of f at $x = \alpha_1$, i.e., $f'(\alpha_1) = \alpha_2$. Note that by defining $X^T = [1\ d_{1:m}\ \mathbf{h}_{1:m}]^T$, we have

$$X\beta = \beta_0 + d_{1:m}\beta_d + \mathbf{h}_{1:m}\beta_h. \tag{3.25}$$

It can be easily shown that

$$\begin{aligned} \alpha_1 &= -\frac{\beta_0 + \mathbf{h}\beta_h}{\beta_d}, & (3.26) \\ \alpha_2 &= \frac{\beta_d}{4}. & (3.27) \end{aligned}$$

Clearly, α_2 does not depend on $\mathbf{h}$. We argue that both of these parameters should be dependent on the hyper-feature values $\mathbf{h}$ to provide sufficient flexibility in the estimation of the above conditional distributions. To this end, we use the representation $X^T = [1\ d_{1:m}\ \mathbf{h}_{1:m}\ (d\mathbf{h})_{1:m}]^T$.

Figure 3.5 shows an illustration of the estimates of posterior probabilities obtained using our model. Note that for this illustration, we consider the identification task for a simpler object class namely cars. Also, the hyper-features used in this illustration are restricted to the monomials of degrees up to three of the y-position of the center of the image region in the image.

In the next section, we compare the performance of the above-mentioned generative and discriminative models for hyper-feature based face identification on a collection of images in unconstrained environments.

3.2.4 Face identification experiments

An approach similar to the above hyper-features based models is Moghaddam et al.'s Bayesian face identification system [76]. This approach also uses probabilistic models of differences of appearance d for a given image pair. In particular, they model $P(d|C = 0)$ and $P(d|C = 1)$ as Gaussian distributions. These two distributions are then used to predict the identification score as the maximum likelihood (ML) estimate $P(d|C = 1)$, or the maximum a posteriori (MAP) estimate $P(C = 1|d)$. We refer to these choices of scoring methods as *Bayesian ML* and *Bayesian MAP* approaches respectively. Note that the Bayesian MAP approach reported the best performance on the data set used in the FERET face recognition competition [84], which was sponsored by the U.S. Department of Defense. Here we include these two Bayesian face identification approaches as baseline approaches for the evaluation of the performance of the generative and discriminative models described in the previous sections.

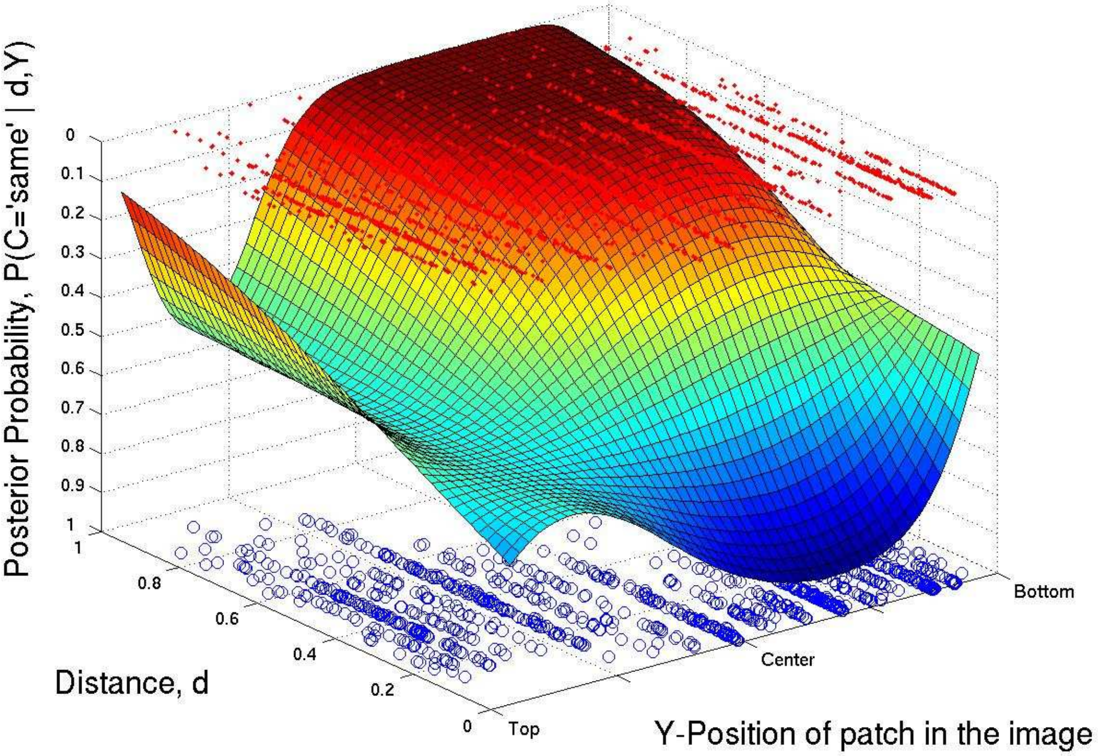


Figure 3.5. Logistic regression based upon a single hyper-feature, the y-position.

The blue circles in the lower plane and the red dots in the upper plane represent the pairs of training images for matched and mismatched cars respectively. Each point is plotted as a function of its match/mismatch label (C), the distance d between the image regions, and a hyper-feature y, the y-position of the left image region of the image region pair. Notice that the points for matching cars (lower plane) which are in the bottom half of the original images have their d values clustered around zero. This is because d values tend to be low for image regions near the bottom of the image when the cars match. On the other hand, for the same image position, the points representing mismatched cars have a more uniform distribution of d values. The goal of logistic regression is to approximate the original data points as well as possible while constraining each "slice" of the surface parallel to the d axis to be a logistic function. Furthermore, the parameters of the logistic curves at various y coordinates should be a smooth polynomial function of y. It is easy to see that the logistic surface "dips" to represent the low d values of the matching cars for image regions in a particular y range.

Figure 3.6. Example face images from the Faces in the Wild data set.

For our experiments, we randomly selected 1000 pairs each of "same" and "different" face images from Berg et al.'s *Faces in the Wild* data set.[5] Some example face images from this data set are shown in Figure 3.6. Note that a discussion of the characteristics of the images in this data set can also be found in the previous chapter (Section 2.2.2). One half of the selected images are held out as the test set, and remaining half is used for training different face identification models. Note that the test set contains faces of persons that are not in the training set.

Let S be the set of image-pairs in which both the images are known to be of the same person. Also, let T be the set of image-pairs for which the *predicted* label implies that both the images in each of these image-pairs are of the same person. We define

$$
\begin{aligned}
Precision &= \frac{|S \cap T|}{|T|}, & (3.28)\\
Recall &= \frac{|S \cap T|}{|S|}, & (3.29)
\end{aligned}
$$

where $|\cdot|$ denotes the cardinality of a set. Varying a threshold on the identification score obtained from a face identification approach, we generate different points on the precision vs. recall curves (shown in Figure 3.7).

As shown in Figure 3.7, the precision vs. recall for our discriminative model clearly dominates the curves for Ferencz et al.'s generative model and Moghaddam et al.'s

[5] http://www.tamaraberg.com/faceDataset/index.html

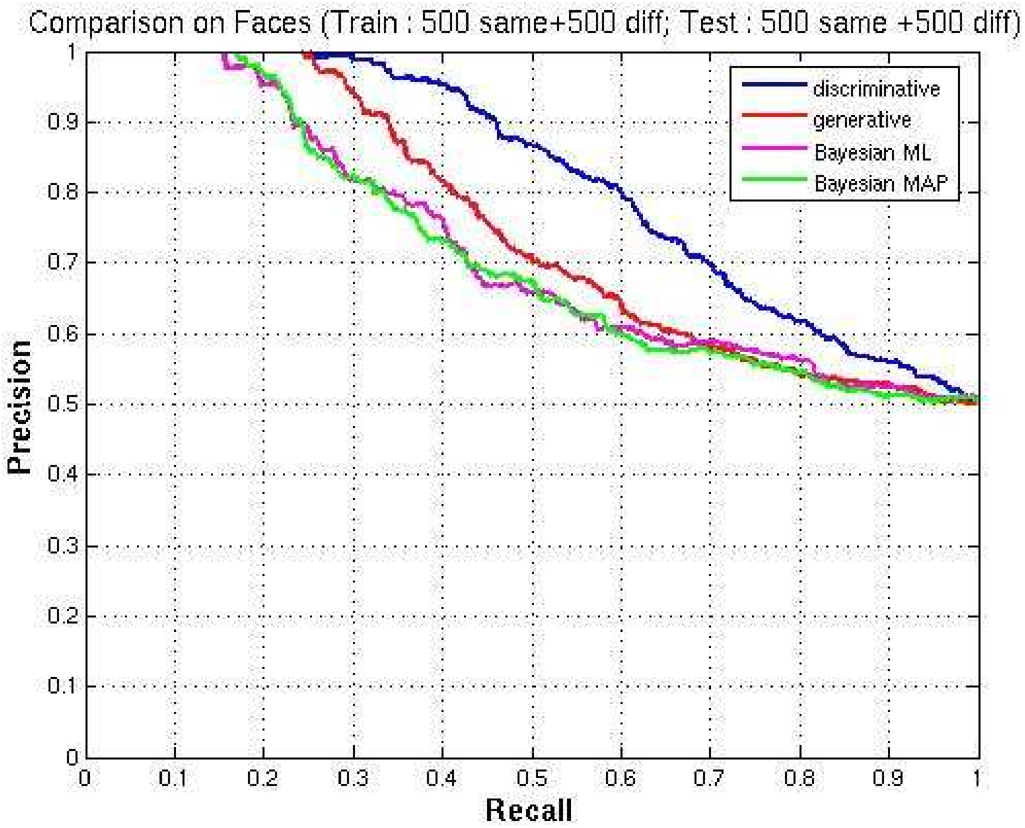


Figure 3.7. Precision vs. recall curves for our face identification experiments.

All of these models are trained using 500 image pairs each of "same" and "different" faces. The test set contains 500 pairs of "same" and "different" faces of persons not included in the training set. The blue curve for our discriminative model dominates the curves for all the other approaches, and the boost in performance is clearly evident over a wide range of recall values. Note that our results also outperform the Bayesian MAP approach [76] that was the best performer on the FERET data set.

	40%	60%	80%
Bayesian ML	74.6± 7.83	60.5±8.38	54.8 ± 2.91
Bayesian MAP	74.8± 9.09	59.9± 8.59	54.3 ± 6.15
Generative	81.2 ± 6.35	63.4± 6.71	54.4 ± 6.37
Discriminative	93.0 ± 6.29	**78.9 ± 8.15**	60.1 ± 6.97

Table 3.1. Precision at 40%, 60%, and 80% recall for different face identifiers.

These precision and recall values correspond to a 10-fold cross-validation on the held-out test set.

two approaches for Bayesian face identification. Table 3.1 shows the comparison of precision values at three different choices (40%, 60%, 80%) of recall for a 10-fold cross validation on the test set. Note that for all of these three settings, the precision values for our discriminative model are higher than the corresponding precision values for the generative model as well as the other two approaches, although the difference in performance is not always statistically significant. Some examples of pairs of face images that are correctly identified using our discriminative model are shown in Figure 3.8.

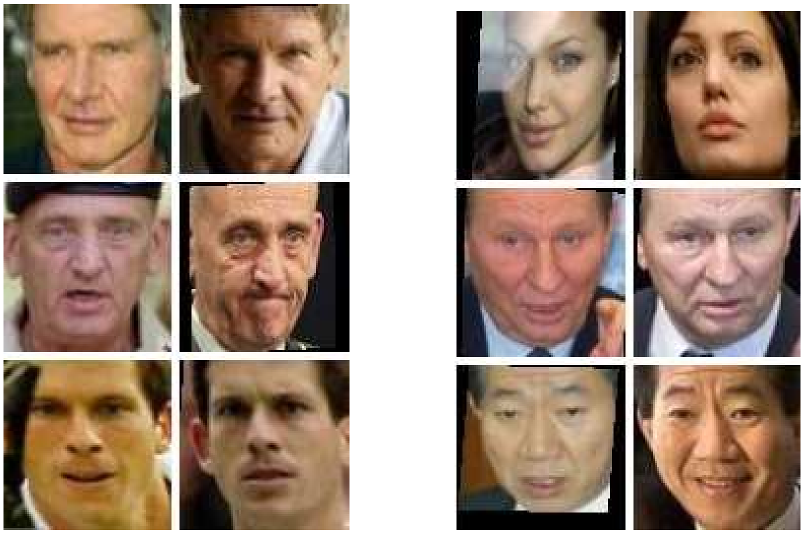


Figure 3.8. Example pairs of face images correctly labeled as "same" by our face identifier.

3.2.5 Discussion

In the previous sections, we introduced the notion of *hyper-features*, which are properties of an image region that can be used to estimate its utility for the identification task. Using these hyper-features, we presented two different models for the prediction of the identification label for a given pair of images. In our experiments, both of these models outperformed Moghaddam et al.'s Bayesian face recognition approach [76], which was the best-performing system on FERET face recognition competition [84]. The difference between the two models described in the previous section is that one of them is a generative model, and the other is a discriminative model.

Similar comparative studies of generative and discriminative learning have attracted significant interest in the machine learning community. One such comparison is due to Ng & Jordan [79]. In their theoretical analysis, they compared the empirical risk minimization for linear classifiers with naïve Bayes classifier. Their results suggest that even though the discriminative model (logistic regression) has a lower asymptotic error, the generative model (naïve Bayes) has a faster convergence towards the asymptotic error. Thus, with only a handful of training examples, naïve Bayes performs better than logistic regression, but with more training examples, the latter outperforms the former. More recently, Jain [46] performed an empirical comparison between naïve Bayes and logistic regression using large-scale experiments on data from the field of information retrieval. Also, Liang & Jordan [65] presented a unified framework for studying the comparison between generative and discriminative estimators, and concluded that when the model is well-specified, the asymptotic error for generative models is less than that of discriminative models, whereas when the model is mis-specified (i.e., the approximation error is not zero), discriminative models have lower approximation and asymptotic estimation errors.

Note that the above comparisons are helpful in understanding the behavior of these models in the limit of infinite training data. In practice, however, we have a limited number of training examples and the conclusions derived in the above comparisons often do not hold. Also, the choice between a generative model and a discriminative model depends on the requirements for the output of such a model. For instance, if the goal is to maximize the classification accuracy, a discriminative model may be the preferred choice, whereas a generative model may be preferred if the goal is to maximize the likelihood of the data under the learned model.

The joint model for image-caption pairs that we describe later in Section 3.4 uses a face identification system to estimate the likelihood of observing the difference in appearance between a given pair of images. Therefore, although the discriminative approach showed better performance than the generative approach in the face identification experiments shown in Section 3.2.4, the latter is a more suitable choice for our proposal for a contextual face recognizer.

Next, we describe a family of statistical models known as "topic models." A model from this family is used in Section 3.4 as our joint model for image-caption pairs.

3.3 Topic models

As described in Steyvers et al. [97], topic models are based on the idea that a text document is a mixture of topics, where a topic refers to a probability distribution of words. These models have recently emerged as powerful tools for the analysis of different types of data such as text documents [16, 32, 38], images [96, 99], and music key-profiles [43].

The topics obtained from these models are often broad and generic, associating large groups of people and issues that are loosely related. For instance, typical topics that emerge from a set of newspaper articles might represent broad areas such as "sports," "politics," or "the Middle East." Of course, as large numbers of topics are extracted from a set of documents on the same narrow subject, topics will become more and more narrow, and "politics" may split into "the White House," "Capitol Hill," and "the Justice Department," or some comparable set of more focused topics.

In many cases, it may be desirable to influence the direction in which these topics emerge. Here, we explore the idea of centering topics around people. In particular, given a large corpus of images featuring collections of people and associated captions, it seems natural to extract topics specifically focused on each person. What words are most associated with George Bush? Which with Condoleezza Rice? Since people play such an important role in life, it is natural to *anchor* one topic to each person. We use the term anchor to connote not only that a person should be a part of a topic, but that the topic should not drift too far from the topic defined by that person and their associations.

3.4 People-LDA

We present a new topic model *People-LDA*, which uses the coherence of face images in news captions to guide the development of topics. In particular, we show how topics can be refined to be more closely related to a single person (e.g., George Bush) rather than describing groups of people in a related area (e.g., politics). To do this our model tightly couples images and captions through a modern face identifier.

Our model produces word topics that are people-specific – it tends to eliminate secondary people or mixtures of people, focusing on a single person that matches a subset of face images. In addition, these people topics improve our ability to cluster faces over a method that uses only images. Thus, in addition to producing word topics that are people-specific (using images as a guiding force), our model is also used to cluster images by person, using the language model to boost performance.

People-LDA tightly couples images and captions through the face identifier described in the previous section. Note that this identifier provides a generative model of the *differences in appearance* of two face images as opposed to a generative model of the appearance of a face. Therefore, it is non-trivial to incorporate this face identification model into a topic model. A significant portion of our contribution represents the adjustment of a standard topic model to accommodate the modeling of differences in appearance rather than appearance.

The generative process for People-LDA is similar to the generative process for Blei et al.'s latent Dirichlet allocation model [16]. In the next section, we describe their model, and present the details of our model in the subsequent sections.

3.4.1 Latent Dirichlet allocation

The generative process for a document under the latent Dirichlet allocation (LDA) model is as follows. First, we sample a multinomial distribution $\boldsymbol{\theta}$ from a K-dimensional Dirichlet distribution with parameters $\boldsymbol{\alpha}$ given by

$$p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \quad = \quad \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{\prod_{i=1}^{K} \Gamma(\alpha_i)} \prod_{i=1}^{K} \theta_i^{\alpha_i - 1}. \tag{3.30}$$

Next, this sampled multinomial is used to generate N samples from K different topics according to the distribution

$$p(z_n = k|\boldsymbol{\theta}) \quad = \quad \theta_k. \tag{3.31}$$

Finally, for each of these topics z_n, we use the corresponding distribution $\boldsymbol{\beta}_{z_n}$ of V words in the vocabulary to sample a word w_n using the multinomial distribution

$$p(w_n = v|z_n, \boldsymbol{\beta}_{z_n}) \quad = \quad \beta_{z_n v}. \tag{3.32}$$

Algorithm 3 outlines the above generative process, and the corresponding graphical model representation is shown in Figure 3.9.

Algorithm 3 Generative process for LDA.

```
1: Choose a multinomial distribution $\boldsymbol{\theta}$ over $K$ topics from a Dirichlet distribution,
   i.e., $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is a Dirichlet prior.
2: for $n = 1$ to $N$ do
3:     Choose a topic $z_n$ from the chosen multinomial distribution in step 1. $z_n \sim$
       $Multinomial(\boldsymbol{\theta})$.
4:     Choose a word $w_n$ from a topic-specific distribution $\boldsymbol{\beta}_{z_n}$.
5: end for
```

Note that there are only two parameters in this model: the parameters of the above-mentioned Dirichlet distribution denoted by $\boldsymbol{\alpha}$; and K different topic-specific distribution of V words denoted by $\boldsymbol{\beta}_{1:K}$. Given these two model parameters, and a

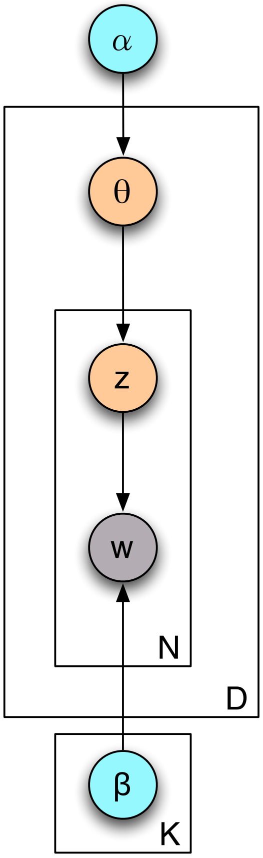


Figure 3.9. Graphical model representation of latent Dirichlet allocation. Here, $\boldsymbol{\theta}$ is a multinomial distribution, which specifies a mixture of topics $\mathbf{z}$. The entire collection contains D documents, each of which is represented as a collection of N words $\mathbf{w}$. The parameters for the model are $\boldsymbol{\alpha}$ and K distributions of words $\boldsymbol{\beta}_{1:K}$, one for each topic. In this graphical model, gray and orange nodes denote observed and latent random variables respectively, and cyan nodes denote the model parameters.

document $\mathbf{w}$, the joint distribution of a topic-mixture $\boldsymbol{\theta}$, topics $\mathbf{z}$, and the observed words $\mathbf{w}$ is given by

$$p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}_{1:K}) \quad = \quad p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{n=1}^{N} p(z_n|\boldsymbol{\theta}) p(w_n|z_n, \boldsymbol{\beta}_{1:K}). \tag{3.33}$$

Determining the distribution of topics in a given document corresponds to the computation of the posterior distribution of the document-specific latent variables θ and $\mathbf{z}$

$$p(\theta, \mathbf{z}|\mathbf{w}, \boldsymbol{\alpha}, \boldsymbol{\beta}_{1:K}) = \frac{p(\theta, \mathbf{z}, \mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}_{1:K})}{p(\mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}_{1:K})}. \quad (3.34)$$

For the exact computation of this posterior distribution, we need to compute the denominator term on the right hand side of the above equation, which is given by

$$p(\mathbf{w}|\boldsymbol{\alpha}, \boldsymbol{\beta}_{1:K}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \right) \left(\prod_{n=1}^{N} \sum_{i=1}^{k} \prod_{j=1}^{V} (\theta_i \beta_{ij})^{w_n^j} \right) d\boldsymbol{\theta}. \quad (3.35)$$

Due to the coupling between $\boldsymbol{\theta}$ and $\boldsymbol{\beta}_{1:K}$ in the summation term in the above equation, this function is intractable to compute [25]. To compute an approximation for the above posterior distribution, Blei et al. used a mean-field variational inference algorithm. Finally, they used a variational EM algorithm to estimate these model parameters for a given document collection. The details of the variational inference and variational EM algorithms can be found in Blei's PhD thesis [13].

In the next section, we present a non-trivial extension of this LDA model that employs a face identifier to obtain people-specific topics from a collection of image-caption pairs.

3.4.2 Generative process for People-LDA

The generative process for an image-caption pair under the People-LDA model is as follows. First, we sample a multinomial distribution $\boldsymbol{\theta}$ from a K-dimensional Dirichlet distribution with parameters $\boldsymbol{\alpha}$ (Equation 3.30). Next, this sampled multinomial is used to generate $N + M$ samples from K different topics according to the multinomial distribution in Equation 3.31. Then, for each of the first N topics z_n, we use the corresponding distribution $\boldsymbol{\beta}_{z_n}$ of V words in the vocabulary to sample a word w_n according to the multinomial distribution in Equation 3.32. Finally, for each of the next M topics z_{N+m}, we sample the difference of appearance according to the generative model described in Section 3.2.3.1. In particular, given the parameters $\boldsymbol{\lambda}$

of a generalized linear model, we obtain H samples of Gamma distributions $\mathbf{\Gamma}$ (Equation 3.4). Each of these Gamma distributions is then used to sample a difference in appearance d_{mh}. In other words, these H samples of the difference in appearance for faces in an image are collectively generated according to the distribution

$$p(\mathbf{d}_m|z_{N+m}, \boldsymbol{\lambda}) = \prod_{h=1}^{H} p(d_{mh}|z_{N+m}, \mathbf{\Gamma}_h) p(\mathbf{\Gamma}_h|\mathbf{I}, \boldsymbol{\lambda}). \tag{3.36}$$

Note that the People-LDA model does not provide a generative process for the appearance of face regions, but provides a generative process for the *difference* in appearance of face regions. The overall parameters for this model are $\boldsymbol{\alpha}$, $\boldsymbol{\beta}_{1:K}$, $\boldsymbol{\lambda}$ and a collection of K fixed reference images $\mathbf{I^M}$, one for each person.

Algorithm 4 outlines the above generative process, and the corresponding graphical model representation is shown in Figure 3.10.

Algorithm 4 Generative process for People-LDA.

```
1:  Choose a multinomial distribution $\boldsymbol{\theta}$ over $K$ people from a Dirichlet distribution,
    i.e. $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha}$ is a Dirichlet prior.
2:  for $n = 1$ to $N$ do
3:      Choose a person $z_n$ from the chosen multinomial distribution in step 1. $z_n \sim$
        $Multinomial(\boldsymbol{\theta})$.
4:      Choose a word $w_n$ from a person specific distribution $\boldsymbol{\beta}_{z_n}$.
5:  end for
6:  for $m = 1$ to $M$ do
7:      Choose a person $z_{N+m}$ from the chosen multinomial distribution in step 1.
        $z_{N+m} \sim Multinomial(\boldsymbol{\theta})$.
8:      for $h = 1$ to $H$ do
9:          Choose parameters $\mathbf{\Gamma}_h$ from a pre-trained generalized linear model with pa-
            rameter $\boldsymbol{\lambda}$.
10:         Choose a difference in appearance $d_{mh}$ from a person-specific hyper-feature
            based distribution, $p(d_{mh}|z_{N+m}, \mathbf{\Gamma}_h)$.
11:     end for
12: end for
```

Given the model parameters $\boldsymbol{\alpha}$, $\boldsymbol{\beta}_{1:K}$, $\boldsymbol{\lambda}$, and $\mathbf{I^M}$, and an observed image I, the joint distribution of a topic mixture $\boldsymbol{\theta}$, a set of $N+M$ topics $\mathbf{z}$, and an image-caption

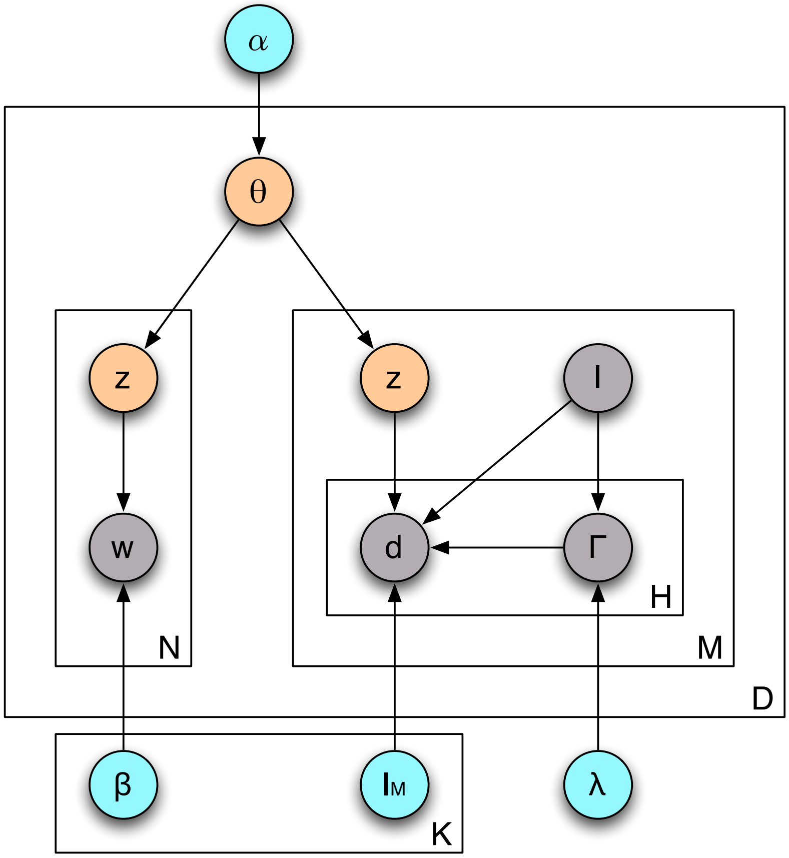


Figure 3.10. Graphical model representation of People-LDA.

Here, $\boldsymbol{\theta}$ is a multinomial distribution, which specifies a mixture of people-topics $\mathbf{z}$. The entire collection contains D image-caption pairs, each of which is represented as a collection of N words $\mathbf{w}$ and a collection of differences in appearance $\mathbf{d}$ between M faces appearing in the image I and the best matching image in a pre-determined set of reference images $\mathbf{I}^{\mathbf{M}}$. The variable $\boldsymbol{\Gamma}$ and parameter $\boldsymbol{\lambda}$ correspond to the hyperfeatures based face identifier described in Section 3.2.3.1. The parameters for this model are $\boldsymbol{\alpha}$, K distributions of words $\boldsymbol{\beta}_{1:K}$ and K reference images, one for each person, and the parameters $\boldsymbol{\lambda}$ of the face identifier. The generative process for this model is described in Algorithm 4. In this graphical model, gray and orange nodes denote observed and latent random variables respectively, and cyan nodes denote the model parameters.

pair with a set of N words $\mathbf{w}$ in the caption and image difference $\mathbf{d}$ between the M faces in the image and the matching reference images in $\mathbf{I}^{\mathbf{M}}$ is given by

$$
\begin{aligned}
p(\boldsymbol{\theta}, \mathbf{z}, \mathbf{w}, \mathbf{d} | \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \mathbf{I}^{\mathbf{M}}) \quad = \quad & p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \\
& \cdot \prod_{n=1}^{N} p(z_n|\boldsymbol{\theta}) p(w_n|z_n, \boldsymbol{\beta}) \\
& \cdot \prod_{m=1}^{M} p(z_{N+m}|\boldsymbol{\theta}) p(\mathbf{d}_m|z_{N+m}, \boldsymbol{\lambda}).
\end{aligned} \tag{3.37}
$$

3.4.3 Variational inference

Similar to the LDA model, the exact computation of the posterior distribution of the latent variables specific to a given image-caption pair is intractable in the above model as well. Following Blei et al.'s approach [16], we use a mean-field variational approximate inference algorithm to approximate this computation. In particular, we define a fully factorized model for the latent variables $\boldsymbol{\theta}$ and $\mathbf{z}$ as

$$
q(\boldsymbol{\theta}, \mathbf{z}|\boldsymbol{\gamma}, \boldsymbol{\phi}_{1:N}, \boldsymbol{\chi}_{1:M}) \quad = \quad q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} q(z_n|\boldsymbol{\phi}_n) \prod_{m=1}^{M} q(z_m|\boldsymbol{\chi}_m), \tag{3.38}
$$

where $\boldsymbol{\gamma}$ is a K-dimensional Dirichlet parameter, $\boldsymbol{\phi}_{1:N}$ and $\boldsymbol{\chi}_{1:M}$ are multinomial distributions. These parameters are also referred to as *variational* or *free* parameters, and this model is also referred to as a variational model. The graphical model representation of our variational model is shown in Figure 3.11.

We estimate the setting of these variational parameters by minimizing the Kullback-Leibler divergence between the approximate model and the true posterior distribution:

$$
(\boldsymbol{\gamma}^*, \boldsymbol{\phi}^*_{1:N}, \boldsymbol{\chi}^*_{1:M}) = \underset{(\boldsymbol{\gamma}, \boldsymbol{\phi}_{1:N}, \boldsymbol{\chi}_{1:M})}{\operatorname{argmin}} D(q(\boldsymbol{\theta}, \mathbf{z}|\boldsymbol{\gamma}, \boldsymbol{\phi}_{1:N}, \boldsymbol{\chi}_{1:M}) || p(\boldsymbol{\theta}, \mathbf{z}|\mathbf{w}, \mathbf{I}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, I_M)). \tag{3.39}
$$

This minimization is achieved using the iterative fixed-point method described in Blei's thesis [13], the update equations for which are obtained by computing the derivatives of the above Kullback-Leibler divergence with respect to the variational parameters and are as follows:

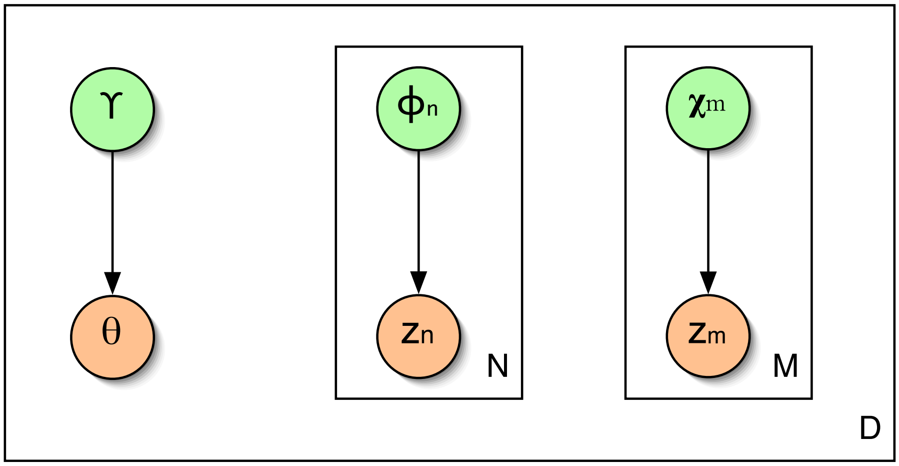


Figure 3.11. Variational model for People-LDA.

This model provides a fully factorized model for the latent variables $\boldsymbol{\theta}$ and $\mathbf{z}$ in terms of the variational parameters $\boldsymbol{\lambda}$, $\boldsymbol{\phi}_{1:N}$, and $\boldsymbol{\chi}_{1:M}$.

$$
\begin{aligned}
\phi_{ni}^{t+1} &= \boldsymbol{\beta}_{iw_n} \exp(\Psi(\boldsymbol{\gamma}_i^t)), && (3.40)\\
\chi_{mi}^{t+1} &= p(d_m | z_{N+m} = i, \mathbf{I}, \boldsymbol{\lambda}) \cdot \exp(\Psi(\boldsymbol{\gamma}_i^t)), && (3.41)\\
\boldsymbol{\gamma}_i^{t+1} &= \alpha_i^{t+1} + \sum_{n=1}^{N} \phi_n^{t+1} + \sum_{m=1}^{M} \chi_m^{t+1}, && (3.42)
\end{aligned}
$$

where $\Psi(.)$ is the digamma function.[6]

In the next section, we discuss how the parameters of the People-LDA model are learned from a collection of image-caption pairs.

3.4.4 Parameter estimation

The parameter estimation in our model becomes complicated due to our choice of the family of distributions to represent the image component. We simplify the estimation procedure by training the face identifier separately as discussed in Section 3.2.3.1. This approach of training some components of a probabilistic model

[6] The digamma function is the derivative of the log-gamma function, and is computable via a Taylor approximation [2].

separately is also known as *pre-training*. The parameters of this pre-trained face identifier are fixed while estimating the other parameters of the joint model.

The parameters of the joint model are learned by maximizing the likelihood of the given collection of image-caption pairs. In particular we used the variational expectation-maximization (EM) procedure [6] for this minimization. This procedure alternates between the following two steps until convergence: (E-step) estimate the variational parameters for each document given the current model parameters (Equation 3.39); and (M-step) compute the maximum likelihood estimates of the model parameters using the variational distribution with parameters estimated in the E-step. Note that the above M-step updates are the same as those of Blei et al. [16].

In this section, we have described the generative process and the procedures for performing inference of the posterior distribution and for estimating the model parameters for People-LDA. However, there are a few other small but critical details such as an automatic selection of reference images, that are necessary for applying our model for an unsupervised analysis of a collection of image-caption pairs. In the next section, we discuss these details and our experimental setup.

3.5 Experimental setup

In our experiments, we used 10000 images and associated captions from the *Faces in the Wild* data set [10]. This data set contains images, each having possibly more than one person appearing in it, and associated captions with one or two sentences of textual description of the scene shown in the image. Note that these image-caption pairs are typical of images appearing in news articles.

3.5.1 Unsupervised selection of reference images

People-LDA requires at least one reference image per people-topic. For an unsupervised selection of these reference images, we use an approach based on the idea that if only one face appears in a given image, and only one name is mentioned in the caption for this image, then this name is very likely to be that of the person whose face is shown in this image. To automatically detect faces in the given images, we used the Viola-Jones detector [108] for faces, and to extract names from the caption text, we employed a conditional random field-based named-entity recognizer [70]. The above-mentioned idea is then applied to the output of the face detector and the named-entity recognizer to obtain a selection of names and face images. For each name in this initial selection, we randomly chose one example face image as the corresponding reference image. Note that the named-entity recognizer is used only for selecting these reference images, and not for processing the caption in our model. Thus, our method is not particularly sensitive to the quality of the chosen named-entity recognizer.

Using this selection of names and faces, we obtained a set of 1077 distinct names (and one reference image for each of these names) in our data set. For our experiments, we randomly select 25 names in the middle frequency range of 20-80 occurrences in our data set. These names can intuitively be categorized as related to sports (e.g., Pete Sampras), politics (e.g., Jacques Chirac), and entertainment (e.g., Winona Ryder). The reference images for the selected names are shown in Figure 3.12.

3.5.2 Inferred distribution of topics for a document

For each of the given image-caption pairs, our model infers a distribution over all the possible people-topics. From the graphical model representation of People-LDA (Figure 3.10), it is not obvious that the inferred mixture $\boldsymbol{\theta}$ of people-topics will capture the co-occurrence relationship between the faces detected in an image and the names identified in the corresponding caption text. However, an appropriate choice of parameter values for the multinomial $\boldsymbol{\theta}$ (when most of the probability mass is on one value) do indeed force a correspondence between names and faces. Since we are

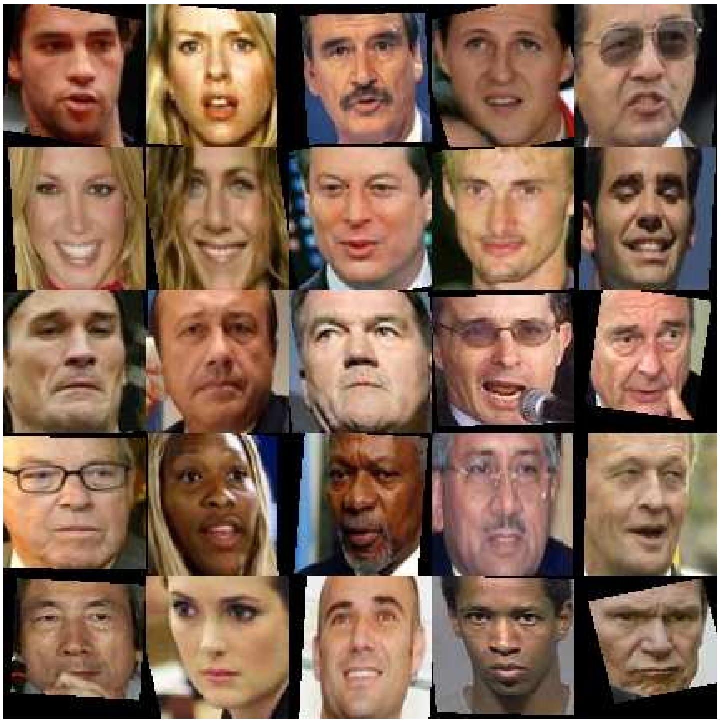


Figure 3.12. The set of reference images used in our experiments.

These reference images were selected using an unsupervised approach, the details of which are presented in Section 3.5.1.

learning the parameters of our model from the data itself, the inferred mixture $\boldsymbol{\theta}$ for a document takes the desired form.[7]

3.5.3 Annotating faces with names

For a given face region, our model makes an inference of the latent people-topic associated with it. The most likely name in the distribution of words associated with the inferred people-topic is then used as the annotation for this face region.

[7] A similar observation was made by Barnard et al. [5] in their model for annotating images with categorical words such as "sky," "grass," and "water."

The parameters of our model are selected to maximize the joint likelihood of the image-caption pairs and *not* to maximize the conditional likelihood of the names given the face regions. Therefore it is conceivable that the estimates of the conditional probability distributions obtained from our model may not be good, and can be better approximated using an explicit modeling of the desired correspondence between the names and faces (similar to the Correspondence-LDA model from Blei et al. [14]). In our model, the learned latent topics are anchored to a single person, i.e., in each of the learned topic-specific distributions of words, the probability of the most likely name is much higher than the probability of the second most likely name. Thus, a precise estimate of the probability values within a single topic is not critical in our model. For this reason, we do not need to specify an explicit correspondence between words and face images in our model.

3.5.4 "Unknown" class

Our model annotates a face in a given image with one of the names selected using the approach described in Section 3.5.1. As a result, it cannot identify names for people outside this set of selected names. Thus, we need to automatically identify the image-caption pairs for which the identities of some of the face regions in the image are likely to be from outside the set of selected identities. To address this issue, we use an additional identity "unknown" as annotation for faces of people whose reference images are not selected.

3.6 Related approaches

In this section, we discuss the face recognition and topic modeling approaches that are used in the comparative evaluation of our People-LDA model in the next section. Based on the input for these approaches, we categorized them as *Image-only*, *Text-only*, and *Image and text* approaches.

3.6.1 Image-only approaches

The following approaches only use the images (and not the captions) in our collection in their analysis.

3.6.1.1 Eigen-Fisher-faces

Zhao et al. [118] proposed a face recognition system based on a two-step process as follows. First, they project all of the training face regions into a low dimensional subspace using principal component analysis. Next, they learn a linear classifier using linear discriminant analysis [8].

Turk and Pentland [106] referred to the subspace obtained by principal component analysis of face images as *eigenfaces*, and Belhumeur et al. [8] referred to the subspace obtained by linear discriminant analysis of face images as *fisherfaces*. Since Zhao et al.'s approach is a combination of both of these dimensionality reduction techniques, we refer to Zhao et al.'s approach as *Eigen-Fisher-faces*. We implemented their approach to obtain a baseline for image-based face recognition methods in our experiments.

3.6.1.2 Hyper-features based face identifier

We trained the face identification system presented in Section 3.2.3.1 on a set of 500 "same" and 500 "different" pairs of images selected from the Faces in the Wild data set. This training set of image-pairs does not contain images of the 25 people (shown in Figure 3.12) used in our experiments. In Section 3.7, we will compare the performance of People-LDA and this approach to assess the boost in performance (if any) due to the use of a language model in addition to the image features.

3.6.2 Text-only approaches

The following approaches only analyze the captions in our collections.

3.6.2.1 Random name from the caption

This approach uses one of the names extracted from the caption (using a named-entity recognizer) and randomly assigns it to all the faces present in the associated image. We observed that this approach is particularly useful for the image-caption pairs with only one name detected in the caption. We implemented this approach to obtain a baseline performance for approaches that use the caption in the analysis of an image-caption pair. As we discuss later in Section 3.7, this naïve approach outperforms some image-based face recognition techniques, which suggests that the context from the caption is very useful for solving face recognition for difficult-to-recognize faces.

3.6.2.2 LDA on captions

We used Blei et al.'s LDA model (described in Section 3.4.1) for an unsupervised analysis of all of the captions in our collection. For each caption, we determine the most likely name under the inferred multinomial distribution of topics and the learned topic-specific word distributions. All of the face regions present in the associated image are annotated with this name.

In Section 3.7, we will compare the learned topic-specific distributions of words obtained using LDA with those obtained using People-LDA to verify if People-LDA is able to obtain people-specific topics.

3.6.3 Image and text

In this Section, we describe the details of some previous approaches that jointly analyze images and text.

3.6.3.1 Mixture of multimodal-LDA (MoM-LDA)

Barnard et al. [5] presented a joint model of the appearance of an image and a set of caption words associated with the given image. They referred to this model as the mixture of multimodal-LDA model. We implemented their model with two modifications. First, instead of representing an image as a collection of regions obtained by an image segmentation algorithm (e.g., normalized cuts algorithm [95]), we represent the image as a collection of face regions that are obtained from a face detection algorithm. Second, in addition to the visual features used by Barnard et al., we use SIFT descriptors [67] at several "interest point" locations[8] to represent a given image region. This approach provides a baseline for approaches that use both images and captions.

3.6.3.2 Correspondence-LDA

The MoM-LDA model provides good estimates of the joint probability of image-caption pairs, but does not provide good estimates of the conditional probabilities that are needed for annotations of different image regions with words. To address this issue, Blei et al. [14] presented an extension of MoM-LDA by including an explicit modeling of correspondences between words and image regions. They refer to this model as *Correspondence-LDA*. We include this model in our face recognition experiments as well.

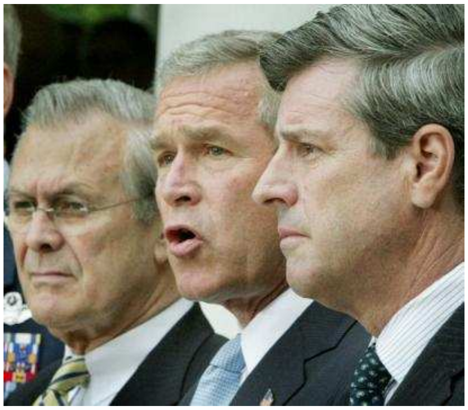
3.6.3.3 "Names and faces"

Berg et al. [10] presented an approach for clustering face images in a data set of image-caption pairs, focusing particularly on the names of people present in the caption. In particular, to obtain clusters of face images, they used a modified K-means algorithm in a joint space of image features and names extracted from the caption. While their method achieved impressive accuracy of annotating face regions with names, it has two main limitations:

[8] These locations are determined as the locations of the extrema of difference of Gaussian operator on the given image region.

1. It relies heavily on the performance of the named-entity recognizer. These programs can be brittle, and it is very difficult to recover from missed names. These programs also cannot recognize that terms such as “the first lady” and “Laura Bush” may refer to the same person. If the name of a person does not appear in the caption, then the person cannot be identified.

2. This method ignores important context and information provided by non-name text. Phrases like “Rose Garden” and “White House” (see Figure 3.13) can provide critical context to identify difficult-to-recognize faces, even if the name of the pictured individual is not shown.



President Bush, center, is flanked by the civilian U.S. administrator of Iraq L. Paul Bremer, right, and Secretary of Defense Donald H. Rumsfeld, left, as he makes remarks on Iraq, Wednesday, July 23, 2003, in the Rose Garden of the White House.

Figure 3.13. An image-caption pair from Berg et al.’s data set [10].

3.7 Experimental results

In this section, we first present a quantitative comparison of the above-mentioned approaches for annotating face regions in images. Next, we present a qualitative comparative analysis of the different clusters of face images and the different distributions of words obtained using some of these approaches.

3.7.1 Annotation of face regions

To quantitatively evaluate the annotations obtained from different approaches for face recognition, we manually labeled all of the face images in our collection with the corresponding names. Using this set of labeled faces, we use the following two measures to evaluate the performance of different approaches for face recognition:

- **Perplexity**

  The perplexity of a probabilistic model q for a data $X = \{x_1, \cdots, x_N\}$ represents how well this model predicts the data X and is given by

  $$Perplexity_q(X) = 2^{-\frac{1}{N}\sum_{i=1}^{N}\log_2 q(x_i)}. \tag{3.43}$$

  The perplexity of the true identities for all of the face regions in our collection is computed.

- **Average class accuracy**

  The probabilistic model is used to classify each of the face regions in the entire collection as one of the 25 selected persons and one "unknown" class, and the average class accuracy for these 26 classes is computed.

As shown in Table 3.2, a joint modeling of images and text outperformed all of the other approaches that model only one of the images or the captions. People-LDA achieved the lowest label perplexity (lower values are better) among all of the methods included in our experiments. People-LDA, and Barnard et al. [5] outperform the approach used by Berg et al. [10] since they model the probability distribution over all the possible names as compared to only the names detected in the caption only (as done by Berg et al.). On the other hand, the method used by Berg et al. had the best average class accuracies among the compared methods. Their method draws advantage from the fact that many captions have a single name present in them

Model	Perplexity	% accuracy
Image Only		
Eigen-Fisher-faces [118]	520.00 ± 24.17	22.02 ± 6.11
Hyper-features [47]	173.90 ± 3.96	44.86 ± 4.30
Text Only		
Random name from the caption	382.05 ± 23.11	31.40 ± 3.82
LDA on captions [16]	1219.60 ± 202.53	39.07 ± 2.44
Image and Text		
MoM-LDA [5]	68.23 ± 1.38	50.63 ± 4.01
Correspondence-LDA [14]	65.77 ± 2.13	52.50 ± 2.88
"Names and faces" [10]	73.05 ± 9.36	68.93 ± 4.69
People-LDA	25.99 ± 4.50	58.56 ± 3.59

Table 3.2. Quantitative evaluation of different face recognition approaches using label perplexity and average class accuracy measures.

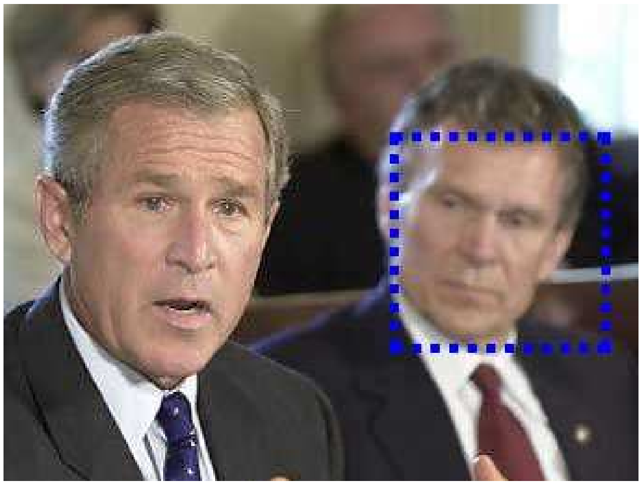
In the first column, we show the perplexity of the true label under different models (lower values are better). In the second column, the average class accuracies are shown. The error terms correspond to 10-fold cross-validation.

(similar to our naïve approach). Furthermore, their approach fails to annotate a face if the corresponding name is not present in the caption (for example, see Figure 3.14).

For a perfect labeling of all the faces in the data set, we still need to correct the misclassified faces. To do this, Berg et al. suggested the cost of correcting clustered data as a evaluation metric for different approaches. An alternative view of this cost is to consider only a few top matches and compute the recall (fraction of true labels present) of a system. In Figure 3.15, our proposed model outperforms the other approaches.

3.7.2 People-Topics

We presented People-LDA as a model that guides topics to automatically emerge around people. In this section, we demonstrate this by comparing the image clusters



President George W. Bush (L) speaks to reporters at the conclusion of a bipartisan congressional meeting, September 4, 2002 at the White House. Bush asked Congress for nearly $1 billion to aid Israel and the Palestinians, fight the spread of AIDS and bolster security at U.S. airports.

Figure 3.14. Difficulties in associating names in the caption with the faces appearing in the corresponding image.

A typical failure case for Berg et al. [10]: Since the name "Tom Daschle" is not present in the caption, Berg et al. do not consider it as a possible label for the detected face in the given image.

(Figures 3.16 and 3.17) that correspond to different people-topics and the topic-specific word distributions (Table 3.3) for different approaches. In particular, we compare the following three approaches:

1. *Image alone*: the model described in Section 3.6.1.2,

2. *Text alone*: the LDA model described in Section 3.6.2.2,

3. *People-LDA*.

3.8 Conclusions

We proposed People-LDA as a model that guides semantic topics to develop around people. We achieved this by combining two successful models: a hyper-feature based face identifier and the latent Dirichlet allocation model, in a novel way. To the best of our knowledge, our model is the first such combination for joint modeling of images and text. We show excellent results of generation of people-specific topics

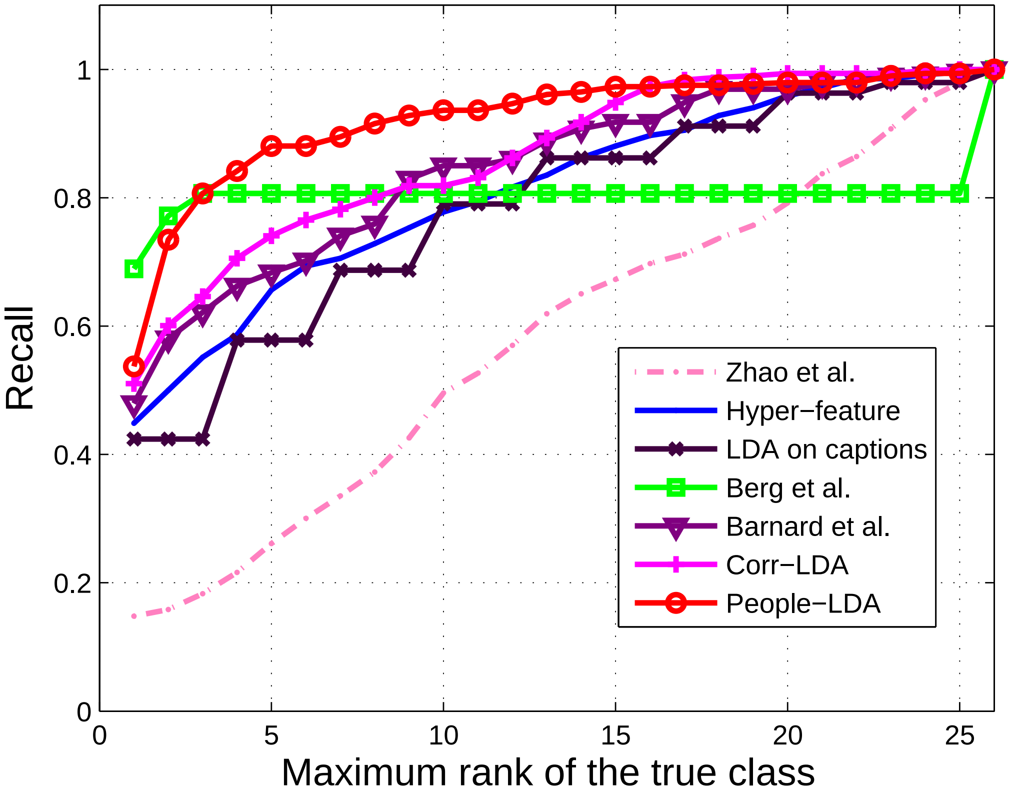
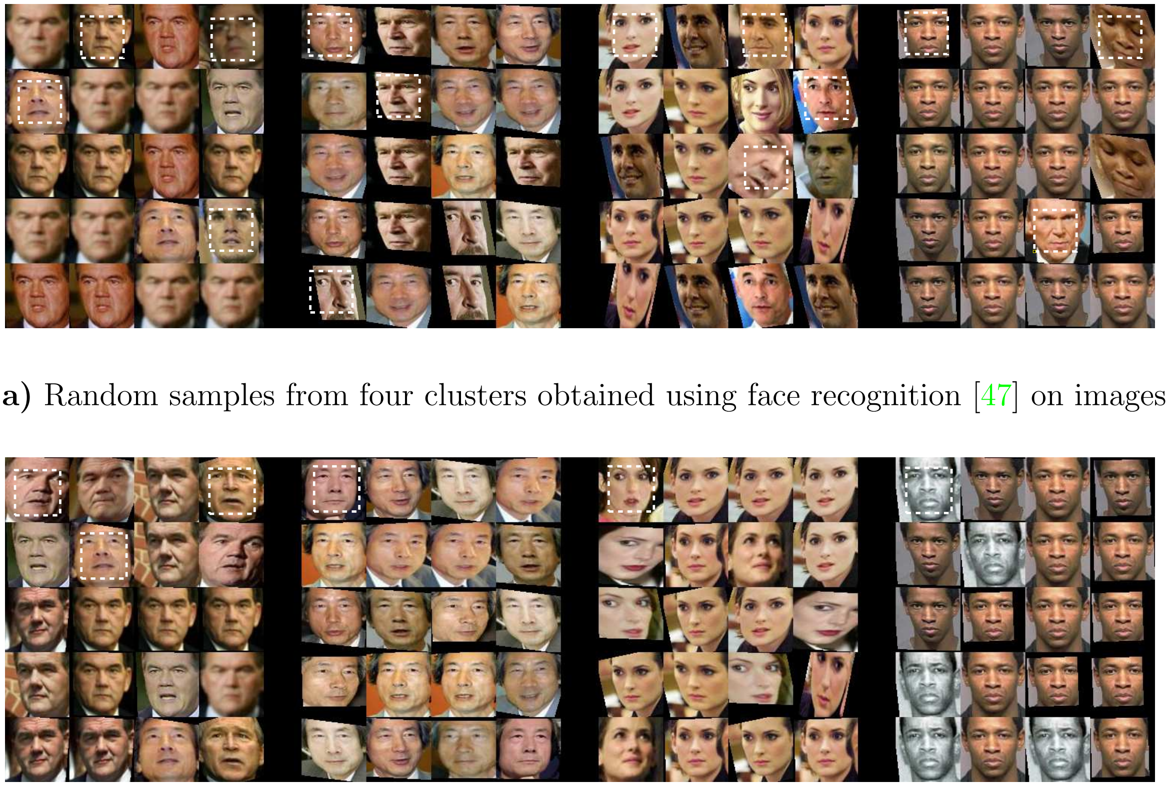


Figure 3.15. Comparison of different approaches for top-K recall.

People-LDA outperforms the other approaches over most of the range. The approach used by Berg et al. [10] shows promising recall up to rank three but levels out as it does not consider names not present in the caption (none of the captions in our data set had more than three distinct names detected in them).

from a data set containing images and associated captions. Our model outperformed different modern approaches in soft clustering of face images.

There are several issues with LDA that affect the performance of our proposed model. First is the assumption that topics are uncorrelated. This causes the clustering results to be sensitive to the number of topics chosen, particularly for a large number of latent topics. Several richer models [15] have been proposed to overcome this weakness. Another issue with LDA arises when we have a highly skewed distribution of cluster frequencies. This causes the very frequent terms to appear in multiple



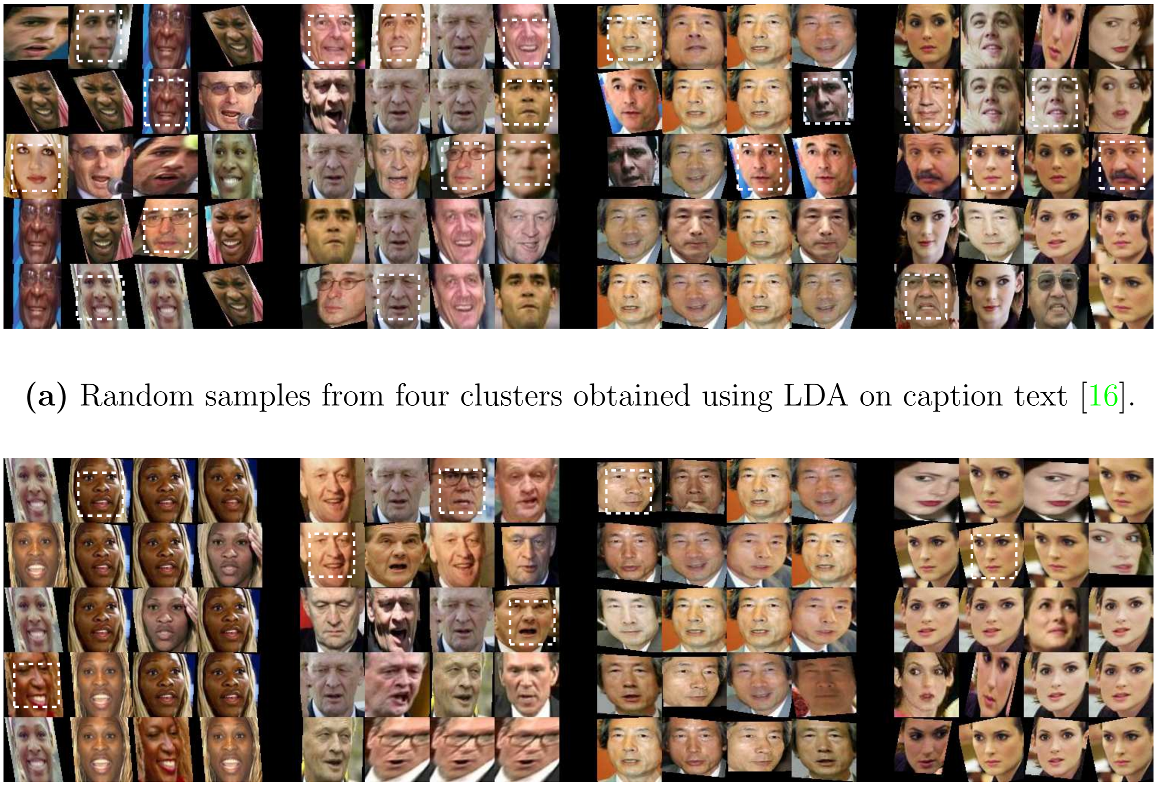
(a) Random samples from four clusters obtained using face recognition [47] on images.

(b) The corresponding clusters obtained by People-LDA.

Figure 3.16. Comparison of clusters obtained using only the images with those obtained using People-LDA.

White squares are drawn manually on top of some of the images to highlight the number of distinct people in a cluster. The clusters are cleaned up significantly using our model and have fewer different people in them.

clusters. To avoid this problem in our implementation, the most frequent terms (stop-words) in the captions were removed. Also, the frequencies of occurrence of the selected individuals are similar to each other. Recently, Elkan [27] proposed a topic model that addresses this issue of frequency skewness. Exploring such richer models for multi-modal documents would be an interesting extension to our work.



(a) Random samples from four clusters obtained using LDA on caption text [16].

(b) The corresponding clusters obtained by People-LDA.

Figure 3.17. Comparison of clusters obtained using only the captions with those obtained using People-LDA.

White squares are drawn manually on top of some of the images to highlight the number of distinct people in a cluster. The clusters are cleaned up significantly using our model and have fewer different people in them.

LDA

schumacher	**chretien**	versace	**williams**
chirac	**bush**	**chretien**	tennis
koizumi	**jean**	**spears**	cup
prix	street	poses	final
grand	cargo	**jean**	won
michael	michigan	**britney**	**uribe**
palace	facility	shows	returns
japan	suicide	women	development
jacques	fort	italian	tokyo
french	detroit	final	princess

People-LDA

schumacher	**chretien**	**spears**	**williams**
germany	**jean**	film	cup
cabinet	house	city	women
france	west	star	player
grand	ottawa	premiere	practice
jean	hill	poses	tennis
position	vote	**britney**	left
announced	action	**watts**	number
michael	question	mexico	montreal
driver	government	week	week

Table 3.3. Comparison of most likely words for people topics obtained by LDA and People-LDA models.

Each column corresponds to a topic learned by the model (LDA on caption text only or People-LDA). The name words are shown in bold face. These are four representative topics obtained using LDA. Topics obtained using People-LDA are more centered around one person compared to the topics for LDA. Moreover, the most likely name in a topic corresponds to the associated reference image.

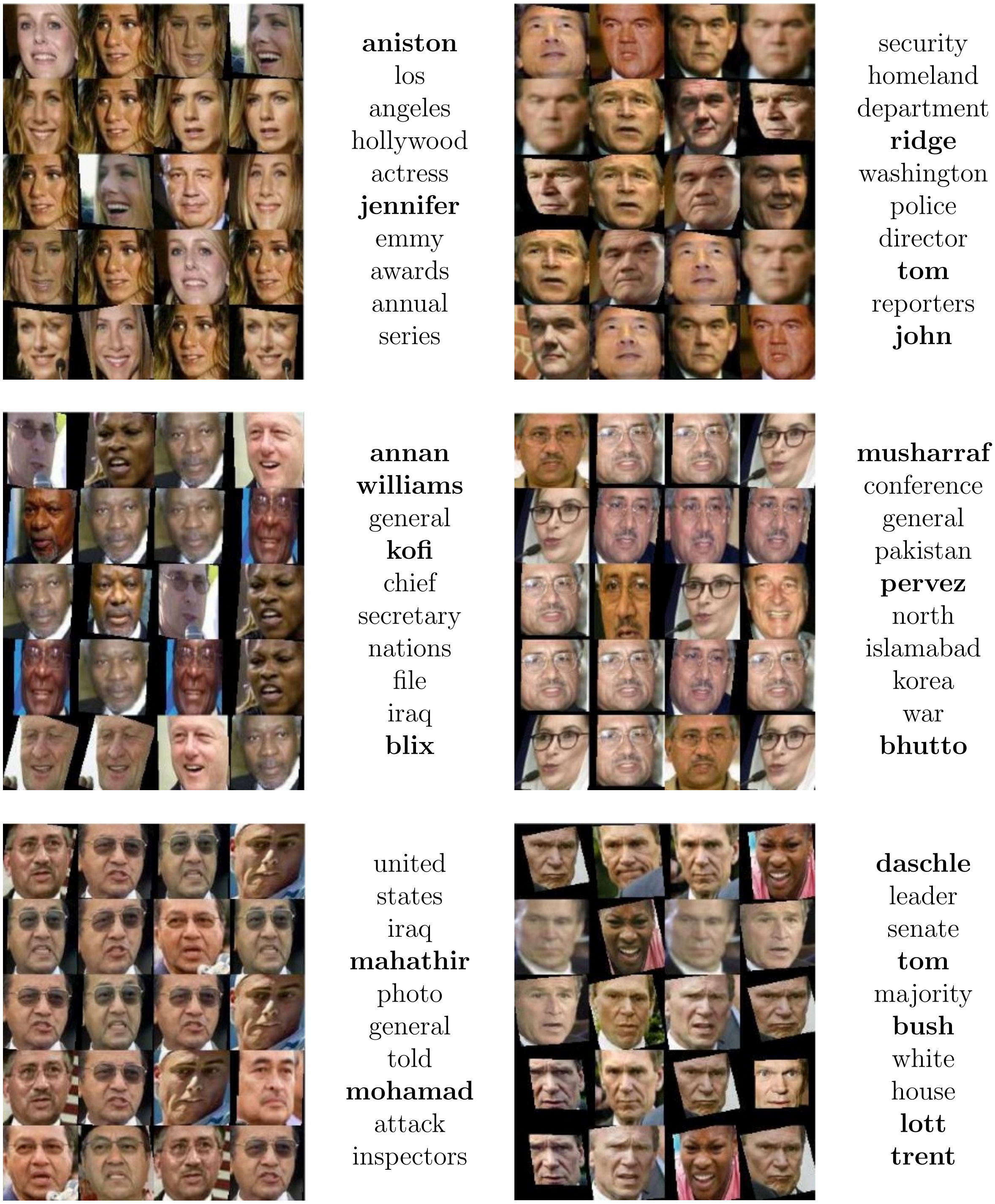


Figure 3.18. Additional examples of image and text clusters obtained using People-LDA.

CHAPTER 4

SCENE CLASSIFICATION AS CONTEXT

4.1 Introduction

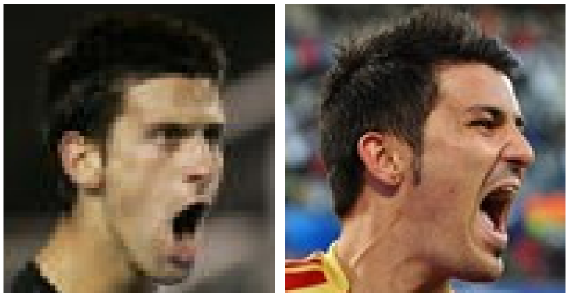


Figure 4.1. Are these two face images of the same person?

Consider the face images shown in Figure 4.1. These two face images are of two different individuals, but they appear very similar to each other in the given pose and emotion (of celebration). An automated system would find it extremely difficult to discriminate between these two persons from these face images alone. In the previous chapter, we presented an approach that reduces the complexity of such difficult cases of face identification (or recognition) by using the context generated from the captions associated with these images. However, that approach is useful only when the captions are available for the given images. In this chapter, we discuss the scenario where the captions for the images are not available, but other useful information can be obtained from other image regions.

For instance, the two face images shown in Figure 4.1 are cropped from the two images shown in Figure 4.2 respectively. It is clear that these two images are of two different sports: the first image is related to tennis, whereas the second image is related to soccer. Based on this knowledge of the sporting event shown in each of these images, it is easier to conclude that these face images are of two different persons. (The highlighted face in the first image is of Serbian tennis player Novak Djokovic, and the highlighted face in the second image is of Spanish soccer player David Villa.)

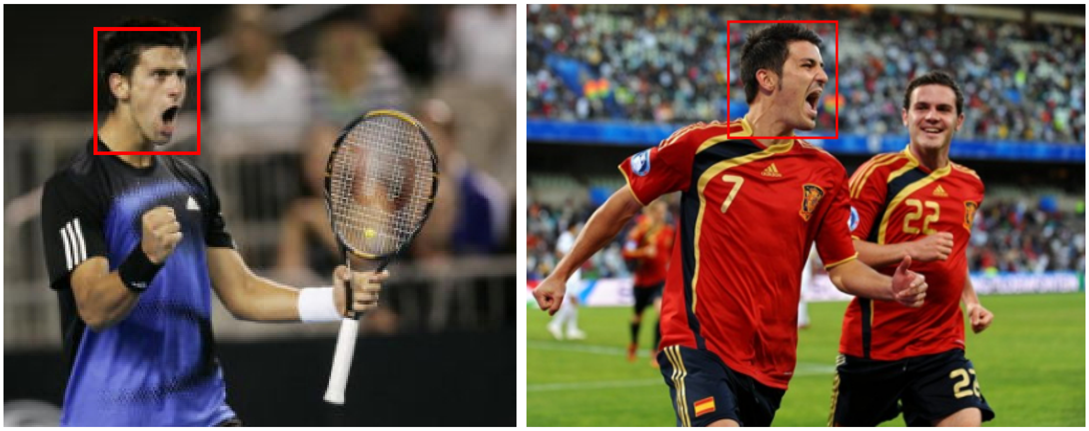


Figure 4.2. Are the two highlighted faces in these two images of the same person?

Most of the existing approaches for face identification [29, 47, 50, 76] ignore any information from the non-face regions in images. As a result, they are solving an extremely difficult problem (as illustrated in Figure 4.1) that may not be necessary to solve in several scenarios, e.g., the scenario shown in Figure 4.2. A face identifier that characterizes the entire image (including both face and non-face regions) to identify such scenarios could avoid solving these difficult identification tasks. In this chapter, we present an approach that characterizes a given image by classifying the scene shown in the image into a set of pre-determined classes. In particular, we study the problem of identifying the sporting event shown in an image, assuming that we know the image is related to sports.

Classifying the scene shown in an image may also be useful in making further inferences about the scene. For instance, consider the image shown in Figure 4.3. Identifying the sporting event (tennis) in this image provides context for recognizing the event (French Open), the venue (Roland Garros stadium), and the player (Rafael Nadal). Some of these annotations (also shown in the caption of Figure 4.3) are difficult to generate using the appearance of the corresponding image regions alone. In particular, for the current resolution of the face region (shown in Figure 4.4), there is no hope that an automated face recognizer (or even a human) will identify the player in this image. However, given the context that this image captures a game of tennis played on a clay court in the year 2007, the identity of the player is more likely to be Rafael Nadal than Björn Borg or Tiger Woods.

In this chapter, we study a collection of sport images taken by amateur photographers and present an approach for automatically recognizing the sport. First, in Section 4.2, the problem of sport classification is specified. Then, in Section 4.3, the intuition behind our solution to this problem is presented. Next, in Section 4.4, some random-fields based approaches are briefly described. This description is used in Section 4.5 to present a new model called *selective hidden random fields*. Next, the details of our experimental setup are included in Sections 4.6, 4.7, and 4.8, and the experiments are presented in Section 4.9. Finally, in Section 4.10 we conclude this chapter with a discussion of other domains where our framework is likely to improve the existing solutions.

4.2 Sport classification

The problem of scene classification has been studied mostly for indoor vs. outdoor [92] scenes, and natural scenes such as mountains, waterfalls, and open fields [68]. For the classes of scenes included in these studies, simple algorithms that employ the statistics of basic image features such as the distribution of color and low-level texture

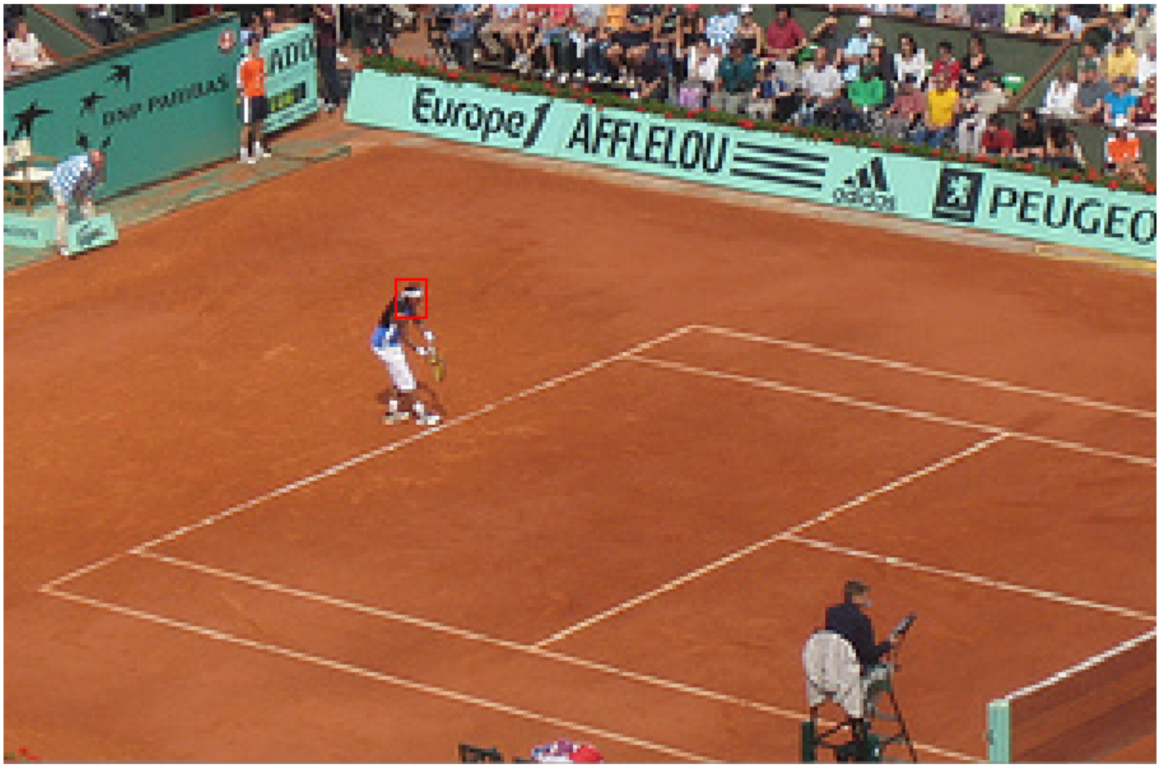


Figure 4.3. Example showing the role of context in event classification.

Most people may identify the sporting event in this image as “tennis” and associate tags like “French Open” and “clay court” with this image. A careful observer may also add “Rafael Nadal” and “serving” to the annotation.

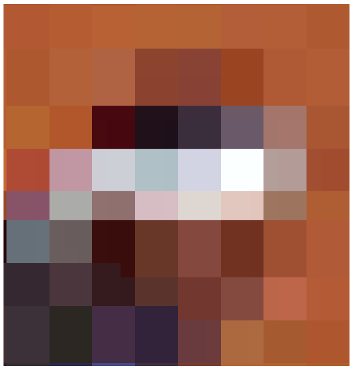


Figure 4.4. An extremely difficult-to-recognize face.

This image shows the face region that is highlighted in the image shown in Figure 4.3.

features reported impressive performance. However, these approaches perform poorly in the domains where the different classes of scenes show high inter-class similarities and high intra-class variations in the appearance of scenes.

Sport classification is one such challenging domain. On one hand, both the soccer field and the football field are green and appear very similar to each other. On the other hand, the tennis courts could be of different colors such as red (clay courts) and green (grass courts). Thus, an approach based on the statistics of basic image features is not likely to perform well for sport classification.

In this chapter, we study the sport classification problem with the classes being five popular sports: baseball, basketball, (American) football, soccer, and tennis. To study this problem in a natural and unconstrained setting, we consider a collection of sports images taken by amateur photographers.[1] Most of the images in this collection were taken by the spectators who are at a considerable distance from the playing surface (e.g., see Figure 4.5).

In most of these images, large image regions are occupied by the people watching the sporting event. These image regions, however, provide little information for classifying the sporting event. Worse yet, these image regions can potentially be distractive to algorithms that analyze the general scene statistics to classify the sporting event shown in an image. Ignoring these regions (of spectators) in the images in our collection, several observations can be made:

- the markings on the playing surface for a single sport are consistent across different venues and over time;

- the markings on the playing surface are different for different sports, e.g., a set of parallel lines in a (American) football field for yard markings as opposed to a diamond in a baseball field.

[1] These images were downloaded from http://www.flickr.com and described later in Section 4.8.

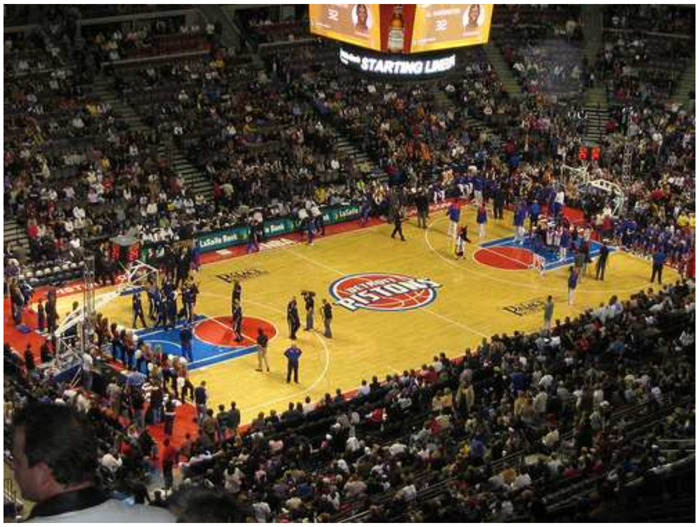


Figure 4.5. Example image from our collection of sports images.

Most of the images in our collection are taken by the people sitting in the spectator area, and often have a wide-angle view of the sporting event with the crowd covering large regions of the image. While the resolution of such images is often too low to recognize sport accessories such as a ball or a racquet, the playing surface stands out as a reliable and robust source of information to identify the sporting event.

Based on these observations, one approach for sport classification would be to identify and characterize the markings on the playing surface in a given sport image. To characterize these markings, we need to identify the image regions that correspond to the playing surface. However, segmenting the playing surface in an image may depend on the sporting event itself. Therefore, there may be a circular dependency between the classification of the sporting event and the segmentation of the playing surface.

In this chapter, we solve both of these problems simultaneously through a novel probabilistic model that jointly segments the regions of interest in an image and selects the features computed on them to predict the classification label for the given image. We refer to our model as *selective hidden random fields*. We start our discussion by providing a brief overview of the idea that led to the development of this model.

4.3 Overview of our solution

Our solution for the problem of sport classification works as follows. First, a given image is partitioned into several image regions with little variation in the appearance within each of these image regions (see Figure 4.6). These image regions are later classified as being a part of the playing surface or not. Each of these image regions are represented using several appearance-based image features (discussed later in Section 4.6). Next, the markings in the image are characterized using several easy-to-identify, long lines (not edges) in the given image (see Figure 4.7). Finally, a probabilistic model (described later in Section 4.5) is used to simultaneously select the image regions that are part of the playing surface, and use the line markings on these selected image regions to classify the sporting event shown in the given image.

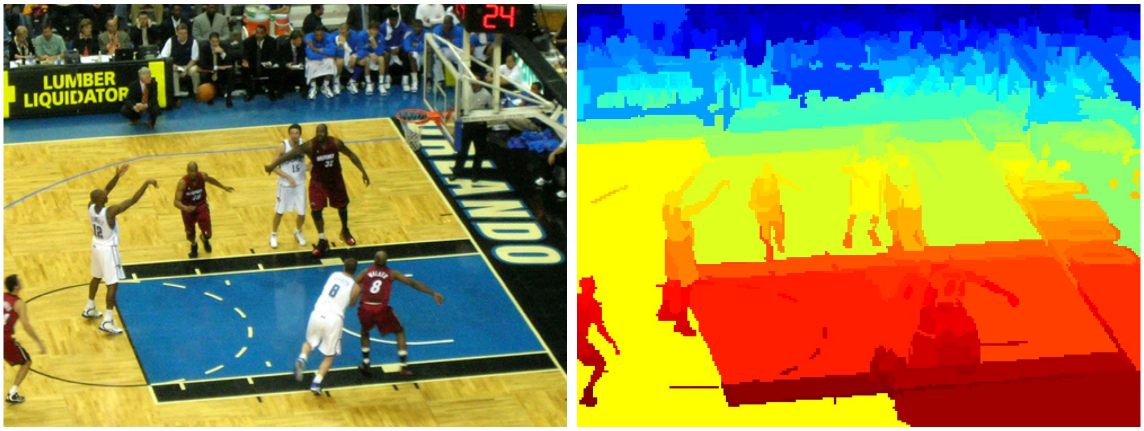


Figure 4.6. An example of the segmentation of an image.

We partition a given image into regions with consistent appearance using Comaniciu and Meer's segmentation algorithm [22]. The output of their segmentation algorithm on the left image in shown in the right image, where each color represents a different image region.

For the segmentation of the playing surface in an image, it is desirable that:

1. The segmented image regions should correspond to the actual playing surface irrespective of the sharp changes in appearances within the playing surface. For example, the painted area in a basketball court (blue area in the left image of

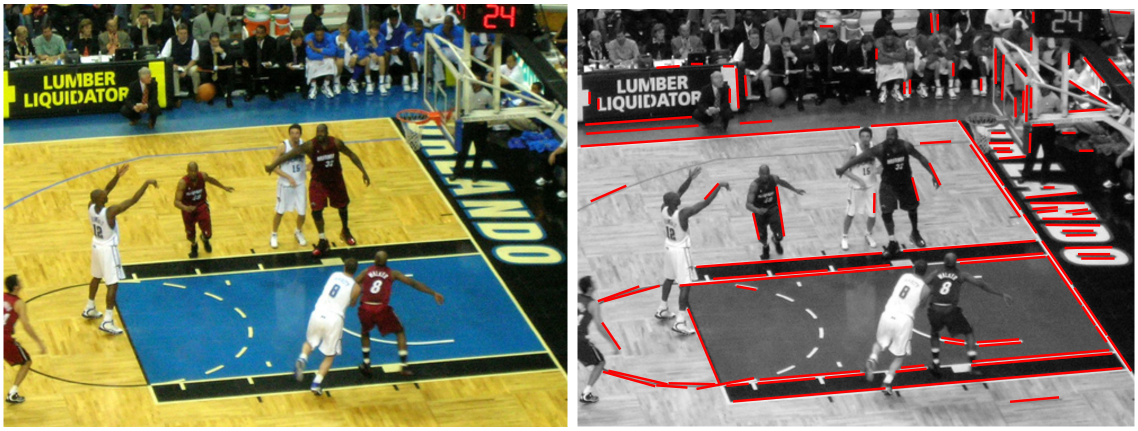


Figure 4.7. An example of a set of line features computed for an image.

We identify several lines (not edges) in a given image using Kosecka et al.'s approach. The line hypotheses in the left image are shown in red in the right image.

Figure 4.7) has a different appearance than the rest of the court, but is still a part of the playing surface.

2. The segmentation labels for image regions are mostly consistent with their neighboring image regions. In other words, if an image region is a part of the playing surface, then its neighbors are also likely to be a part of the playing surface.

To address the first issue, we employ a separate classifier for identifying the playing surface across different sports. This classifier is trained using a set of images where all the image regions that are part of the playing surface are marked. Obtaining such labels is a very tedious task. Therefore, we have these labels only for a small subset of images in our collection. Note that due to the large variations in the appearance of the playing surfaces across different sports, we do not expect this classifier to work very well. Nevertheless, we expect these predictions to be useful as features in our model. We refer to the predictions from this classifier for different image regions as the *base hypotheses*.

To address the second issue, our model penalizes the labeling for two neighboring image regions that are inconsistent with the similarity measure between them. In other words, if two neighboring regions have similar appearances, then our model would prefer the labeling that assigns the same label for these image regions (and vice versa for the image regions with dissimilar appearances). We refer to this preference for consistent labeling as the *neighborhood compatibility.*

One probabilistic model that employs features similar to the above base hypothesis and neighborhood compatibility features is *conditional random fields* [60]. In the next section, we present a brief overview of this model and describe variants of this model that are suitable for solving the problem of sport classification.

4.4 Related approaches

4.4.1 Conditional random fields

Let $\mathbf{X}$ denote the observations for the regions in an image and $\mathbf{y}$ represent the binary random variables that specifies if these image regions are part of the playing surface or not. Also let G denote the dependency graph used to model the above-mentioned properties for the desired segmentation of a given image. Now, we define a joint probability distribution $p(\mathbf{X}, \mathbf{y})$ that is factorized over the cliques C of the graph G as

$$p(\mathbf{X}, \mathbf{y}) = \prod_{c \in C} \phi_c(\mathbf{X}_c, \mathbf{y}_c). \tag{4.1}$$

Note that this model specifies a Markov random field with respect to the graph G. The functions $\phi_c(\cdot)$ are often referred to as *factors* of the graph G. In this model, the probability of the labels conditioned on the observations is computed as

$$p(\mathbf{y}|\mathbf{X}) \;\; = \;\; \frac{p(\mathbf{X}, \mathbf{y})}{\sum_{\mathbf{y}} p(\mathbf{X}, \mathbf{y})}. \tag{4.2}$$

In the above equation, the denominator is computed by enumerating over the set of all possible observations $\mathbf{y}$, which is exponential in $|\mathbf{y}|$ and may not be reasonably enumerable in general. To address this issue, Lafferty et al. [60] proposed conditional random fields that directly model the conditional probability distribution factorized over the cliques C of a graph G as

$$p(\mathbf{y}|\mathbf{X}) \quad = \quad \frac{1}{Z(\mathbf{X})} \prod_{c \in C} \phi_c(\mathbf{X}_c, \mathbf{y}_c), \tag{4.3}$$

where $\mathbf{X}_c$ and $\mathbf{y}_c$ respectively represent the subsets of $\mathbf{X}$ and $\mathbf{y}$ that are in the clique C. The term $Z(\mathbf{X})$ is also known as the *partition function* and is given by

$$Z(\mathbf{X}) \quad = \quad \sum_{\mathbf{y}} \prod_{c \in C} \phi_c(\mathbf{X}_c, \mathbf{y}_c), \tag{4.4}$$

When each of the factors $\phi_c(\cdot)$ is from the exponential family,[2] the conditional distribution is given by

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \quad = \quad \frac{1}{Z(\mathbf{X}, \boldsymbol{\theta})} \exp\left(\sum_{c \in C} \sum_{k=1}^{n_c} \theta_{ck} f_{ck}(\mathbf{X}_c, \mathbf{y}_c) \right), \tag{4.5}$$

where $\boldsymbol{\theta} = \{\theta_{ck}\}$. Note that this model has a large number ($\sum_{c \in C} n_c$) of parameters that need to be learned from the training data. In practice, the number of these parameters is reduced by sharing them among different sets of factors. These sets of factors that share parameters are also known an *factor templates.*

The CRF model has been successfully used in a variety of domains, including text processing [60], bioinformatics [89], and computer vision [34, 59]. One limitation with these models is that they do not include latent variables to capture the intermediate structure in the data. Quattoni et al. [86] addressed this issue by extending the

[2]

CRF models to include latent variables. They refer to their models as *hidden-state conditional random fields* (HCRF).

4.4.2 Hidden-state conditional random fields

Denoting the latent variables as $\mathbf{h}$, the HCRF model defines the conditional probability distribution

$$p(\mathbf{y}, \mathbf{h}|\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{Z(\mathbf{X}, \boldsymbol{\theta})} \exp \left(\sum_{c \in C} \sum_{k=1}^{n_c} \theta_{ck} f_{ck}(\mathbf{X}_c, \mathbf{y}_c, \mathbf{h}_c) \right), \tag{4.6}$$

where $\mathbf{X}_c$, $\mathbf{y}_c$, and $\mathbf{h}_c$ respectively represent the subsets of $\mathbf{X}$, $\mathbf{y}$, and $\mathbf{h}$ that are in the clique C, and $\boldsymbol{\theta} = \{\theta_{ck}\}$. The conditional probability of the label given the observations is computed as

$$p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) = \sum_{\mathbf{h}} p(\mathbf{y}, \mathbf{h}|\mathbf{X}; \boldsymbol{\theta}). \tag{4.7}$$

The parameters $\boldsymbol{\theta}$ of this model were estimated by maximizing the regularized data log-likelihood

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_i \log P(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\theta}) - \frac{1}{2\sigma^2} \|\boldsymbol{\theta}\|^2. \tag{4.8}$$

A gradient ascent algorithm is employed to perform this maximization. Note that computing the gradient of the log-likelihood term requires the inference of the marginal distributions for $P(\mathbf{h}_c|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})$. In the HCRF model, the size of the cliques including the hidden variables were restricted to be at most two, i.e., $\mathbf{h}_c$ can include up to two hidden variables. Thus, for the gradient computation in this model, we need to infer $P(h_i|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})$ and $P(h_i, h_j|\mathbf{y}, \mathbf{X}, \boldsymbol{\theta})$. Quattoni et al. further reduced the graph over the hidden variables to a tree structure, which enabled them to apply belief propagation to make inference about these probability distributions. A similar approach is used to perform the inference

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmax}}\ p(\mathbf{y}|\mathbf{X}), \tag{4.9}$$

which is required for computing the most likely labels for an unseen data instance.

Several variants of the HCRF model have been applied to different applications such as recognizing human gestures [110] and learning discriminative object parts [54]. There are two main limitations of these models:

1. The learned hidden layer in these models are not necessarily interpretable. The model is optimized to predict the final label $\mathbf{y}$. Hence, unless explicitly specified, these models are not guaranteed to generate any semantic interpretations to the inferred hidden layer.

2. In these models, the target variables $\mathbf{y}$ are connected only to the hidden layers $\mathbf{h}$. In other words, there are no edges between $\mathbf{y}$ and $\mathbf{X}$, which implies that the information flow from the observations to the classification label happens through the hidden layer. Since the hidden layer is composed of discrete random variables, these models are useful when the structure in the inferred hidden layer provides sufficient information to generate the final labels $\mathbf{y}$. Note that replacing these discrete random variables with continuous random variables or variables with a large number of possible states would make the required inference tasks intractable.

In the next section, we present an extension of the HCRF model that addresses both of these issues. In particular, our model

1. enforces semantics on the hidden layer through the use of a pre-trained classifier (related to the desired semantics) and a careful choice of the family of functions for specifying the factors of different types of random variables.

2. includes edges between the observed and unobserved layers, thereby enabling a richer information flow from the data to the labels.

In our model, the hidden layer is used to *select* some of the observations to be used for predicting the sport label for the given image. Due to this selective nature of the processing obtained from our model, we refer to this model as *selective hidden random fields* (SHRF). Next, we present the details of our model for sport classification.

4.5 Selective hidden random fields

A selective hidden random field (SHRF) is an extension of the hidden-state conditional random field model. The SHRF model employs binary random variables as latent variables to select some of the observations to be used for predicting a label for the observed data. In the context of sport classification, we represent the i^{th} region in a given image as $\mathbf{x}_i$ and denote by h_i the corresponding latent variable that specifies if this image region is a part of the playing surface. Also, the sport label for the given image is denoted by $\mathbf{y}$. Figure 4.8 shows the factor graph representation of this model.

Given a set of observations $\mathbf{X}$ and the parameters $\boldsymbol{\theta}$, the conditional probability of the class (sport) label $\mathbf{y}$ is given by

$$p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}) = \sum_{\mathbf{h}} p(\mathbf{y},\mathbf{h}|\mathbf{X},\boldsymbol{\theta}) \tag{4.10}$$

$$= \frac{\sum_{\mathbf{h}} \exp(\Phi(\mathbf{y},\mathbf{h},\mathbf{X},\boldsymbol{\theta}))}{\sum_{\mathbf{y}} \sum_{\mathbf{h}} \exp(\Phi(\mathbf{y},\mathbf{h},\mathbf{X},\boldsymbol{\theta}))}, \tag{4.11}$$

where

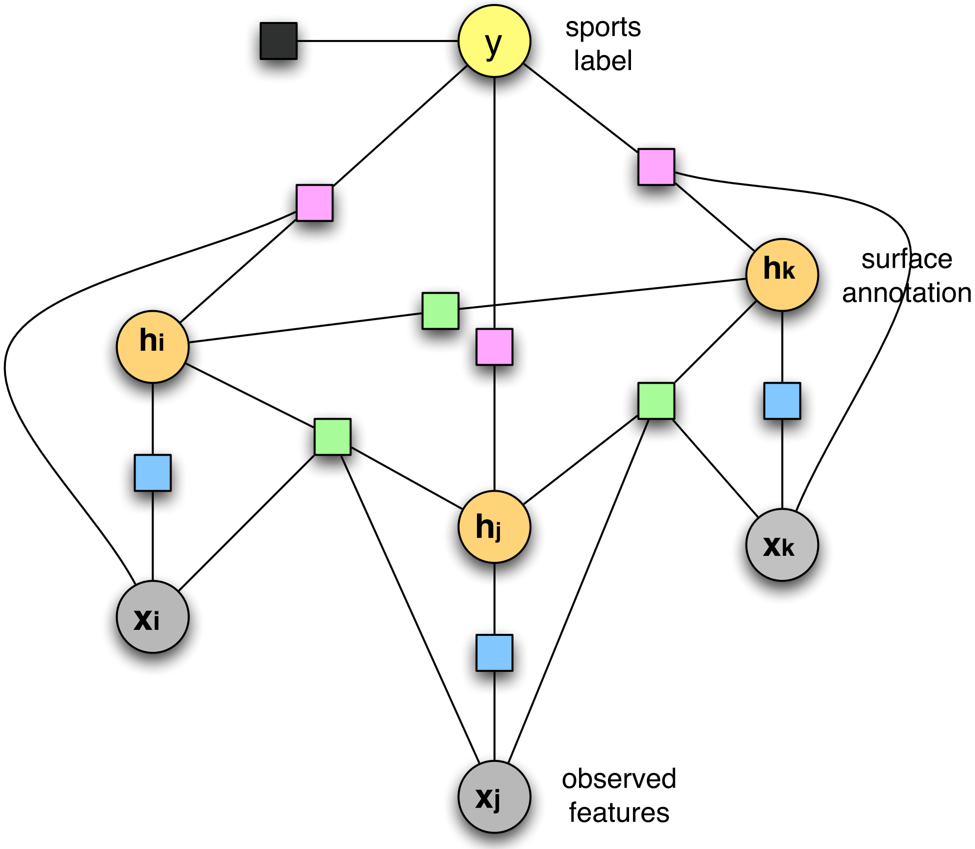


Figure 4.8. Factor graph representation of a selective hidden random field.

In this graphical model, the node y represents the sport label, and h_i and x_i represent the surface annotation and observed features for the i^{th} image region, respectively. We used colored squares to specify different factor templates, i.e., all of the factors of the same color share the same parameter values. A blue factor represents the local evidence for the surface annotation of an image region, a green factor denotes compatibility between the annotations for connected image regions, a purple factor represents the contribution of an image region towards the sport label for the image, and the black factor represents the prior probabilities for different sporting events. Note that some of the edges (e.g., the edge connecting the green factor between h_i and h_k with x_i) are omitted for clarity.

$$\begin{aligned}
\Phi(\mathbf{y}, \mathbf{h}, \mathbf{X}, \boldsymbol{\theta}) &= \sum_i \phi^b(h_i, \mathbf{x}_i, \boldsymbol{\theta}_b) && (4.12)\\
&+ \sum_{i,j \in E} \phi^g(h_i, h_j, \mathbf{x}_i, \mathbf{x}_j, \boldsymbol{\theta}_g) && (4.13)\\
&+ \sum_i \phi^p(\mathbf{y}, h_i, \mathbf{x}_i, \boldsymbol{\theta}_p) && (4.14)\\
&+ \phi^k(\mathbf{y}). && (4.15)
\end{aligned}$$

In this equation, $\boldsymbol{\theta}^T = [\boldsymbol{\theta}_b^T \;\; \boldsymbol{\theta}_g^T \;\; vec(\boldsymbol{\theta}_p)^T]$ and the factors ϕ are described as follows:

- *Base hypothesis* (**blue box**). For each of the regions $\mathbf{x}_i$ in the given image, we use a pre-trained classifier to estimate the probability that this region is a part of the playing surface. We defer the discussion of the details of this classifier to Section 4.7. Denoting the estimated probability by $f_1(\mathbf{x}_i)$, we compute

$$\phi^b(h_i, \mathbf{x}_i; \boldsymbol{\theta}_b) = \boldsymbol{\theta}_b^T [\delta_{h_i=1} f_1(\mathbf{x}_i) \;\; \delta_{h_i=0}(1 - f_1(\mathbf{x}_i))]^T. \tag{4.16}$$

- *Neighborhood compatibility* (**green box**). For every pair of neighboring[3] image regions, we compute a value $f_2(\mathbf{x}_i, \mathbf{x}_j)$ that represents the similarity in appearance between these two image regions. In particular, we use the cosine similarity between the two feature-vector representations of these two image regions. (The details of our representation are presented in Section 4.6). This computed similarity value $f_2(\mathbf{x}_i, \mathbf{x}_j)$ is used to compute

$$\phi^g(h_i, h_j, \mathbf{x}_i, \mathbf{x}_j; \boldsymbol{\theta}_g) \;\; = \;\; \boldsymbol{\theta}_g^T [\delta_{h_i=h_j} f_2(\mathbf{x}_i, \mathbf{x}_j) \;\; \delta_{h_i \neq h_j}(1 - f_2(\mathbf{x}_i, \mathbf{x}_j))]^T. \tag{4.17}$$

- *Selection of image regions* (**purple box**). This factor is used to select the image regions that are labeled as part of the playing surface. The selected image regions are represented as $f_3(\mathbf{x}_i)$ in the factor

$$\phi^p(\mathbf{y}, h_i, \mathbf{x}_i) = \mathbf{y}^T \boldsymbol{\theta}_p^T [\delta_{h_i=1} f_3(\mathbf{x}_i)]^T. \tag{4.18}$$

- *Prior probability for different sport labels* (**black box**). This factor specifies the prior information about the frequencies of the labels of different sporting

[3] We apply different heuristics, such as sharing of boundary and threshold on similarity in appearance, to reduce the connectivity in graph, as opposed to a fully connected graph. This is done to ensure that the approximate inference algorithm converges.

events in a given data set. For example, if the owner of a given collection is passionate about soccer and tennis, but would rarely go to a baseball game, then the prior probabilities could be appropriately set to specify that soccer and tennis images are more likely to occur than baseball images in this collection. In our experiments, we assume a uniform (non-informative) prior, i.e.,

$$\phi^k(y) = 1. \tag{4.19}$$

Similar to the HCRF model discussed in the previous section, the parameters of our model $\boldsymbol{\theta}$ are estimated by maximizing the regularized data log-likelihood. A conjugate-gradient method is used for this optimization, where the computation of the gradient of the log-likelihood involves the evaluation of the marginal probabilities of the hidden variables and of the pairs of hidden variables. Unlike the HCRF model, the presence of cycles in the connectivity graph prevents the use of exact methods for inference of these quantities. In our experiments, we use the loopy belief propagation algorithm [77] for doing approximate inference in this graph. Finally, given the parameters $\boldsymbol{\theta}$ and observed image $\mathbf{X}$, we apply similar approximate inference techniques for computing the sport label for this image

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmax}}\; p(\mathbf{y}|\mathbf{X}, \boldsymbol{\theta}) \tag{4.20}$$

as specified in Equation 4.11.

In the next section, we present the details of the representation of image regions. The classifier used for computing the base hypotheses is described in Section 4.7.

4.6 Representation of an image region

Given an image, we use Comaniciu and Meer's mean-shift algorithm [22] to partition the image into several image regions. Each of these image regions are then represented using a set of features (shown in Table 4.1) to describe its location, shape, appearance, and geometry.

Type	Features
Location and shape	
Position	mean x-position normalized by the width of the image. mean y-position normalized by the width of the image.
Shape	number of pixels (i.e., area) in the image region. second moment of the region.
Appearance	
Color	mean of the red-green-blue (RGB) channel values. mean of the hue-saturation-value (HSV) channel values.
Texture	mean responses to eight difference-of-oriented-Gaussian filters.
“Interest points”	vector-quantized scale-invariant feature transform (SIFT) descriptors are computed at the locations of interest points detected by the maximally stable extrema regions (MSER) detector, the extrema of the difference of Gaussians (DoG) detector, and the affine-invariant Harris corner detector.
Geometry	
Single line	histogram of lines in nine different orientations.
Pair of lines	number of line intersections length of the pairs of parallel lines

Table 4.1. The set of features used to represent an image region.

The details of the computation of the interest points features and the geometric features are presented in Sections 4.6.1 and 4.6.2.

4.6.1 Interest points

Interest points are used to represent an image region in terms of the appearance at a few sample locations in this region. An interest point (or region) detector is used to determine the locations of several informative samples in the given image region. The appearances of the image at these sampled locations are represented using a feature descriptor computed at these locations.

In a detailed comparison of scale- and affine-invariant interest point detectors [74], Mikolajczyk and Schmid found the extrema of difference of Gaussian (DoG) operator [67] and the maximally stable extrema regions (MSER) [69] detector to be useful for a variety of visual scene categories. Another useful interest point detector is the affine-invariant Harris corner detector, which responds to relatively local yet salient interest points in an image region. In our experiments, the union of the output of these three interest point detectors is used.

To represent the image appearance at each of these interest point locations, we follow Mikolajczyk and Schmid's recommendation [73] and use the scale-invariant feature transform (commonly known as SIFT) [67] descriptor. Note that the SIFT descriptor requires the scale at which the interest point is detected to determine the scale of the detected interest point. The DoG detector provides this information but the MSER and affine-invariant Harris corner detectors do not. To address this issue for the MSER detector, the scale of the detected interest point is estimated using the dimensions of the ellipse (estimated using the method of moments) around the output region. A similar procedure was followed for estimating the scale for the affine-invariant Harris corner detector.

To further reduce the dimensionality of the representation of an image region, the computed SIFT descriptors are clustered into several bins (or "visual words") using the *k-means* algorithm. All of the resulting bins are collectively referred to as a “visual vocabulary,” and the resulting representation for the image region is called the “bag of visual words” representation.

4.6.2 Geometric features

In the previous sections, we discussed that the sport shown in an image can be easily identified using an effective characterization of the playing surface shown in the given image. In particular, we argued the case for the markings on the playing surface as a useful characterization of the playing surface in a sports-related image. Automatically determining these markings on the playing surface is difficult particularly for images with low resolution, occlusions, and extreme perspective view. To determine these markings, our approach is as follows. First, we identify the straight lines in the given image, and then use the interactions (e.g., intersections and parallelism) between pairs of lines to represent the markings. In particular, we use Kosecka et al.’s approach [57] to determine long line segments in an image and compute a histogram of orientations weighted by the length of the line segments. To compensate for the difference in the viewing angle across images, this histogram is appropriately adjusted such that the most frequent orientations are aligned. The average orientation histograms for the lines detected on the entire image and the playing surface are shown in Figure 4.9. It is important to note that these lines were estimated once for the entire image. To compute the line features for a particular image region, we use the overlap between all of the estimated lines and this image region.

The representation described in this section computed for the i^{th} region in an image is used as the representation of the random variables $\mathbf{x}_i$ in the SHRF model (described in Section 4.5). The SHRF model also uses a classifier for generating a base hypothesis for each of the image regions that denotes if this image region is part of the playing surface. In the next section, we present the details of this classifier.

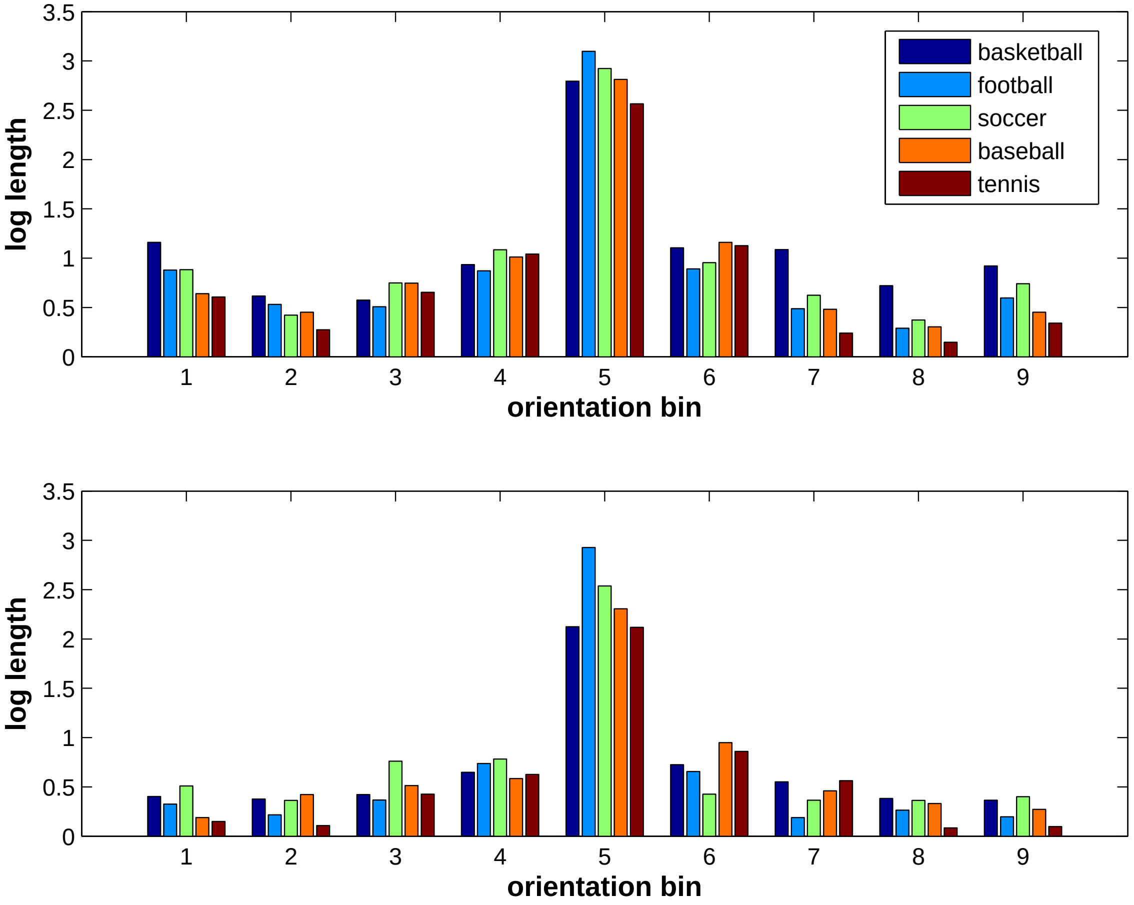


Figure 4.9. Histogram of orientation of lines detected in an image.

Average distribution of the cumulative length of lines detected in the entire image (top) and on the playing surface (bottom). The lines are clustered by their orientation, and the resulting histogram is rotated to center around the bin with maximum cumulative length. The distribution at the bottom is more useful for discrimination among the classes than the distribution on the top.

4.7 A simple classifier for identifying the playing surface

One observation about the sports included in our collection is that the playing surface in each of these sports is horizontal. (Our collection does not include sports such as downhill skiing and golf, where this observation is not true.) Thus it is conceivable that we can generate the base hypothesis for image regions by predicting their surface orientation.

Hoiem et al. [41] developed a system that classifies the surface orientation of different regions in an image as horizontal and vertical. Figure 4.10 shows two example annotations obtained using their approach: the first image shows the output of their system on an image showing a city scene containing streets and buildings; the second image shows the output of their system on a sports (tennis) image. In general, they achieved impressive surface annotations for city scenes, but failed to produce good annotations for the sport scenes. This is perhaps because the features and statistics used in their model were engineered for the domain of outdoor scenes and are not useful for annotating sports images.

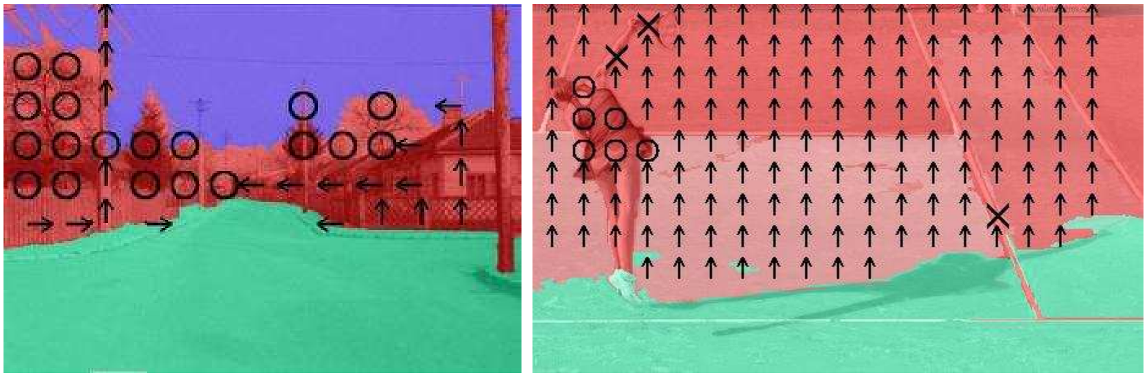


Figure 4.10. Examples of surface orientation annotations obtained by Hoiem et al.'s approach.

Surface orientation annotation obtained by Hoiem et al. [41]. Green color represents horizontal and red represents vertical. For this work, we ignore the subdivisions of the vertical surfaces: planar orientations (arrows), non-planar solid ('x') and porous ('o'). The left image shows the results on an outdoor scene with buildings, and the right image shows the results on a sports-related image. While their results are very impressive on street scenes, we did not find the learned statistics to be useful in modeling the surface orientations in sports images.

Since a general-purpose classifier for the surface orientation of an image region (e.g., the classifier discussed above) could not be applied to solve the problem of segmenting playing surfaces in sports images, we trained a classifier specifically for this problem. In particular, we used a support vector machine (SVM) based classifier.

To train this classifier, we labeled all of the regions in a small set of images with labels $\mathbf{l}$ to denote if the corresponding image regions are part of the playing surface. Also, each of these image regions is represented using the set of features shown in Table 4.1.

Note that the parameters of an SVM classifier are estimated by minimizing the label error for the training samples. Since the number of image regions that are part of the playing surface is much less than the number of image regions that are not, the learned classifier tends to classify any image region as not a part of the playing surface. This effect can be avoided either by using a different loss function for training the SVM classifier, or by balancing the class frequencies in the training set. Following the the latter option, we use the synthetic minority over-sampling technique (SMOTE) [19] to balance the class frequencies.

For a given image region, the trained SVM classifier projects it into a reproducing kernel Hilbert space and determines the classification label using a hyper-plane (whose parameters are learned during training) that separates the instances of the two classes. We use the distance d between the projection of the image region and this separating hyper-plane to define

$$p(\mathrm{l}|d) = \frac{\exp(d)}{1+\exp(d)}, \tag{4.21}$$

where l is a binary label that denotes if this image region is a part of the playing surface. This probability term is used as the representation of the base hypothesis in our model.

This section concludes the description of the details of our model. Next, we present the data set used in our experiments.

4.8 FlickrSports-5 data set

Flickr[4] is an online photo management application that provides an API[5] to search and download images using text queries. We used this API to retrieve images for several text queries specifying various team-names and venues for five popular sports: baseball, basketball, football, soccer, and tennis, Some examples of these queries are: "Red Sox," "Miami Heat," "New England Patriots," "FIFA," and "French Open." From the set of retrieved images, we discarded the images without a significant view of the playing field, but did not restrict the images to include the entire view of the field. Note that some of the images include players, balls, or other objects occluding the distinctive markings on the playing surface. Figure 4.11 shows some examples of the images in our collection.

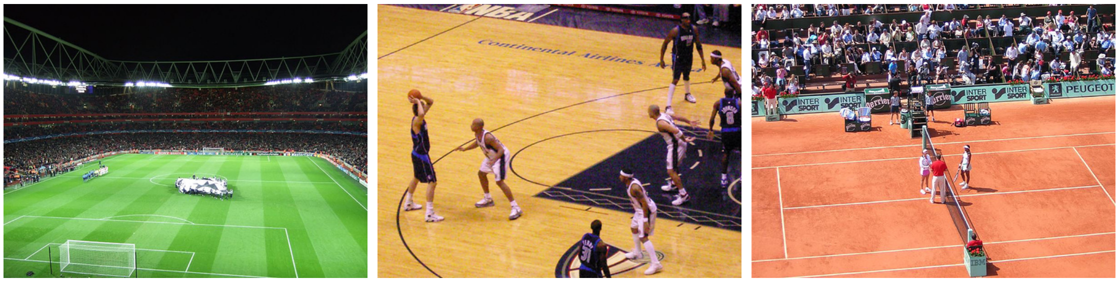


Figure 4.11. Example images from the FlickrSports-5 data set.

Our final collection contains 2449 images with roughly the same number of images for each of the five sports. For our experiments, we split this data set into three parts: 50% for training, 25% for validation, and 25% for testing. The training and validation sets are used for tuning the parameters, and the test set is used for evaluating different approaches for sport classification.

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

[5] Application Programming Interface

4.9 Experiments

To train the classifier to generate the base hypotheses for the image regions (discussed in Section 4.7), we labeled all of the image regions in 200 images from our collection. As shown in Table 4.2, we achieved a significant improvement in the average class accuracy by balancing the class frequencies in the training data.

Training set	Average class accuracy
original data	55.87 ± 3.70
data balanced by SMOTE [19]	89.58 ± 2.33

Table 4.2. Effect of balancing the class frequencies on a image region classifier.

Some example predictions (base hypotheses) obtained from this classifier are shown in Figure 4.12. In Figure 4.13, we show some examples of the estimation of lines that are used to compute the geometric features described in Section 4.6.2.

In this section, our SHRF model is compared with the three different approaches for scene classification:

1. **SVM+SVM**. We train linear SVM classifiers that take an image region as input and predicts a binary label that denotes if this region is part of the playing surface. In this section, we report the performance of this approach for the following four choices of representations of image regions (as discussed in Section 4.6):

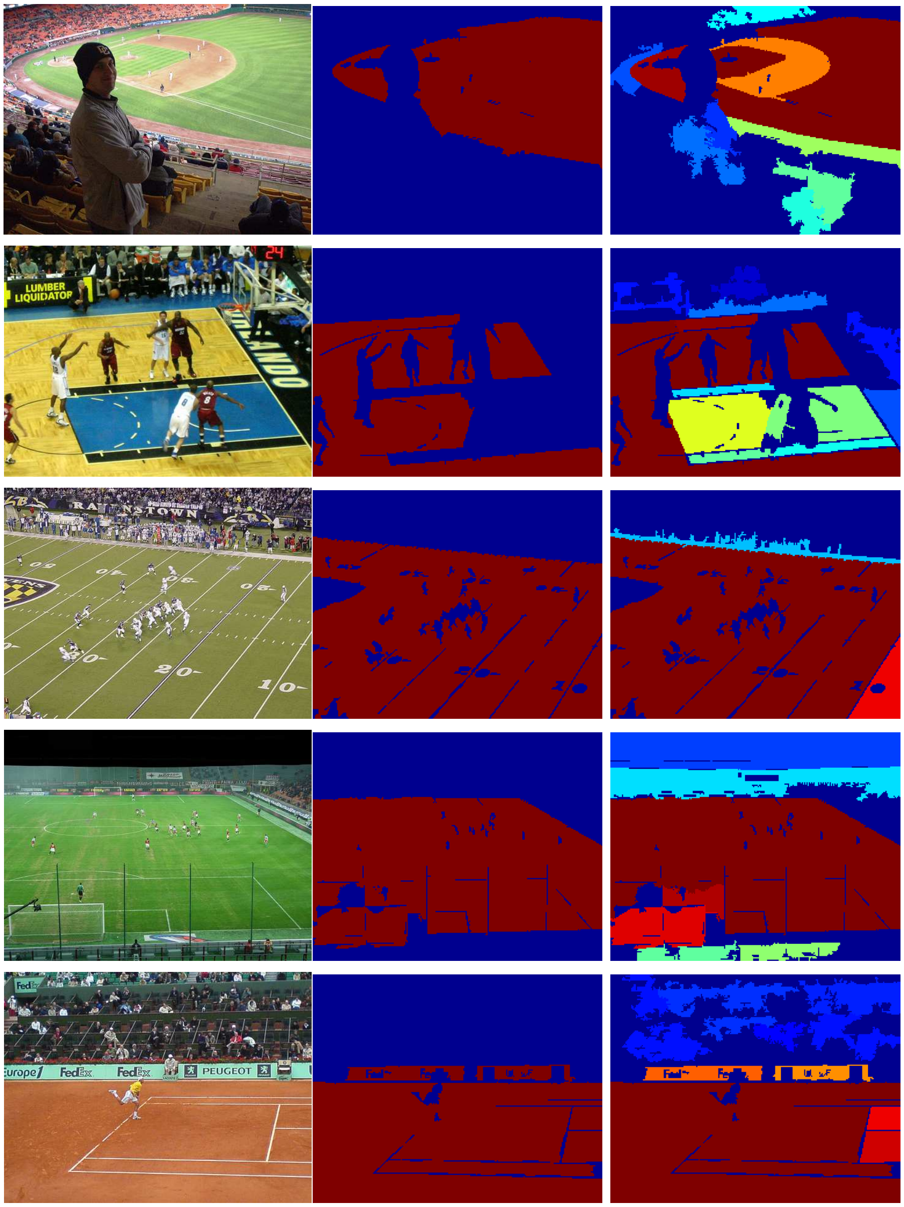
   (a) location and appearance features,

   (b) geometric features,

   (c) bag of visual words on interest points,

   (d) all of the above features.

   The image regions that these model predicted to be a part of the playing surface are then used as input to a linear SVM classifier to predict the final sport label



Original image *Binary prediction* *Continuous prediction*

Figure 4.12. Examples of the predictions of the base hypotheses.

The first column shows example images of five different sports. The second column shows the binary predictions from the base classifier, where the predicted segmentation of the playing surface are shown in red color. The third column shows a real-valued representation of the base hypotheses (given by Equation 4.21) using a range of colors with extremes being the blue and red colors. The blue color represents a low probability value for an image region to be a part of the playing surface, and red color represents a corresponding high probability value.

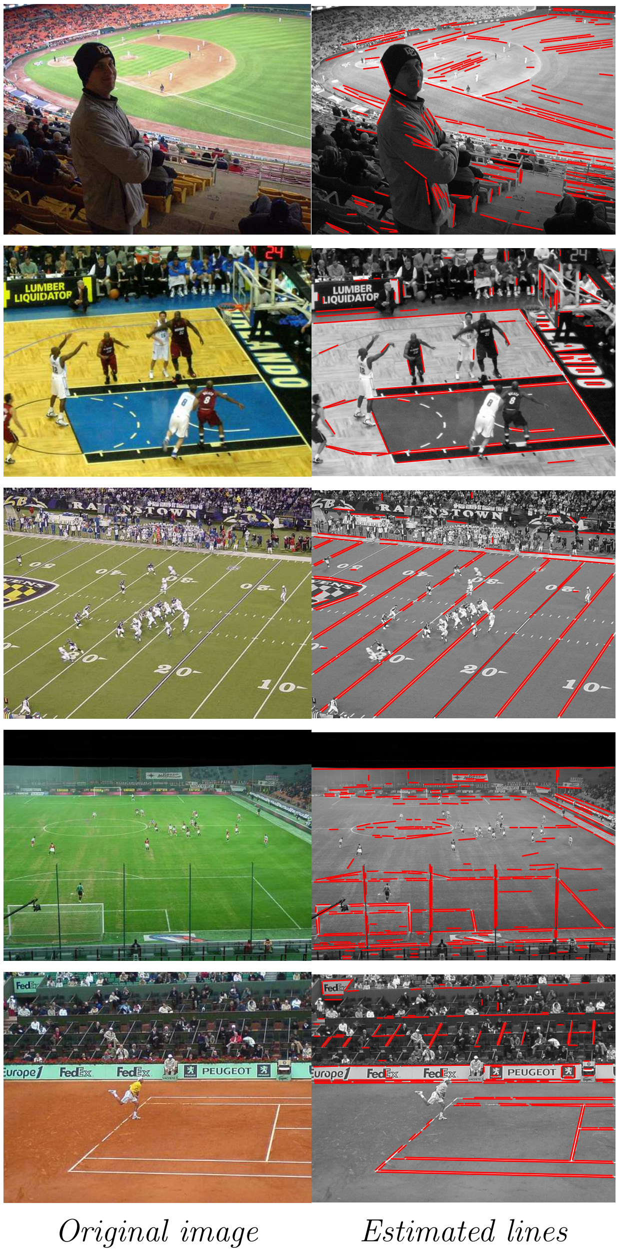


Figure 4.13. Examples of the estimation of lines in images.

The first column shows example images of five different sports. The lines identified in these images are displayed in red in the second column.

for the given image. Because of the sequential processing of images through two different SVMs, we refer to this approach as *SVM+SVM*.

2. **CRF+SVM**. We build a CRF model for predicting the segmentation of the playing surfaces in images. In this model, the connectivity over the random variables for the prediction labels is defined to be the same as the connectivity over the latent variables in the SHRF model. This CRF model is trained using the labeled examples used for training the base classifier for playing surface (Section 4.7). The image regions that this model predicted to be a part of the playing surface are then used as input to a linear SVM classifier to predict the final sport label for the given image. We refer to this approach as *CRF+SVM*.

3. **HCRF**. We also include a hidden-state conditional random field discussed in Section 4.4.2 in our experiments. In particular, we implemented Quattoni et al.'s model [85] for gesture recognition with the choices of potential functions similar to the ones used in our SHRF model.

Figure 4.14 shows a qualitative comparison of the predicted surface annotations obtained using these approaches. Note that the ground truth annotations for the regions in all of the images in our collection are not available. Hence we can not perform a quantitative comparison of the segmentation results of different approaches.

Table 4.3 compares the average class accuracies for 5-fold cross validation experiments on the test set of the FlickrSports-5 data set (Section 4.8). The important observations in this comparison are as follows.

- Both of the CRF+SVM and SVM+SVM models use similar approach for predicting the sport label for a given image. There is one key difference between these models: the SVM+SVM model performs an independent classification of the playing surface label for different image regions, whereas CRF+SVM uses a random field with the neighborhood compatibility constraint (Section 4.3) for performing a collective inference for all of the regions in a single image. Using this additional compatibility constraint, an improvement of about four percent in the average class accuracy is observed in our experiments.

- The SHRF model outperforms all of the other approaches in average class accuracies and provides qualitatively better segmentations (see Figure 4.14) of the playing surface. Another key observation in the comparison between SHRF and CRF+SVM is that a joint segment-*and*-classify approach improves upon a sequential segment-*then*-classify approach.

- The classification results obtained from the HCRF model are not competitive with other approaches. Also, the learned hidden variables do not correspond to the segmentation of the playing surfaces in images.

Approach	Average class accuracy
SVM+SVM	
Visual Vocabulary	41.56 ± 1.79
image region features	52.07 ± 0.81
Line features	50.83 ± 4.22
All features	56.76 ± 3.40
CRF + SVM	61.38 ± 2.01
HCRF	31.94 ± 4.19
SHRF	**65.28 ± 3.85**

Table 4.3. Average class accuracy for sport classification.

The error terms correspond to 5-fold cross-validation experiments. The results for HCRF are not competitive with the other approaches.

4.10 Conclusions

In this chapter, we presented an extension of hidden-state conditional random fields, which we refer to as selective hidden random fields. This model simultaneously does the segmentation of the object of interest in an image and uses it for scene classification. We applied this model to solve a very challenging scene classification problem, i.e., sport classification. In the context of sport classification, our model simultaneously identifies the playing surface and characterizes the playing surface to classify the sporting event.

While the experiments in this chapter have been limited to classifying sporting events in images, we believe that this model can be applied to other domains where a selective processing of data needs to be done. For instance, this model could be applied to autonomous navigation, where it could be used to simultaneously segment the horizontal surfaces and characterize them to determine the locations of the valid driving areas or the cross-walks on the segmented horizontal surfaces.

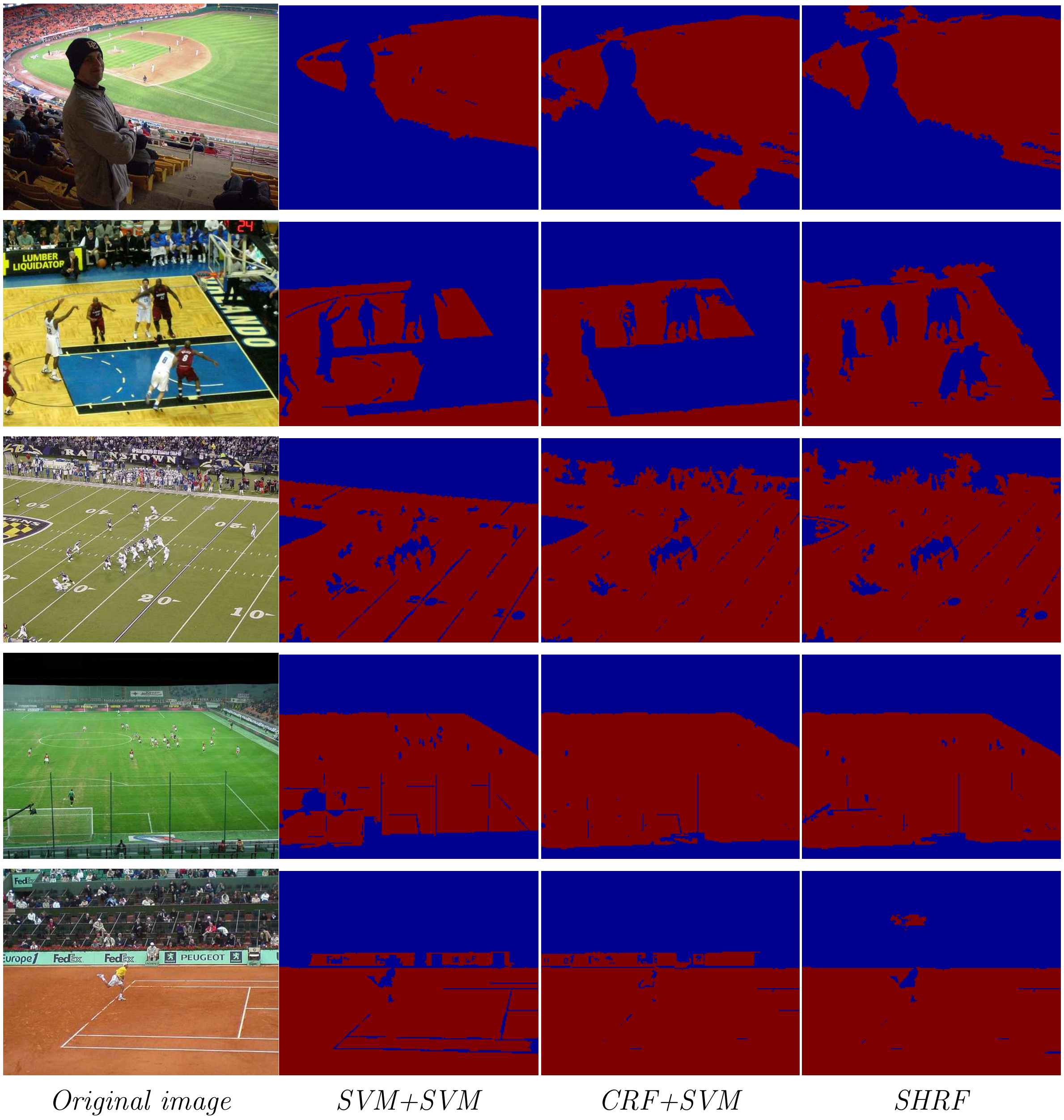


Figure 4.14. Example predictions of the playing surface using different approaches.

The first column shows example images of five different sports. The next three columns show the segmentation of the playing surface obtained using different approaches, where the predictions for the playing surface are shown in red. It is clear that the segmentation obtained by an independent image region model (column 2) are improved by including the dependencies on the neighbors (column 3) as the "holes" are filled and most of the field markers are correctly labeled. The segmentations obtained by our model (column 4) are similar to those of CRF, and further improves the labeling for surfaces with very dissimilar appearance for neighboring image regions (see basketball image, row 2).

CHAPTER 5
CONCLUSIONS

In this dissertation, some contextual cues have been explored to improve the solutions for the face detection and recognition problems.

First, we establish a competitive benchmark for evaluating face detection algorithms. To build a data set of face images in unconstrained settings, the faces appearing in a collection of news photographs are annotated. This FDDB data set has more faces and more accurate annotations for face regions than in previous data sets. To further establish a benchmark for the evaluation of face detection algorithms, two rigorous evaluation schemes are presented.

Next, an algorithm for face detection is presented, which uses the context from easy-to-detect faces in an image to help in the analysis of the difficult-to-detect faces in the image. This algorithm uses an on-line approach for rapidly adapting a "black box" classifier to a new test data set without retraining the classifier or examining the original optimization criterion. Assuming the original classifier outputs a continuous number whose threshold gives the class, points near the original boundary are reclassified using a Gaussian process regression scheme. This face detection algorithm achieved substantial improvement in performance over the state-of-the-art on the FDDB benchmark.

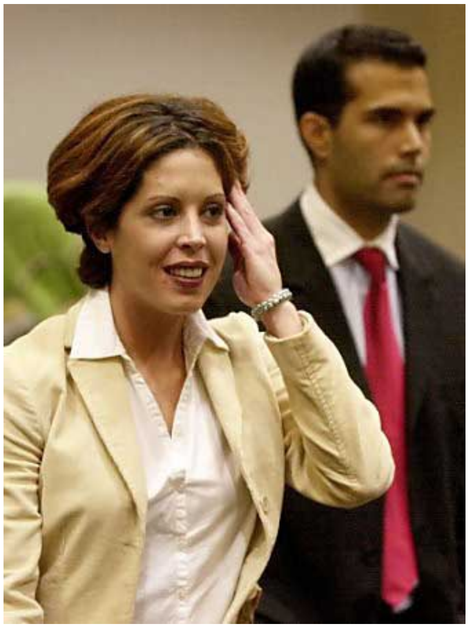
Then, for face recognition, a joint probabilistic model (People-LDA) for image-caption pairs is presented, which captures the coherence between the faces appearing in the image and the names appearing in the associated caption. Using a pre-trained face identifier in a probabilistic topic modeling framework, People-LDA guides semantic topics to develop around people. In addition to using the language model to boost the performance for face recognition in constrained environments, this model learns different distributions of words, each of which are closely related to a single person.

Finally, we presented a probabilistic model that simultaneously segments the object of interest in an image and uses it for scene classification. This model is applied to solve a very challenging scene classification problem, i.e., sport classification. The predicted scene classification label from this model is likely to be helpful in specifying additional context for both of the face detection and face recognition problems. Building models that employ this type of context for face analysis would be an interesting future research direction.

5.1 Future work

This dissertation presented a few ways to incorporate contextual cues in the solutions for analyzing images with faces. This document is clearly not an exhaustive study of the use of context for such an analysis. There are several other kind of contextual cues that would be useful for detecting and recognizing faces in images. For instance, the caption of an image could provide useful information about the expected number of faces appearing in the image (see Figure 5.1 for an example). The estimated number of faces could then be used to appropriately adapt the face detector.

We presented separate solutions for the face detection and recognition problems, which need to be integrated into an end-to-end system for contextual face analysis. Note that one of our models is a joint model and another is a conditional model. Combining these different kinds of models is very challenging. Hoiem et al. [40, 41, 39] presented one such combination of different models for object recognition, surface orientation estimation, and occlusion boundary detection to obtain improvements



Noelle Bush (L) is shown in Orange County, Florida court after her status hearing July 19, 2002. At right is her brother, George P. Bush.

Figure 5.1. Using the caption of an image to infer the number of faces appearing in the image.

Since there are two names (Noelle Bush and George P. Bush) present in the caption, it is very likely that there are exactly two faces appearing in this image. It is possible to have more or fewer faces present in the image, but the posterior probability for the number of faces, given the presence of two names in the caption, is likely to be centered around two.

in the performance of each of these modules independently. In more recent work, Heitz et al. [35, 36] proposed a framework called cascaded classification models. This framework arranges multiple copies of the models for scene categorization, object detection, and segmentation in different layers and uses the output from all of the models in one layer to bootstrap the performance of the models in the subsequent layer. Evaluating these frameworks for combining our models would be an interesting extension to this dissertation.

APPENDIX

GUIDELINES FOR ANNOTATING FACES USING ELLIPSES

Multiple human annotators were asked to draw ellipses around the face regions in images. They were instructed to approximate the shape of a human head as the union of two ellipsoids as shown in Figure A.1. To ensure consistency across the annotators, we developed a set of instructions that are illustrated in Figure A.2. A formal description of these instructions using a flow-chart is shown in Figure A.3. These instructions specify how to use facial landmarks to fit an ellipse depending on the pose of the head. Figure A.4 presents an illustration of the resulting ellipses on line drawings of a human head. The annotators were futher instructed to follow a combination of these guidelines to fit ellipses to faces with complex head poses.

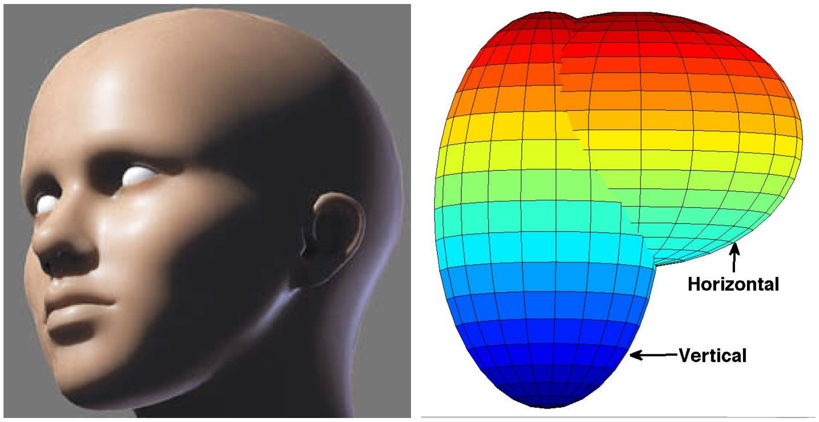


Figure A.1. An approximation of the shape of a human head.

We approximate the shape of a human head (**left**) as the union of two ellipsoids (**right**). We refer to these ellipses as vertical and horizontal ellipsoids.

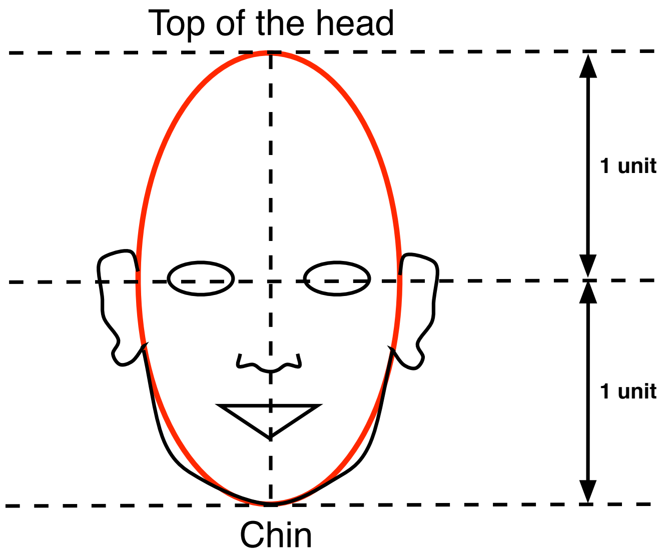


Figure A.2. Guidelines for drawing ellipses around face regions.

The extreme points of the major axis of the ellipse are respectively matched to the chin and the "top of the head," which is defined as the topmost point of the hypothetical vertical ellipsoid (see Figure A.1) used for approximating the human head. Note that this ellipse does not include the ears. Also, for a non-frontal face, at least one of the lateral extremes (left or right) of this ellipse are matched to the boundary between the face region and the corresponding (left or right) ear. The details of our specifications are included in Section A.

Figure A.5 shows some example annotations obtained using these guidelines. The illustrations shown in Figure A.4 use faces with neutral expressions. A presence of some expressions such as laughter, often changes the shape of the face significantly. Moreover, even bearing a neutral expression, some faces have shapes markedly different from the average face shape used in these illustrations. Such faces (e.g., faces with square-jaw or double-chin) are difficult to approximate using ellipses. To annotate faces with such complexities, the annotators were instructed to refer to the following guidelines:

- **Facial expression**. Since the distance from the eyes to the chin in a face with facial expression is not necessarily equal to the distance between the eyes and

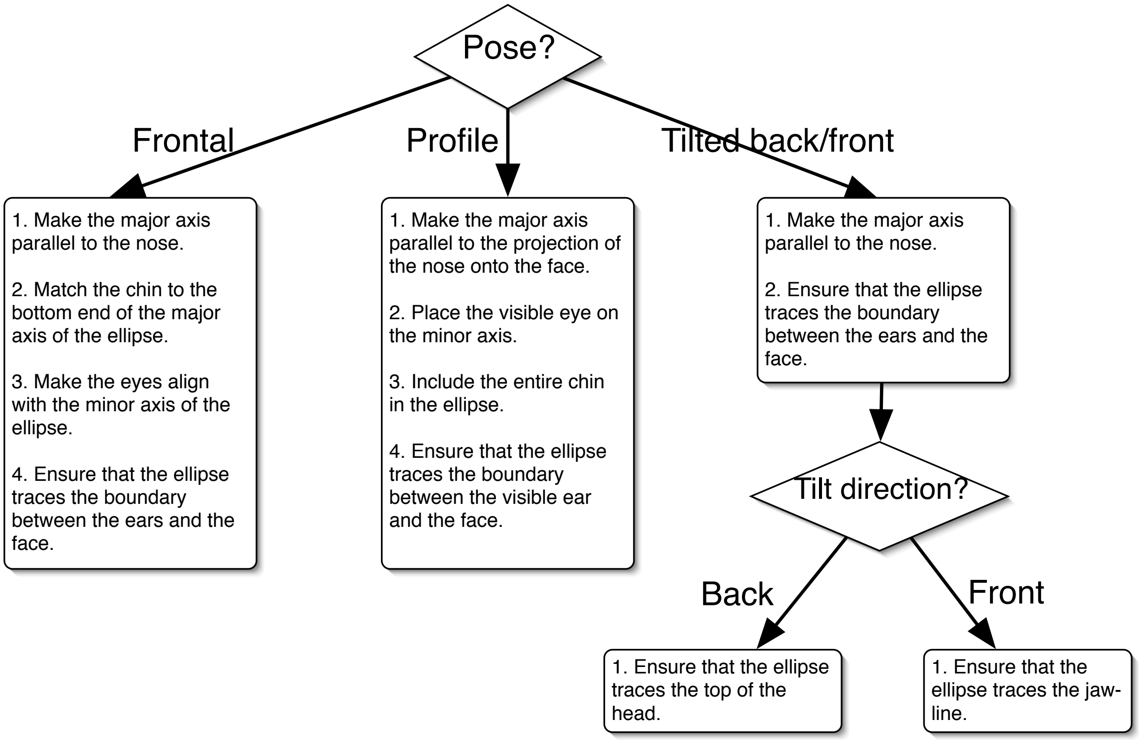

Pose?
Frontal
Profile
Tilted back/front
1. Make the major axis parallel to the nose.
2. Match the chin to the bottom end of the major axis of the ellipse.
3. Make the eyes align with the minor axis of the ellipse.
4. Ensure that the ellipse traces the boundary between the ears and the face.
1. Make the major axis parallel to the projection of the nose onto the face.
2. Place the visible eye on the minor axis.
3. Include the entire chin in the ellipse.
4. Ensure that the ellipse traces the boundary between the visible ear and the face.
1. Make the major axis parallel to the nose.
2. Ensure that the ellipse traces the boundary between the ears and the face.
Tilt direction?
Back
Front
1. Ensure that the ellipse traces the top of the head.
1. Ensure that the ellipse traces the jaw-line.

Figure A.3. Procedure for drawing ellipses around an average face region.

The annotators were instructed to follow this flowchart to draw ellipses around the face regions. The annotation steps are a little different for different poses. Here, we present the steps for three canonical poses: frontal, profile and tilted back/front. The annotators were instructed to use a combination of these steps for labeling faces with derived, intermediate head poses. For instance, to label a head facing slightly towards its right and titled back, a combination of the steps corresponding to the profile and tilted-back poses are used.

the top of the head (an assumption made for the ideal head), the eyes do not need to be aligned to the minor axis for this face.

- **Double-chin**. For faces with a double chin, the average of the two chins is considered as the lowest point of the face, and is matched to the bottom extreme of the major axis of the ellipse.

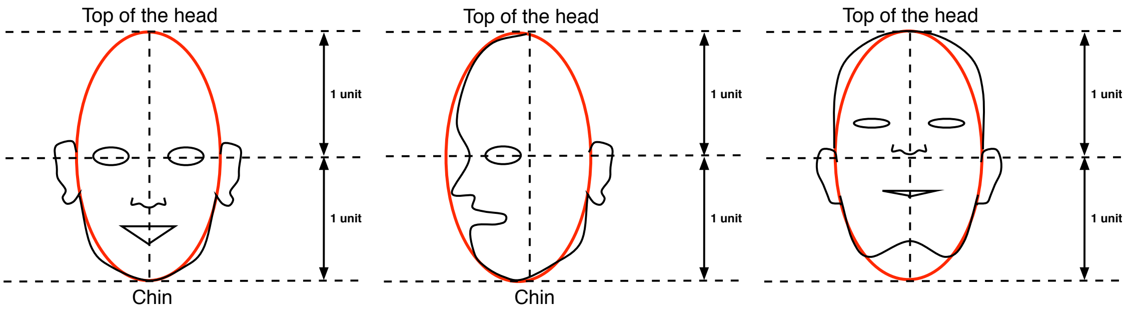


Figure A.4. Illustrations of ellipse labeling on line drawings of human head.

The black curves show the boundaries of a human head in *frontal* (**left**), *profile* (**center**), and *tilted-back* (**right**) poses. The red ellipses illustrate the desired annotations as per the procedure shown in Figure A.3. Note that these head shapes are approximations to an average human head, and the shape of an actual human head may deviate from this mean shape. The shape of a human head may also be affected by the presence of factors such as emotions. The guidelines on annotating face regions influenced by these factors are specified in Section A.

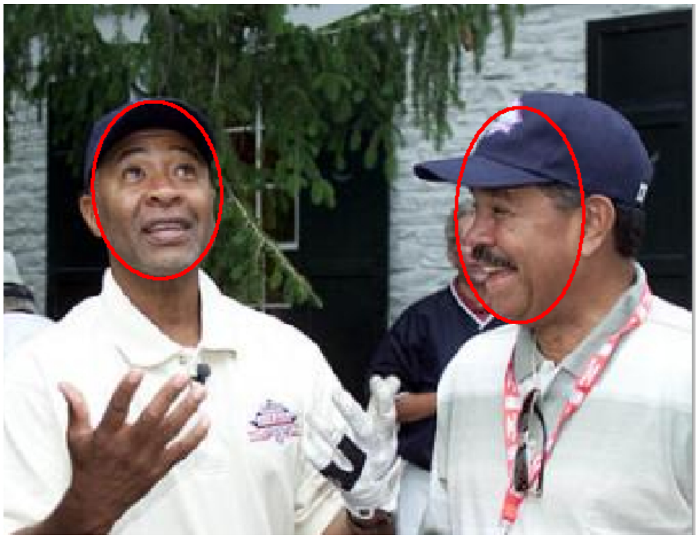


Figure A.5. Example elliptical annotations for face regions.

The two red ellipses in this image specify the two faces present in this image. Note that for a non-frontal face (**right**), the ellipse traces the boundary between the face and the visible ear. As a result, the elliptical region includes pixels that are not a part of this face.

- **Square jaw**. For a face with a square jaw, the ellipse traces the boundary between the face and the ears, while some part of the jaws may be excluded from the ellipse.

- **Hair**. Ignore the hair and fit the ellipse around the hypothetical bald head.

- **Occlusion**. Hypothesize the full face behind the occluding object, and match all of the visible features.

Figure A.6 shows some example annotations for complex face shapes.

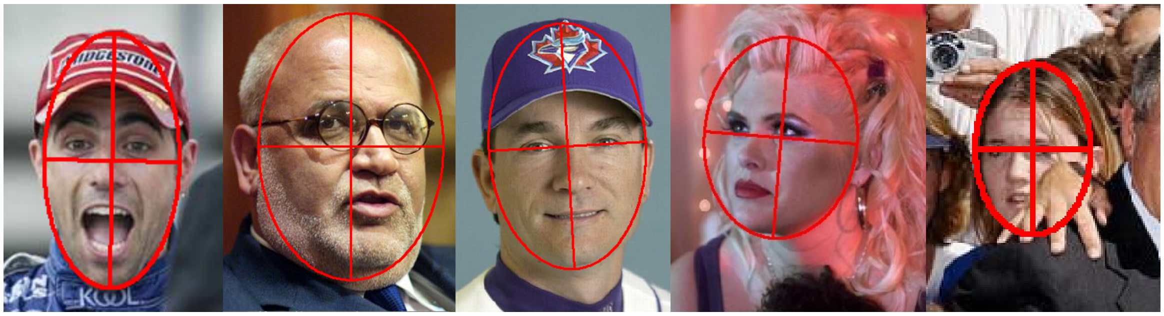


Figure A.6. Illustrations of labeling for complex face appearances.

Illustrations of labeling for complex face appearances. These images show example annotations for human heads with shapes different from an average human head due to the presence of *facial expression*, *double chin*, *square jaw*, *hair-do*, and *occlusion*, respectively.

BIBLIOGRAPHY

[1] Abdallah, Abdallah S., El-nasr, Mohamad Abou, and Abbott, A. Lynn. A new color image database for benchmarking of automatic face detection and human skin segmentation techniques. In *International Conference on Machine Learning and Pattern Recognition* (2007). 9

[2] Abramowitz, Milton, and Stegun, Irene A. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Dover, New York, 1964. 79

[3] Ahuja, Narendra, and Yang, Ming-Hsuan. *Face Detection and Gesture Recognition for Human-Computer Interaction*. Kluwer, 2001. 1

[4] Bar, Moshe. Visual objects in context. *Nature Reviews: Neuroscience 5* (August 2004), 617–629. 2

[5] Barnard, Kobus, Duygulu, Pinar, Forsyth, David A., de Freitas, Nando, Blei, David M., and Jordan, Michael I. Matching words and pictures. *Journal of Machine Learning Research 3* (2003), 1107–1135. 82, 85, 88, 89

[6] Beal, Matthew J. *Variational Algorithms for Approximate Bayesian Inference*. PhD thesis, University of Cambridge, 2003. 80

[7] Behrmann, Marlene, and Avidan, Galia. Congenital prosopagnosia: Face-blind from birth. *Trends in Cognitive Sciences 9*, 4 (2005), 180 – 187. 1

[8] Belhumeur, Peter N., Hespanha, Joao, and Kriegman, David J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. In *European Conference on Computer Vision* (1996), vol. 1, pp. 45–58. 84

[9] Belkin, Mikhail, Niyogi, Partha, and Sindhwani, Vikas. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research 7* (2006), 2399–2434. 50

[10] Berg, Tamara L., Berg, Alexander C., Edwards, Jaety, Maire, Michael, White, Ryan, Teh, Yee Wye, Learned-Miller, Erik, and Forsyth, David A. Names and faces in the news. In *IEEE Conference on Computer Vision and Pattern Recognition* (2004), vol. 2, pp. 848–854. xvi, 10, 12, 80, 86, 87, 88, 89, 90, 91

[11] Bickel, Steffen, Brückner, Michael, and Scheffer, Tobias. Discriminative learning under covariate shift. *Journal of Machine Learning Research 10* (2009), 2137–2155. 50

[12] Biederman, Irving, Mezzanotte, Robert J., and Rabinowitz, Jan C. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive Psychology 14*, 2 (April 1982), 143–177. 2

[13] Blei, David M. *Probabilistic Models of Text and Images.* PhD thesis, University of California Berkeley, 2004. 75, 78

[14] Blei, David M., and Jordan, Michael I. Modeling annotated data. In *ACM SIGIR Conference on Research and Development in Informaion Retrieval* (2003), ACM Press, pp. 127–134. 83, 86, 89

[15] Blei, David M., and Lafferty, John. Correlated topic models. In *Advances in Neural Information Processing Systems.* MIT Press, 2006. 91

[16] Blei, David M., Ng, Andrew Y., and Jordan, Michael I. Latent Dirichlet allocation. *Journal of Machine Learning Research 3* (March 2003), 993–1022. 71, 72, 78, 80, 89, 93

[17] Blitzer, John, Crammer, Koby, Kulesza, Alex, Pereira, Fernando, and Wortman, Jenn. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems* (2008), MIT Press. 30, 50

[18] Carbonetto, Peter, de Freitas, Nando, and Barnard, Kobus. A statistical model for general contextual object recognition. In *European Conference on Computer Vision* (2004), pp. Vol I: 350–362. 3

[19] Chawla, Nitesh V., Bowyer, Kevin W., Hall, Lawrence O., and Kegelmeyer, W. Philip. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research 16* (June 2002), 321–357. 117, 119

[20] Chum, Ondřej, Philbin, James, Isard, Michael, and Zisserman, Andrew. Scalable near identical image and shot detection. In *ACM International Conference on Image and Video Retrieval* (2007), ACM, pp. 549–556. 14

[21] Chum, Ondřej, Philbin, James, and Zisserman, Andrew. Near duplicate image detection: min-hash and tf-idf weighting. In *British Machine Vision Conference* (2008). 14

[22] Comaniciu, Dorin, and Meer, Peter. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 603–619. 15, 102, 111

[23] Corduneanu, Adrian, and Jaakkola, Tommi. On information regularization. In *Conference on Uncertainty in Artificial Intelligence* (2003), pp. 151–158. 30, 49

[24] Daumé III, Hal, and Marcu, Daniel. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research 26* (2006), 101–126. 30, 49

[25] Dickey, James M. Multiple hypergeometric functions: Probabilistic interpretations and statistical uses. *Journal of the American Statistical Association 78*, 383 (1983), 628–637. 75

[26] Efron, Bradley, Hastie, Trevor, Johnstone, Iain, and Tibshirani, Robert. Least angle regression. *Annals of Statistics 32* (2004), 407–499. 58

[27] Elkan, Charles. Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution. In *International Conference on Machine Learning* (2006). 92

[28] Feix, Wolfgang H., and Ruell, Hartwig E. Personal access control system using speech and face recognition, May 1984. 1

[29] Ferencz, Andras, Learned-Miller, Erik G., and Malik, Jitendra. Learning to locate informative features for visual identification. *International Journal of Computer Vision 77*, 1-3 (2008), 3–24. 56, 61, 97

[30] Foo, Jun Jie, Zobel, Justin, Sinha, Ranjan, and Tahaghoghi, S. M. M. Detection of near-duplicate images for web search. In *ACM International Conference on Image and Video Retrieval* (2007), ACM, pp. 557–564. 14

[31] Gauthier, Isabel, Behrmann, Marlene, and Tarr, Michael J. Can face recognition really be dissociated from object recognition? *Journal of Cognitive Neuroscience 11*, 4 (1999), 349–370. 1

[32] Griffiths, Thomas L., and Steyvers, Mark. Finding scientific topics. *Proceedings of the National Academy of Sciences 101*, Suppl. 1 (April 2004), 5228–5235. 71

[33] Gross, Ralph, Baker, Simon, Matthews, Iain, and Kanade, Takeo. Face recognition across pose and illumination. In *Handbook of Face Recognition*, Stan Z. Li and Anil K. Jain, Eds. Springer-Verlag, 2004. 1

[34] He, Xuming, Zemel, Richard S., and A.Carreira-Perpinian, Miguel. Multiscale conditional random fields for image labeling. In *IEEE Conference on Computer Vision and Pattern Recognition* (2004), vol. 2, pp. 695–702. 105

[35] Heitz, Geremy, Gould, Stephen, Saxena, Ashutosh, and Koller, Daphne. Cascaded classification models: Combining models for holistic scene understanding. In *Advances in Neural Information Processing Systems* (2008). 128

[36] Heitz, Geremy, and Koller, Daphne. Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision* (2008), pp. I: 30–43. 128

[37] Hjelmas, Erik, and Low, Boon Kee. Face detection: A survey. *Computer Vision and Image Understanding 83*, 3 (September 2001), 236–274. 27

[38] Hofmann, Thomas. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning 42*, 1-2 (2001), 177–196. 71

[39] Hoiem, Derek, , Efros, Alexei A., and Hebert, Martial. Putting objects in perspective. In *IEEE Conference on Computer Vision and Pattern Recognition* (2006). 127

[40] Hoiem, Derek. *Seeing the World Behind the Image: Spatial Layout for 3D Scene Understanding*. PhD thesis, Carnegie Mellon University, 2007. 127

[41] Hoiem, Derek, Efros, Alexei A., and Hebert, Martial. Geometric context from a single image. In *IEEE International Conference on Computer Vision* (2005). 3, 116, 127

[42] Hsu, Rein-Lien, Abdel-Mottaleb, Mohamed, and Jain, Anil K. Face detection in color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 696–706. 9

[43] Hu, Diane J., and Saul, Lawrence. A probabilistic topic model of unsupervised learning for musical-key profiles. In *International Society for Music Information Retrieval Conference* (2009). 71

[44] Huang, Chang, Ai, Haizhou, Li, Yuan, and Lao, Shihong. High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 4 (2007), 671–686. 24

[45] Huang, Gary B., Jain, Vidit, and Learned-Miller, Erik. Unsupervised joint alignment of complex images. In *IEEE International Conference on Computer Vision* (2007). 57

[46] Jain, Vidit. Naïve Bayes vs. logistic regression: An assessment of the impact of the misclassification cost. In *NIPS'09 Workshop on the Generative and Discriminative Learning Interface* (2009). 70

[47] Jain, Vidit, Ferencz, Andras, and Learned-Miller, Erik. Discriminative training of hyper-feature models for object identification. In *British Machine Vision Conference* (2006), pp. 357–366. 4, 89, 92, 97

[48] Jain, Vidit, and Learned Miller, Erik. FDDB: A benchmark for face detection in unconstrained settings. Tech. rep., University of Massachusetts Amherst, 2010. 4

[49] Jain, Vidit, and Learned-Miller, Erik. Online domain-adaptation of a pre-trained cascade of classifiers. In *Under Review* (2010). 4

[50] Jain, Vidit, Learned-Miller, Erik, and McCallum, Andrew. People-LDA: Anchoring topics to people using face recognition. In *IEEE International Conference on Computer Vision* (2007). 4, 97

[51] Jain, Vidit, Singhal, Amit, and Luo, Jiebo. Selective hidden random fields: Exploiting domain-specific saliency for event classification. In *IEEE Conference on Computer Vision and Pattern Recognition* (2008). 5

[52] Jain, Vidit, and Varma, Manik. Learning to re-rank: Query-dependent image re-ranking using click data. In *Under Review* (2010). 4

[53] Jones, Michael J., and Viola, Paul A. Fast multi-view face detection. Tech. Rep. TR2003-96, Mitsubishi Electric Research Laboratories, August 2003. 24

[54] Kapoor, Ashish, and Winn, John. Located hidden random fields: Learning discriminative parts for object detection. In *European Conference on Computer Vision* (2006). 107

[55] Kelly, David J., Quinn, Paul C., Slater, Alan M., Lee, Kang, Gibson, Alan, Smith, Michael, Ge, Liezhong, and Pascalis, Olivier. Three-month-olds, but not newborns, prefer own-race faces. *Developmental Science 8*, 6 (2005), F31–6. 1

[56] Kienzle, Wolf, Bakır, Gökhan H., Franz, Matthias O., and Schölkopf, Bernhard. Face detection — efficient and rank deficient. In *Advances in Neural Information Processing Systems* (2005), MIT Press, pp. 673–680. 42

[57] Kosecka, Jana, and Zhang, Wei. Video compass. In *European Conference on Computer Vision* (2002). 114

[58] Kuhn, Harold W. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2* (1955), 83–97. 22

[59] Kumar, Sanjiv, and Hebert, Martial. Discriminative random fields. *International Journal of Computer Vision 68*, 2 (June 2006), 179–201. 105

[60] Lafferty, John, McCallum, Andrew, and Pereira, Fernando. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning* (2001), pp. 282–289. 104, 105

[61] Lawrence, Neil D., and Jordan, Michael I. Semi-supervised learning via Gaussian processes. In *Advances in Neural Information Processing Systems* (2005), MIT Press, pp. 753–760. 30, 49

[62] Learned-Miller, Erik G., and Jain, Vidit. Many heads are better than one: Jointly removing bias frommultiple mris using nonparametric maximum likelihood. In *Information Processing in Medical Imaging* (2005), pp. 615–626. 57

[63] Li, Stan Z., Zhu, Long, Zhang, ZhenQiu, Blake, Andrew, Zhang, HongJiang, and Shum, Harry. Statistical learning of multi-view face detection. In *European Conference on Computer Vision* (2002), pp. 67–81. 24

[64] Li, Zhifeng, and Tang, Xiaoou. Bayesian face recognition using support vector machine and face clustering. In *IEEE Conference on Computer Vision and Pattern Recognition* (2004), pp. II: 374–380. 1

[65] Liang, P., and Jordan, M. I. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. In *International Conference on Machine Learning* (2008). 70

[66] Loui, A.C., Judice, C.N., and Liu, Sheng. An image database for benchmarking of automatic face detection and recognition algorithms. In *IEEE International Conference on Image Processing* (Oct 1998), vol. 1, pp. 146–150 vol.1. 9

[67] Lowe, David G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110. 86, 113

[68] Maron, Oded, and Ratan, Aparna Lakshmi. Multiple-instance learning for natural scene classification. In *International Conference on Machine Learning* (1998), pp. 341–349. 98

[69] Matas, J, Chum, O, Martin, U, and Pajdla, T. Robust wide baseline stereo from maximally stable extremal regions. In *British Machine Vision Conference* (London, 2002), vol. 1, pp. 384–393. 113

[70] McCallum, Andrew, and Li, Wei. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Conference on Computational Natural Language Learning* (2003). 81

[71] McCullagh, Peter, and Nelder, John A. *Generalized Linear Models.* Chapman and Hall, 1989. 58

[72] Mikolajczyk, K., Choudhury, R., and Schmid, C. Face detection in a video sequence - a temporal approach. In *IEEE Conference on Computer Vision and Pattern Recognition* (2001), vol. 2, p. 96. 6

[73] Mikolajczyk, Krystian, and Schmid, Cordelia. Comparison of affine-invariant local detectors and descriptors. In *European Signal Processing Conference* (2004). 113

[74] Mikolajczyk, Krystian, and Schmid, Cordelia. Scale & affine invariant interest point detectors. *International Journal of Computer Vision 60*, 1 (2004), 63–86. 113

[75] Mikolajczyk, Krystian, Schmid, Cordelia, and Zisserman, Andrew. Human detection based on a probabilistic assembly of robust part detectors. In *European Conference on Computer Vision* (2004), pp. 69–82. 11, 42

[76] Moghaddam, Baback, Jebara, Tony, and Pentland, Alex. Bayesian face recognition. *Pattern Recognition 33*, 11 (2000), 1771–1782. 65, 68, 69, 97

[77] Murphy, Kevin P., Weiss, Yair, and Jordan, Michael I. Loopy belief propagation for approximate inference: An empirical study. In *Conference on Uncertainty in Artificial Intelligence* (1999). 111

[78] Murphy, Shelley, and Bray, Hiawatha. Face recognition devices failed in test at logan. The Boston Globe, March 2003. 2

[79] Ng, Andrew Y., and Jordan, Michael I. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *Advances in Neural Information Processing Systems*. MIT Press, 2002. 70

[80] Ng, Andrew Y., Jordan, Michael I., and Weiss, Yair. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems* (2001), MIT Press, pp. 849–856. 14

[81] Oliva, Aude, and Torralba, Antonio. The role of context in object recognition. *Trends in Cognitive Sciences. 11*, 12 (2007), 520–527. 2

[82] Osadchy, Margarita, LeCun, Yann, and Miller, Matthew L. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research 8* (2007), 1197–1215. 24

[83] Pentland, Alex, and Choudhury, Tanzeem. Face recognition for smart environments. *IEEE Computer 33*, 2 (2000), 50–55. 1

[84] Phillips, P. Jonathon, Moon, Hyeonjoon, Rizvi, Syed A., and Rauss, Patrick J. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22* (2000), 1090–1104. 65, 70

[85] Quattoni, Ariadna, Collins, Michael, and Darrell, Trevor. Conditional random fields for object recognition. In *Advances in Neural Information Processing Systems*. MIT Press, 2005. 122

[86] Quattoni, Ariadna, Wang, Sy Bor, Morency, Louis-Philippe, Collins, Michael, and Darrell, Trevor. Hidden-state conditional random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 10 (October 2007), 1848–1853. 105

[87] Rasmussen, Carl E., and Williams, Christopher K. I. *Gaussian Processes for Machine Learning*. The MIT Press, December 2005. 33

[88] Rowley, Henry A., Baluja, Shumeet, and Kanade, Takeo. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 1 (January 1998), 23–38. 1, 9, 27, 30

[89] Sato, Kengo, and Sakakibara, Yasubumi. RNA secondary structural alignment with conditional random fields. *Bioinformatics 21*, 2 (2005), 237–242. 105

[90] Schneiderman, Henry, and Kanade, Takeo. Probabilistic modeling of local appearance and spatial relationships for object recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (1998), p. 45. 9, 27, 30, 42

[91] Schneiderman, Henry, and Kanade, Takeo. A statistical method for 3D object detection applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition* (2000), vol. 1, pp. 746–751 vol.1. 1, 9

[92] Serrano, Navid, Savakis, Andreas, and Luo, Jiebo. A computationally efficient approach to indoor/outdoor scene classification. In *IEEE International Conference on Pattern Recognition* (2002), vol. 4, p. 40146. 98

[93] Seshadrinathan, Manoj, and Ben-Arie, Jezekiel. Pose invariant face detection. In *4th EURASIP Conference focused on Video/Image Processing and Multimedia Communications, 2003.* (July 2003), vol. 1, pp. 405–410 vol.1. 24

[94] Sharma, Prag, and Reilly, Richard B. A colour face image database for benchmarking of automatic face detection algorithms. In *EURASIP Conference focused on Video/Image Processing and Multimedia Communications* (July 2003), vol. 1, pp. 423–428 vol.1. 1, 9

[95] Shi, Jianbo, and Malik, Jitendra. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (August 2000), 888–905. 86

[96] Sivic, Josef, Russell, Bryan C., Zisserman, Andrew, Freeman, William T., and Efros, Alyosha A. Unsupervised discovery of visual object class hierarchies. In *IEEE Conference on Computer Vision and Pattern Recognition* (June 2008), pp. 1–8. 71

[97] Steyvers, Mark, and Griffiths, Tom. Probabilistic topic models. In *Latent Semantic Analysis: A Road to Meaning*. Laurence Erlbaum, 2007. 71

[98] Subramaniam, S, and Biederman, Irving. Does contrast reversal affect object identification. *Investigative Ophthalmology & Visual Science 38* (1997), 998. 1

[99] Sudderth, Erik B., Torralba, Antonio, Freeman, William T., and Willsky, Alan S. Describing visual scenes using transformed objects and parts. *International Journal of Computer Vision 77*, 1-3 (2008), 291–330. 71

[100] Sung, Kah-Kay, and Poggio, Tomaso. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 1 (1998), 39–51. 1, 9, 30

[101] Szummer, Martin, and Jaakkola, Tommi. Information regularization with partially labelled datas. In *Advances in Neural Information Processing Systems* (2002), MIT Press. 49

[102] Torralba, Antonio. Contextual priming for object detection. *International Journal of Computer Vision 53*, 2 (2003), 169–191. 3

[103] Torralba, Antonio, Murphy, Kevin P., and Freeman, William T. Contextual models for object detection using boosted random fields. In *Advances in Neural Information Processing Systems* (2005), MIT Press, pp. 1401–1408. 3

[104] Torralba, Antonio, and Oliva, Aude. Depth estimation from image structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 9 (September 2002), 1226–1238. 3

[105] Torralba, Antonio, and Sinha, Pawan. Statistical context priming for object detection. In *IEEE International Conference on Computer Vision* (2001), pp. I: 763–770. 3

[106] Turk, Matthew, and Pentland, Alex. Eigenfaces for recognition. *Journal of Cognitive Neuroscience 3*, 1 (1991), 71–86. 84

[107] http://mplab.ucsd.edu. The MPLab GENKI Database, GENKI-4K Subset. 9

[108] Viola, Paul A., and Jones, Michael J. Robust real-time face detection. *International Journal of Computer Vision 57*, 2 (May 2004), 137–154. 1, 25, 28, 30, 42, 81

[109] Wang, Peng, and Ji, Qiang. Multi-view face and eye detection using discriminant features. *Computer Vision and Image Understanding 105*, 2 (2007), 99–111. 24

[110] Wang, Sy Bor, Quattoni, Ariadna, Morency, Louis-Philippe, and Demirdjian, David. Hidden conditional random fields for gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (2006). 107

[111] Wolf, Lior, and Bileschi, Stanley. A critical view of context. *International Journal of Computer Vision 69*, 2 (2006), 251–261. 2

[112] Wu, Dan, Lee, Wee Sun, Ye, Nan, and Chieu, Hai Leong. Domain adaptive bootstrapping for named entity recognition. In *Empirical Methods in Natural Language Processing* (Singapore, August 2009), Association for Computational Linguistics, pp. 1523–1532. 50

[113] Yang, Ming, Wu, Ying, Crenshaw, James, Augustine, Bruce, and Mareachen, Russell. Face detection for automatic exposure control in handheld camera. In *International Conference on Computer Vision Systems* (2006), vol. 0, p. 17. 6

[114] Yang, Ming-Hsuan, Kriegman, David J., and Ahuja, Narendra. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 1 (2002), 34–58. 27

[115] Yin, Robert K. Looking at upside-down faces. *J. Experimental Psychology 81* (1969), 141. 1

[116] Yue, Xiaomin, Tjan, Bosco S., and Biederman, Irving. What makes faces special? *Vision Research 46*, 22 (2006), 3802 – 3811. 1

[117] Zhang, Dong-Qing, and Chang, Shih-Fu. Detecting image near-duplicate by stochastic attributed relational graph matching with learning. In *ACM International Conference on Multimedia* (2004), pp. 877–884. 14

[118] Zhao, Wenyi, Chellappa, Rama, and Krishnaswamy, Arvindh. Discriminant analysis of principal components for face recognition. In *International Conference on Automatic Face and Gesture Recognition* (1998). 84, 89

[119] Zhou, Shaohua, Krueger, Volker, and Chellappa, Rama. Probabilistic recognition of human faces from video. *Computer Vision and Image Understanding 91*, 1-2 (2003), 214–245. 52

[120] Zhou, Shaohua Kevin, Chellappa, Rama, and Zhao, Wenyi. *Unconstrained Face Recognition*, vol. 5 of *International Series of Biometrics*. Springer, 2006. 52

[121] Zhu, Xiaojin. *Semi-Supervised Learning with Graphs*. PhD thesis, Carnegie Mellon University, 2005. 30, 49