Finding Celebrities in Video

Nazli Ikizler
Jai Vasanth
Linus Wong
David Forsyth

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2006-77
http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-77.html

May 23, 2006

Copyright © 2006, by the author(s).
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

Finding Celebrities in Videos

ABSTRACT

We present a system for finding celebrities in videos that uses face information in conjunction with text or speech. We achieve an approximate tripling of precision for searches over the use of transcripts or speech alone. Our work is motivated by the recent growth of personal video recording devices such as TiVo, which makes watching television more like information retrieval. We use a large dataset consisting of 13.5 hours of commercial video, which presents a challenging speech and face recognition environment. Faces are extracted using a face detector and processed via kernel PCA, LDA for use in one-vs-many SVM face classifiers. We evaluate two scenarios, one where transcripts are provided and the other more difficult scenario with speech as the only language cue. Wordspotting over audio is done using an HMM and SVM combination. We demonstrate our system's improved retrieval under realistic conditions using video recorded directly from television.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis; I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Experimentation

Keywords

Multimedia retrieval, video and image retrieval

1. INTRODUCTION

It is now common to digitize television and watch it later, but one must currently choose what to digitize. As disks get more capacious, physically smaller, and cheaper, it will become possible to digitize an entire days television and then choose what to watch. As a result, television viewing — a major leisure activity in the rich world — will involve significant information retrieval problems. There will be a demand for technologies that can identify "chunks" of the day's television that meet various search criteria. In this paper, we show how to obtain high precision searches for shots pertaining to celebrities by using straightforward vision, speech and natural language techniques. In particular, we show that checking whether a face is present in a shot can triple or better the precision of a search for a small cost in recall.

Background: It is now a commonplace that relations between visual data and other data provide useful information; only a sketchy review is possible in the available space. This observation has been exploited to provide **clustering** of collections of annotated pictures ([4] using methods due to [13, 14]; see also [17, 15, 21]); **browsable summaries** of collections of museum images with complex annotations (see [3]); **object labeling**, where one learns correspondence between individual image regions and individual nouns in an annotation, using a method analogous with those used to learn lexicons from an aligned bi-text (see [9] using methods due to [7, 19]; [2] compares a wide variety of different models); **automatic face dictionary**, which uses correspondence methods to link faces detected in news images with names extracted from captions ([28]; [6] shows that language cues improve the correspondence); **word spotting**, where ink patterns are linked to transcriptions ([24, 11]).

The process is more difficult for video. First, video transcripts are widely available for US source video because of legal closed captioning requirements, but may not be available for other video. Second, transcripts tend to be quite poorly aligned to video, meaning that correspondence is, at best, a rough cue (e.g. see [10]). Third, the spatial resolution of video is poor, meaning that tools such as face finders can become unreliable.

However, there are some advantages to working with video. First, there is a well-established commercial interest. For example, Google offers a service to search closed-caption transcripts (`http://video.google.com/`). Second, like searches of the web, this is a search regime in which precision is crucial, recall less important. Users would like the page(s) returned from a query for, say, Paula Abdul, to contain objects relevant to the query, but are unlikely to be concerned to obtain everything.

An initial attempt exploiting the relationship between names and faces is Satoh et al's Name-It system [25]. They use the co-occurrence statistics to associate names and faces. Chen

This paper was not, in fact, published by SIGIR-06

et al's method [8] is based on text retrieval and structure of the news videos.

Dataset: We demonstrate our system on a large challenging dataset that reflects a real-world scenario in which our system may be applied. We have recorded 25 episodes of the closed captioned celebrity news show "Entertainment Tonight" using a consumer-level television capture card, totalling approximately 13.5 hours of video including commercials. The embedded closed captions are fairly clean, with only occasional typos in names or short omissions of transcription. The captions thus provide a convenient aid in training and a way to assess our performance. Of the 25 episodes on hand, 12 are reserved for training our models, while the other 13 are used in testing. Details of the difficulties that this collection poses to audio and face recognition are given in the corresponding sections below.

Our system: We demonstrate searches for celebrities in this large collection of commercial video. We compare four methods:

- **Searching transcripts for names** using straightforward string matching methods.
- **Transcript names and video faces**, where one checks that a face corresponding to the name is nearby (using the methods of section 2). Notice that one cannot simply use exact correspondence, because transcripts are poorly aligned. However, it is sufficient to determine that a face lies within a time-window of the name. The fact that video is naturally arranged as shots (section 4.1) is a help here, too.
- **Audio names and video faces**, where a name is found in audio (using the methods of section 3) and one then checks that a face corresponding to that name is nearby.
- **Video faces**, where one simply searches for faces using a face recognizer.

By a long way, the most effective method is to find names in transcript and faces in video. This leads to a startling improvement in precision (often tripling or better) with relatively small loss of recall. Searching audio for names suffers from the difficulty that names are very hard to find in audio; as a result, performance of audio only searches on names is extremely poor, with very low precision even at quite low recall. However, if one is without a transcript and must search the audio, looking for a relevant face in nearby video will produce a usable, if not perfect, search.

2. DETECTING, REPRESENTING AND CLASSIFYING FACES

2.1 Face detection

We use Mikolajczyk's implementation of the face detector described by Schneiderman and Kanade [20]. The face detector works by decomposing the training images into a set of wavelet coefficients and binning them into a histogram. The probability of a new image being a face is the number of face images assigned compared to the number of non-face images assigned to its bin.

As shown in Fig 2, the face detector can fail to find faces especially when there is low resolution and/or occlusion caused by sunglasses or other objects. In some cases, it also fails to detect faces that are not frontal enough. This noticeably limits the recall rate of our approach. For instance, the face detector was able to locate only half of the faces of Angelina Jolie in one episode (a recall rate of 50%). Since we use only face detector outputs as the input of our approach, our recall is upper-bounded by the recall of the face detector in use. Furthermore, when the detector does spot profiles, the face recognizer performs poorly. Improvements in the face detector should increase the recall of our approach.

Figure 1: ***Representative shots from Entertainment Tonight dataset. Different poses, differing presentations, low resolution, celebrity style changes, sunglasses and other occlusions make this dataset extremely hard to recognize even for the human eye in some cases. Here you can see example frames of Renee Zellweger, Paula Abdul, Angelina Jolie, Tom Cruise, Katie Holmes and Oprah Winfrey***

2.2 Face representation

We do feature selection over the faces by first preprocessing for lighting variations, then performing kernel principal component analysis (kPCA) [26] and linear discriminant analysis (LDA). Kernel PCA uses a kernel function (radial basis function in our case) to efficiently compute a principal component basis in a high-dimensional feature space related to the input space by some nonlinear map. Although kPCA extracts nonlinear features that explain the variance in our dataset, these features are not all necessarily discriminative. Hence we also perform LDA over the kPCA-projected faces to incorporate class information. LDA finds a linear projection that simultaneously maximizes the distance between class means and minimizes class variance. See [5, 12] for examples of LDA's effectiveness in face discrimination tasks.

Our dataset contains many faces of the same person under varying illumination conditions, which may result in poor classification. We attempt to compensate for this prior to computing kPCA by performing histogram specification on each face as a form of illumination normalization. For each face, we fit the histogram of the log of the pixel intensities to that of our canonical face. The canonical face is selected as having a representative illumination of an ideal face. This step should reduce classifier dependencies on lighting and contrast variations for discrimination.

Given an input dataset, Kernel PCA is done as follows: Compute a kernel matrix K, where K_{ij} is the value obtained by feeding $Image_i$ and $Image_j$ into the kernel function. The kernel matrix is then centered in the feature space by subtracting off the mean row, mean column and adding the average element values. The top k normalized eigenvectors of K represent our new reduced dimension set.

Figure 2: ***Representative shots of Angelina Jolie where face detector fails to find the face.***

Direct kPCA is not possible given our training dataset, which comprises of 34445 extracted face detections. This makes the $N \times N$ kernel matrix too large to be stored and computed (around 10^9). Hence we use the Nyström approximation to compute an approximate value for K (justified by our kernel matrix having low rank; c.f [28, 6]). The Nyström approximation is computed as follows: Choose n face images (in our case 1000) at random (this requires care because faces tend to be heavily correlated in time in video). Next build the kernel matrix A by comparing the images with themselves and B by comparing it with all the rest of the images. K can be partitioned as

$$K = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \quad (1)$$

The Nyström approximation approximates C as $\hat{C} = B^T A^{-1} B$. This gives an approximation to K as

$$\hat{K} = \begin{pmatrix} A & B \\ B^T & \hat{C} \end{pmatrix} \quad (2)$$

We expect the number of large eigenvalues for K to be small as the effective column rank of K is very low when compared to its size. This is the main motivation to use the Nyström approximation, as we observe in our matrix that the eigenvalues of A tend to drop quickly since the effective rank of the whole matrix is small.

In order to perform LDA over these dimensionality-reduced faces, we need a set of classes on which to train. However, our large collection of extracted faces makes it infeasible for us to manually label and train on all faces. Instead, we only label a subset of faces in the training videos, forming an initial training set of positive examples for each class. For each class formed, we remove "duplicates" from its positive example set by simply eliminating any face whose Euclidean distance to the remaining falls below a small threshold. This lessens the bias in the LDA parameter estimates and training of the face classifiers (section 4). In this manner we construct 25 classes for the purpose of computing linear discriminants, consisting of 3309 faces in total.

Good estimation of the parameters in LDA is difficult when the number of examples is small. To improve our estimates, we augment the set of training faces with warped versions of each original face. We rotate each face clockwise and counter-clockwise by a small angle to artificially increase the number of training examples for each class.

The end result is a discriminative, low dimensional feature space in which our faces lie, suitable for efficiently training our face classifiers.

2.3 Face classification

Given a query for a particular celebrity, we wish to find all shots containing that person with his or her face, conditioned on the fact that the person's name was mentioned nearby (in audio or text). This is just a binary classification problem for each possible person we may wish to find. For this task we build 6 one-vs-all SVMs, one per person of interest. See [18] for a review of SVM use in face recognition.

The training set for each classifier is assembled as follows. Each person of interest already has a positive example set constructed earlier when computing linear discriminants (section 2.2). We use the same set of faces for training our SVMs. We construct each negative example set by randomly sampling 4500 faces from those remaining in a subset of the training episodes.

We train each celebrity SVM using the KPCA+LDA feature vectors of the faces corresponding to its training class. The RBF kernel was chosen for our SVMs, as they exhibited considerably better precision over linear SVMs. Parameter selection was done using a grid search with 5-fold cross validation.

3. NAME SPOTTING IN AUDIO

In the presence of close captioning, it is relatively easy to search through the video for a particular topic by querying text. However, in television programs, close captions may not always be available. In order to make those video documents searchable, aligned audio can be used.

For locating celebrity names in audio, we have implemented a keyword spotting system which works on the vocabulary of the names and/or surnames of the celebrities. Our dataset is small and very challenging compared to those used by existing speech recognition systems. There are varying types of background music, different and nonnative speakers and different acoustic environments, all of which makes the task of recognition considerably more difficult. For example, [23] gives low word accuracy rates(40%) for speech recognition systems in the presence of background music and low SNR (speech to noise ratio). Moreover, short names are harder to locate. It is shown in [16] that keyword spotting is dependent on the number of phones in the word.

Our keyword spotter proceeds in two phases. In the initial preprocessing phase, we significantly improve performance by eliminating the fragments of music from the audio signal. A music SVM is trained for this purpose. It eliminates the fragments of the audio where the music is significantly louder than the speech. The precision of the music classifier is 80%. Silence breaks are eliminated in a similar fashion with a silence model.

In order to spot the names in audio, 5 hours of training data are used to extract utterances of names, yielding an average of 10 instances per name. Acoustic feature extraction from the audio is done as follows: Using 25ms frames, with a frame shift of 10ms, 13 MFCC(Mel-Frequency Cepstral Coefficient) features (12 cepstra,1 energy function) are extracted for each frame using Sphinx 4 [30]. We used 5-state HMM models to represent each phone in our dictio-

Table 1: Number of false positives generated by the word spotter per hour of Entertainment Tonight Dataset. Results present that our name search is as good as a state-of-art keyword spotter.

Number of Phones	FP rate
3-phone	78
4-phone	74
5-phone	23
6-phone	20
7-phone	7
8-phone	4

nary[22]. 30 context-independent phones are trained, each phone state is modeled with 2 gaussian mixture pdfs. Each word HMM is then formed by the concatenation of those phone HMMs and one pass of Viterbi training for the whole keyword model. In the recognition phase, the HMM model outputs most probable audio sequences where a name may occur. However, since it is a generative model, it is more likely to produce a high number of false positives. To increase its performance, an SVM is trained for each name by using the true and false positives for that name. In Table 1 the performance of name spotter in audio is shown. As expected, longer names are easier to find. In [16], 100 false alarms per hour are reported for a four-phone word in TREC dataset [29], which is cleaner compared to our "Entertainment Tonight" dataset. Amir et al [1] states that in the presence of background noise, degraded acoustics and nonnative speakers in a real-time speech recognition system, error rates of 35% to 65% can be expected. The results show that our name spotter produces similar results to the keyword spotters described in [23] and [1].

4. RESULTS

Our dataset consists of 13.5 hours of "Entertainment Tonight", video recorded directly from television. This dataset poses a number of challenges to face recognition, not all seen in typical face recognition datasets. Resolution, expression, pose, and illumination of a single person's set of faces may all vary considerably. The resolution of detected faces can be quite low due to a variety of reasons, such as the low quality of the interlaced television recording, out-of-focus and skewed screens displaying a celebrity's face, or transitions between shots. Such transitions may also skew the face, change its intensity, or generally add noise to the faces. Furthermore, actors and actresses may appear in differing hair styles and colors over short periods of time. Sunglasses are also commonly worn among celebrities. Faces may also be partially occluded by other people. Fig 1 shows example frames reflecting the challenges in this dataset.

We compare four different models of retrieval. Our baseline model is a simple text search over closed captions (`Text` in the figures). The next model incorporates the face classifier with the closed captions (`Text+Face` in the figures). The third method combines the face and audio classifiers (`Audio+Face` in the figures). Finally, we can simply use the face recognizer (`Face` in the figures).

One cannot simply look for a face when the name is uttered, because the person may be looking away and because the transcripts tend to be seriously misaligned. Instead, we use a search window. Given a full name query, each method

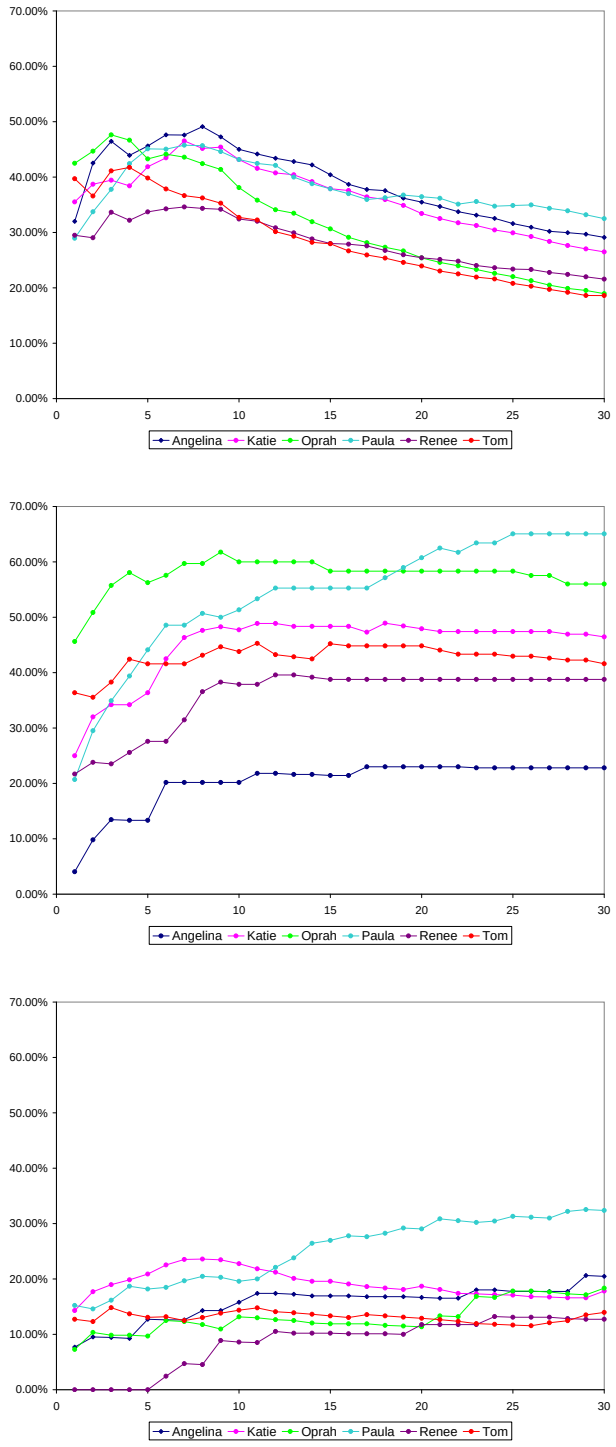

Figure 4: ***The curves show the F1 statistic (which is*** $2pr/(p+r)$ ***— the ideal is 1) plotted against window half-width for three methods for six different searches (Angelina Jolie, Katie Holmes, Oprah Winfrey, Paula Abdul, Renee Zellweger, Tom Cruise).*** **Top:** ***text only;*** **center:** ***text+face;*** **bottom:** ***audio+face. The plot is meaningless for a face recognizer alone. Notice that the statistic is affected by the window half-width, peaks around 10s, and falls off relatively slowly.***

Figure 3: ***Representative frames of shots for a query for "Paula Abdul" in one episode. The relevant shots are outlined in green and non-relevant shots are shown in red. Notice the visual richness of the clips.*** `Text` ***precision is very low compared to*** `Text+Face` ***method, and it can be seen that shots mostly do not include the face of the celebrity. Using our face classifier together with text/audio improves the precision significantly (c.f. Figure 6). Precision is more than tripled and almost all retrieved shots include the celebrity. Table 2 gives complete results. When word spotting on audio is used instead of text-based search, there is an improvement in precision in this episode, but the number of false positives is increased due to the challenges of locating the name in the audio.***

searches for all instances of the full name or last name or first name, estimating their time of utterance using the caption timing information or via audio classification. The transcript may be misaligned by up to 8 seconds. We retrieve all shots that intersect a window centered on this instant in time. We chose a ± 10 second window, as it accommodates the maximum delay we observed in caption timing information. Changing the width of the window leads to some change in precision and recall, as figure 4 indicates.

4.1 Shot Boundary Detection and Retrieval

Since we are working on videos, it does not make sense to return frames. A more suitable granularity for a video query system would be at the shot level. We built a simple shot detector that works on grayscale, segments every frame into 12 subimages (3×4) and histograms each of them using 15 grayscale bins. We then compute the mean squared distance between corresponding histograms in adjacent shots to get a 12 dimensioned vector (each dimension corresponding to the i^{th} histogram distance between adjacent frames). We include the 2 adjacent frames in a single shot if the scalar value of the distance vector is within a threshold. We verify visually the good performance of our shot detector, noting the consistency among frames within each shot, while avoiding oversegmentation of long shots.

We define a shot to be relevant towards a query if it is possible to manually identify the person using only their face, without additional context (such as the co-occurrence of a co-star, or a mention of their name). Therefore shots containing the query person whose face is heavily occluded, too small to differentiate, or turned away from the camera would all be considered not relevant.

The text method returns the entire set of shots lying within the window as relevant. The `Text+Face` method adds an additional filtering step, in which we classify all the face detections within these shots as positive or not with respect to the query. We return only those shots that contain at least one positive face classification, conditioned on finding a name instance nearby. The audio+face method uses the same ± 10 second window centered on name instances, but finds the names based on the audio classifier output, rather than captions. The shots are subsequently filtered using a face classifier, again returning only those with at least one positive face. This presents the more realistic scenario when transcripts are not available. Finally, the face method identifies shots containing a face recognized as the celebrity.

4.2 Retrieval results

Each retrieval model's performance is measured over 13 of the 25 recorded episodes. The 13 test set episodes are comprised of 75516 frames (sampled 3x per second) containing 38408 face detections. 7301 shots are identified in this pool of frames. Ground truth relevancy of each shot was generated manually for the 5 celebrities.

Current video retrieval methods available often rely on a search based around text or meta-data, which can result in a high portion of irrelevant shots presented to the user. In many applications, however, precision tends to be far more important than returning hundreds or more results, since users normally do not inspect all of them. Furthermore, should a user be searching for a particular person, the user may wish to actually see the person in question. In such

Figure 5: *Representative frames of shots for a query for "Oprah Winfrey" in a single episode. The relevant shots are outlined in green and non-relevant shots are shown in red. Using our face classifier together with text/audio improves the precision significantly (c.f. figure 6). Precision is more than tripled and almost all retrieved shots include the celebrity. When word spotting on audio is used instead of text-based search, there is an improvement in precision, but the number of false positives is increased due to the challenges of locating the name in the audio.*

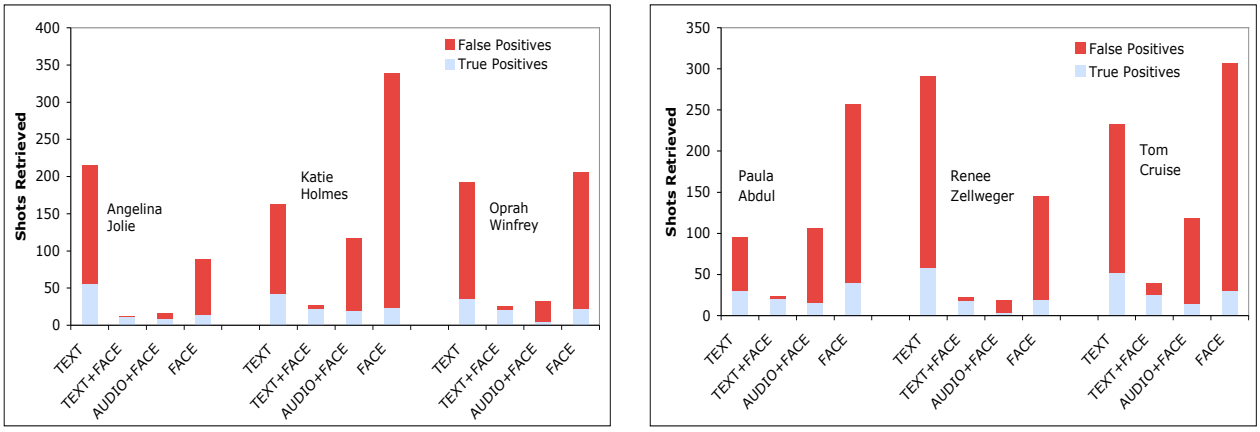

Figure 6: *Number of shots retrieved by each method. The ratio of the lighter bar to the whole bar represents the precision rate. Notice that the precision is more than tripled when our face classifier is used. It can be seen that text querying introduces too many false positives whereas false positives are almost completely eliminated by the* `text+face` *method, increasing the precision to a great extent. Note that the recall of* `text+face` *and* `audio+face` *method is effected by the recall of the face detector.*

cases, precision is far more important than recall. Experimental results below show our system to be well suited to such applications, achieving a tripling of precision or more over a baseline text search.

In figures 3 and 5, shots retrieved by each method are shown. Here we see that the baseline (text-only) method retrieves shots that do not include the celebrity most of the time.

5. DISCUSSION

While face recognition is not reliable in difficult commercial video data, and name spotting is not a great cue, the combination yields a high-precision search. This effect extends to word spotting in audio, which, while not the method of choice for finding names, may be necessary when one has video without transcripts. This is almost certainly because the errors that each method makes are not correlated. A second important phenomenon is that the shot — the natural quantum of video retrieval — allows considerable room for error. For example, for our face method to fail, the face must be either not be detected or misclassified for the *whole shot*; this is of considerable help, given the current state of face recognition on commercial video data. Furthermore, this effect means that the method of Sivic *et al.* for linking faces across profile views in video [27] is unlikely to provide a significant improvement in performance. Future work will include investigating the effects of improvements in face detection and face recognition.

6. REFERENCES

[1] A. Amir, S. Srinivasan, and A. Efrat. Search the audio, browse the video-a generic paradigm for video collections. *EURASIP Journal on Applied Signal Processing*, 2(1):209–222, 2003.

[2] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

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

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

[5] P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *PAMI Special issue on face recognition*, pages 711–720, July 1997.

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

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

[8] M. Chen and A. Hauptmann. Searching for a specific person in broadcast news video. In *IEEE Conference on Acoustics, Speech and Signal Processing*, May 2004.

[9] P. Duygulu, K. Barnard, N. de Freitas, and D. Forsyth. Object recognition as machine translation. In *Proc. European Conference on Computer Vision*, pages IV: 97–112, 2002.

[10] P. Duygulu, J.-Y. Pan, and D. A. Forsyth. Towards auto-documentary: tracking the evolution of news stories. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 820–827, New York, NY, USA, 2004. ACM Press.

[11] J. Edwards, Y. W. Teh, D. Forsyth, R. Bock, M. Maire, and G. Vesom. Making mediaeval latin manuscripts searchable using ghmm's. In *Proc. NIPS*, 2004.

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

[13] T. Hofmann. Learning and representing topic. a hierarchical mixture model for word occurrence in document databases. In *Workshop on learning from text and the web*, CMU, 1998.

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

[15] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using crossmedia relevance models. In *SIGIR*, 2003.

[16] D. A. V. Leeuwen, W. Kraaij, and R. Ekkelenkamp. Prediction of keyword spotting performance based on phonemic contents. In *ESCA,ERTW Workshop:Accessing Information in Spoken Audio*, 1999.

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

[18] S. Li and A. Jain, editors. *Handbook of Face Recognition*. Springer, 2005.

[19] I. D. Melamed. *Empirical Methods for Exploiting Parallel Texts*. MIT Press, 2001.

[20] K. Mikolajczyk. Face detector. Technical report, INRIA Rhone-Alpes. Ph.D report.

[21] J.-Y. Pan, H.-J. Yang, C. Faloutsos, and P. Duygulu. Automatic multimedia cross-modal correlation discovery. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 653–658, New York, NY, USA, 2004. ACM Press.

[22] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[23] B. Raj, V. N. Parikh, and R. M. Stern. The effects of background music on speech recognition accuracy. In *IEEE Conference on Acoustics, Speech and Signal Processing*, 1997.

[24] T. M. Rath and R. Manmatha. Word image matching using dynamic time warping. In *Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 521–527, 2003.

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

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

Table 2: ***Precision and recall values for all results. Notice the increased precision by using text with face. We do not have the formal estimates of precision for audio method, but one can expect from the inevitable high false positive rate that the precision is not strongly different from zero.***

	Text		*Face*		*Text+face*		*Audio+face*	
	P	R	P	R	P	R	P	R
Angelina Jolie	26.05%	57.73%	16.85%	15.46%	91.67%	11.64%	52.94%	9.28%
Katie Holmes	25.77%	68.85%	7.05%	39.34%	81.48%	36.07%	16.95%	32.79%
Oprah Winfrey	18.75%	83.72%	10.73%	51.16%	80.77%	48.84%	15.63%	11.63%
Paula Abdul	31.58%	57.69%	15.56%	76.92%	87.50%	40.39%	14.95%	30.77%
Renee Zellweger	19.93%	79.45%	13.1%	26.03%	81.82%	24.66%	21.05%	5.48%
Tom Cruise	22.32%	76.47%	9.77%	44.12%	64.10%	36.77%	11.61%	22.06%
Overall	24.06%	70.65%	12.18%	42.17%	81.22%	33.06%	22.19%	18.67%

[27] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *International Conference on Image and Video Retrieval (CIVR 2005), Singapore*, 2005.

[28] T.Miller, A. Berg, J. Edwards, M.Maire, R. White, Y.-W. Teh, E. Miller, and D. Forsyth. Faces and names in the news. In *Proc IEEE Conf. on Computer Vision and Pattern Recognition, 2004.*, 2004.

[29] E. Voorhees and D. Harman, editors. *The seventh Text REtreival Conference*, 1998.

[30] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel. Sphinx4: A flexible open source framework for speech recognition. Technical report, Sun Microsystems Inc., 2004.