Large Scale Learning and Recognition of Faces in Web Videos

Ming Zhao, Jay Yagnik, Hartwig Adam, David Bau
Google Inc.
{mingzhao,jyagnik,hadam,davidbau}@google.com

Abstract

The phenomenal growth of video on the web and the increasing sparseness of meta information associated with it forces us to look for signals from the video content for search/information retrieval and browsing based corpus exploration. A large chunk of users' searching/browsing patterns are centered around people present in the video. Doing it at scale in videos remains hard due to a) the absence of labeled data for such a large set of people and b) the large variation of pose/illumination/expression/age/occlusion/quality etc in the target corpus.

We propose a system that can learn and recognize faces by combining signals from large scale weakly labeled text, image, and video corpora. First, consistency learning is proposed to create face models for popular persons. We use the text-image co-occurrence on the web as a weak signal of relevance and learn the set of consistent face models from this very large and noisy training set. Second, efficient and accurate face detection and face tracking is applied. Last, the key faces in each face track is select by clustering to get compact and robust representation. The face tracks are further clustered to get more representative key faces and remove duplicate key faces. For each cluster of face tracks, a combination of majority voting and probabilistic voting is done with the automatically learned models. The effectiveness of our framework is demonstrated by results on image and video corpora, in which we can achieve 92.68% in 37 million images and 80% top-5-precision in 1500 hours videos.

1. Introduction

Increasing amount of video content on the web is marking a new phase of how users consume information. Users often look for specific people-related video content. The need for recognizing people in these videos is tremendous. However, this large scale face recognition task presents several challenges: (1) it is not practical to manually label training data for a large set of people du to very large variation of pose/illumination/expression/age/occlusion/quality etc; (2) it is very important to develop efficient, high precision, and robust algorithms for recognizing faces.

We address the first challenge by correlating people names with faces directly from the web using people name appearances as a weak prior to finding the right face models. Consistency learning and correlation sampling are proposed to build a set of accurate and clean models from the weak and noisy text-image correlation data. As the relative ratios of text to image content on web pages change, it becomes increasingly harder for text based retrieval systems to find the most relevant image/video. However, the large size of this corpus presents a very unique opportunity, i.e using text based priors to bias the learning of image based models. Systems that rely on very complex models can often be outperformed by very simple systems with statistics estimated from a very large dataset. So, we proposed consistency learning to automatically learn the models from this weakly labeled corpus of text-image co-occurrence.

[1, 2] addresses this issues by using Google Image Search to find relevant images and using TSI-pLSA based method to learn relevant models. They deal with the problem of general object categories while we deal with faces in particular. One difference is that they pick the best model for each category while we retain a large number of models spanning different variations in the face space which is key to robust recognition invariant to different variations. [5] deals with a very similar problem for news broadcasts. They extract discriminant coordinates for the faces and apply a clustering step to estimate the likely labels. As their dataset is constrained in the news pictures, it's much smaller and cleaner than whole web images, and the variations of faces are also limited by the nature of news pictures. The case of "structured noise", i.e noise occurring as a cluster, isn't handled by any of these methods. Other related work for faces can be found in [6, 7] while the idea of using text and image features for learning has be explored in [8, 9]. [10] wants to build a comprehensive face data set with lowest manual effort.

Consistency learning uses completely discriminative estimates in measuring goodness and hence under assump-

978-1-4244-2154-1/08/$25.00 2008 IE

tions of completeness in the people name set (not unrealistic for celebrities) we can handle noisy results even when they form clusters. We first outline the idea of consistency learning and later present a correlation sampling scheme to handle correlation bias in web image results. We learn from a very large dataset of 37 million images and end up with 200 thousand face models to represent 6 thousand people.

Real world video face recognition is getting more interesting [11, 13, 14]. To meet the second challenge, we developed a very efficient, highly precise and robust algorithm for video face recognition. A very good face detectors are available with the success of efficient and accurate face detection algorithm following Viola and Jones [12]. We developed a highly accurate and efficient face tracker which tracks facial feature points. Key face selection and track clustering is performed for efficient and robust video face representation. And finally, a combination of majority voting and probabilistic voting is proposed for exemplar based video face recognition.

The rest of this paper is organized as follows: The system overview is outlined in section 2. Consistency learning is presented in section 3. Video face recognition is described in section 4. Experimental results are presented in section 5, and conclusion and future work is presented in section 6.

2. System Overview

As shown in Figure 1, the whole system is composed of two phases:

1. Automatic Model Learning:

   (1) We use a person name entity detection algorithm [15] to extract candidates names. We use the total number of occurrence of each name on the web as a measure of popularity and select the top 6000 candidate names. From this set, we have manually eliminated a small number of false positives in the person name entity detection process. Working with the top names also helps us get over errors in the name entity detection algorithm. Hence, we refer to these names as celebrity names.

   (2) We then extract Google image search results for all the celebrity names.

   (3) Faces are detected from the images, and a feature vector is extracted for each face.

   (4) Consistency learning is applied to the resulting dataset, and celebrity models are built.

2. Video Face Recognition:

   (1) For each web video, we perform face detection and tracking.

   (2) For each track, some key faces are selected to represent this track by face clustering.

   (3) We perform track clustering to form very tight clusters such that each cluster represents only one person (a person could have multiple clusters).

   (4) For each track-cluster, video face recognition is performed with the automatically learned models from phase 1.

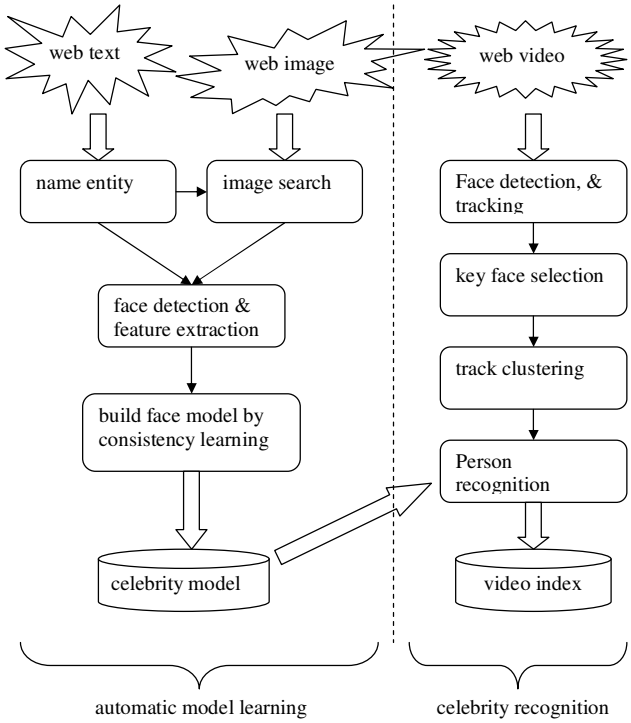

Figure 1. system overview

3. Automatic Learning of Face Models

3.1. Consistency Learning

Precision and generalization of a machine learning system depends heavily on the amount of training data it is presented with. Training data is often hand annotated and hence curation of large datasets becomes expensive. Most learning formulations rely heavily on the training data being noise-free and hence need this high accuracy data curation process. In the recent years, world wide web has grown to cover all kinds of information. With some prior filtering we can find data that is weakly relevant to the problem at hand. We present a method that operates on a large weak-hypothesis set to generates a very clean dataset. It simultaneously approximates the classifier surface with a piecewise linear fit by the nearest neighbors on the filtered set.

The basic idea is to have a smoothness prior over the concept space in determining if a label is accurate or noisy. In

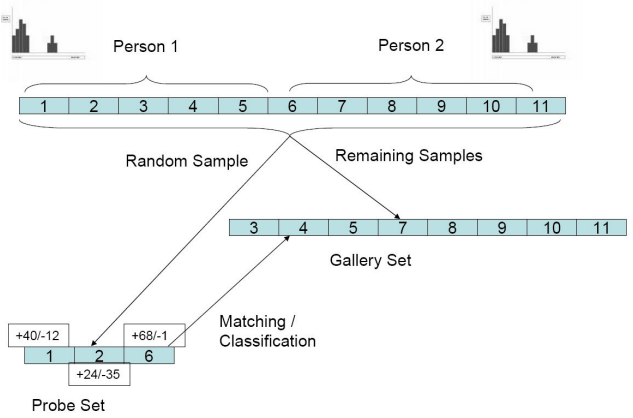

Figure 2. Illustrative diagram for self validation.

particular we propose to use matching behavior over large sets as a discriminative estimator for consistency. Figure 2 illustrates the proposed system for two classes (with a trivial generalization to multi-class formulation) . We have a set of hypothesis classes, with each class containing a set of weakly confident hypothesis. Since no explicit ground truth is given, we term this method as self validation. The goal of self validation is to come up with a posterior for confidence on the hypothesis space using the information about other hypothesis in the space. Algorithm 1 outlines a scheme to estimate consistency based on a discriminative approach. Key insight in the method is the observation that samples that confirm to classifiers learned from an unbiased set of weak hypothesis have a higher probability of correctness. We assume here that the optimal Bayes discriminative surface is within the representative ability of the classifier and hence attribute confirming classification to the smoothness of the hypothesis space. This results in a very intuitive estimation scheme. We randomly sample a probe set from the weak-hypothesis set and then assume the remaining to form the gallery of true samples. Every time a probe hypothesis matches its hypothesis class in the gallery our confidence in its trueness increases. If this sampling operation is done often enough then we can compute a statistically significant number for each sample which we call precision.

3.2. Correlation Sampling

Most machine learning algorithms assume that the input data is IID. The IID assumption is particularly violated in the current context since images on the web and in particular celebrity images exhibit a lot of correlation. There are edit/duplication paths that the original images go through to find multiple appearances on the web.

A graph based sampling algorithm is proposed to sample data either to favor correlation or to maximize decorrelation. Let's assume a weighted graph $G(V, E)$ which represents connectivity of the input data. We wish to sample a subset of nodes $S \subset G$ from this graph such that the quantity $F(S)$ is minimized.

1. Randomly sample a probe set from the result-set of all faces (i.e the weak hypothesis set).
2. Train a classifier from the remaining samples (could be k-NN).
3. Classify the probe samples.
4. If samples match the hypothesis class, consider it a positive vote, else a negative vote.
5. Repeat steps 1-4 a large number of times.
6. For each sample compute precision as

$$Precision = \frac{positive}{positive + negative}$$

Algorithm 1: Basic algorithm to determine consistency based on discriminative precision.

- Sample two nodes $v_1$and v_2 uniformly at random.
- Compute their proximity functions $F(S, v_1)$ and $F(S, v_2)$.
- Emit the smaller one of the two as the desired sample.

Algorithm 2: Rank proportional sampling

$$F(S) = \sum_{v_1 \in S} \sum_{v_2 \in S} f(v_1, v_2) .R(\mid S \mid) \tag{1}$$

where $f(v_1, v_2)$ is the proximity function we wish to optimize on and $R(\mid S \mid)$ is a regularization term to control the size of S. $F(S)$ can be optimized in a stochastic and greedy manner in following steps:

$$F^{'}(S, v) = \sum_{v^{'} \in S} f\left(v, v^{'}\right) \tag{2}$$

$$P_{(S,G)}(v) = \frac{F^{'}(S, v)}{\sum_{v \in G} F^{'}(S, v)} \tag{3}$$

$$v_{good} \leftarrow P_{(S,G)}(v) \tag{4}$$

$$S = S \cup \{v_{good}\} \tag{5}$$

3.3. The sampling algorithm

The current formulation requires $O(n)$ computation to sample one point due to the normalization step. We further approximate this by defining a distribution $P_{rank_{(S,G)}}(v)$ which samples v proportional to its rank in the list of

nodes sorted by the proximity function $F^{'}(S, v)$, in which the smaller $F^{'}(S, v)$ is, the higher rank v has. By introducing the notation of rank proportional sampling we get around the need to normalize the distribution and hence avoid $O(n)$computation. In fact this particular formulation of the problem leads us to a surprisingly simple algorithm, shown in Algorithm 2, which can be proven to yield a ramp distribution on the rank of the numbers. Let's consider the probability distribution, $P_{rank}(r)$, of the number being emitted in rank r (the smaller $F^{'}(S, v)$ is, the larger r is):

$$P_{rank}(r) = P(r)P(r^{'} \leq r) = \frac{1}{n} \cdot \frac{r}{n} = \frac{r}{n^2} \qquad (6)$$

Hence the probability of sampling is proportional to the rank of the node in the list sorted by the proximity function.

Its worth noting here that the technique can work with any proximity function. In particular using min or max instead of the summation are attractive alternatives in that they give a measure proportional to a smooth version of similarity/distance. All results presented here will use the min proximity function. It should also be noted that since we got rid of the normalization term, the entire distribution need not be pre-computed and we can do away with just 2 comparisons per sampling step.

4. Video Face Recognition

To recognize people in videos, face detection and tracking are applied to extract faces from videos. Then, key faces are selected for each track for fast and robust recognition. Face tracks are further clustered to get more compact and robust representation. A combination of majority voting and probabilistic voting algorithm is used to recognize each cluster of face tracks.

4.1. Face detection and Tracking

We use frontal face detection to locate the faces. Our face detection is based an extension of a cascade of boosted classifiers of Viola's work [12].

For face tracking, besides computational efficiency, high precision is preferred by us as it is of high importance for retrieval. To this end, we build on Google's proprietary Neven Vision facial feature tracking SDK [17], whose tracker component supports real-time and off-line tracking of up to 22 facial feature points in video streams. The tracker uses Gabor wavelet-based features to represent the facial landmarks and is highly robust against face size and pose variations as well as illumination changes. A very useful feature of the tracking engine is a person-independent classifier which produces a tracker confidence score, which can be used as a reliable indicator whether the system is still on-track or in need to be reinitialized. This is very important for recognition, since face recognition performance can be severely impaired, if the face tracks become cluttered due to the tracker drifting off to other faces or objects.

The whole face detection and tracking algorithm is shown in Figure 3. Shot boundary detection is used to prevent tracking across different shots, and periodical face detection is performed to detect new faces. Each time when we track faces, we also check the confidence of the facial feature tracker. If the facial feature tracker fails, we will try to give it another chance by using face detection to confirm it. By this way, we can reliably track faces.

4.2. Key Face Selection

After face detection and tracking, the faces are represented by face tracks where each track is a sequence of faces of the (ideally) same person in consecutive frames. The intuitive way is to recognize all the faces of each track. However, there are two disadvantages here: (1) It is time consuming; (2) It does not get rid of the redundant information; (3) It does not remove noise because some of the frames are not good due to low quality of compression, occlusion, pose, illumination, and expression.

In fact, speed is very important for large scale face recognition, and the many faces are not good for recognition in the real-world video and the quality of many web videos are not good. So, it is very important to select good key faces for recognition. A clustering-based key face selection algorithm is used in this paper. Some papers used k-means clustering. However, as we do not know the number of clusters in a track, we used the hierarchical clustering algorithm to cluster the faces of a track. To cluster the faces into separate clusters, complete-linkage is used. The distance between two faces are based on the selected local Gabor features extracted from facial feature points. After the clustering, each cluster will consist of different faces according to pose, occlusion, and quality. To remove noise, we discard the clusters with smaller number of faces. Also, as face recognition does not work well with non-frontal faces, we also remove the clusters which are non-frontal faces. The pose estimation is based on the facial feature points. A PCA model is built for some training samples of facial feature points, of which the first component corresponds to the left-right face rotation. Then, the facial feature points of new faces are projected into the PCA model and its pose is estimated by the parameter of the first PCA component.

4.3. Face Track Clustering

It is usually the situation that one person can appear several times inside a video. If we know which faces belong to the same person, then they can be recognized as a whole. There will be two benefits: (1) We can save the computation as there could be duplicate faces in different tracks and we do not need to recognize them one-by-one; (2) It will

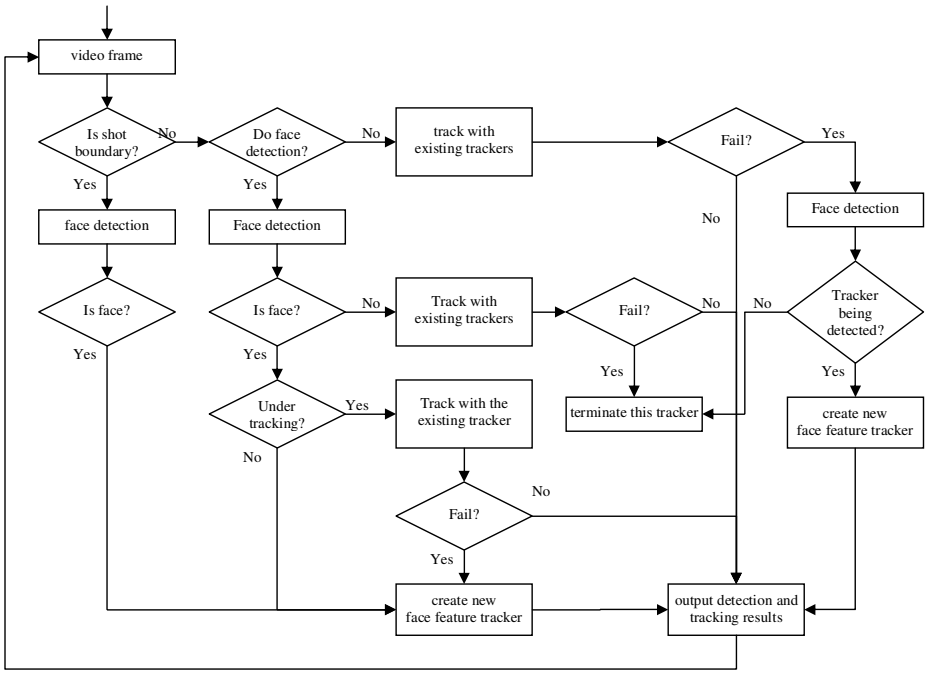

Figure 3. face detection and tracking

be more robust for recognition as we can see more samples of the same person. So, it is necessary to do clustering of the face tracks. Also, the hierarchical clustering algorithm is used. The similarity between two track is the maximum similarity of the key faces. After the tracks are clustered, we do further clustering on the key faces of all the tracks in the same cluster, and select the key faces of this cluster. In this way, we can get more representative key faces and remove unnecessary key faces.

4.4. Track Cluster Recognition

As the models are built from images, an exemplar-based video face recognition method is used in this paper where each cluster of face tracks is represented by some key faces.

There are two popular recognition algorithms for exemplar based recognition [16]. One is the majority voting algorithm which recognizes the face in every exemplar and then the identity of the person in the sequence is the one who is recognized with the most times. The other is the probabilistic voting algorithm which combines the recognition confidences in every exemplar and the identity of the person in the sequence is the one who has the most summed recognition confidence.

To get highly accurate recognition results, our recognition algorithm is a combination of the majority voting and probabilistic voting because we have a special person called "unknown", which means it is not any one of the persons in our face model. Let's suppose the key faces of a cluster of face tracks is:

$$KF = \{f_1, f_2, ..., f_N\} \tag{7}$$

First, each key face f_i is recognized by k-nearest neighbor as person $p(f_i)$ with confidence $c(f_i)$. Then, for each person in all the recognized persons $p_j \in \{p(f_i)\}$, we can calculate the times $N(p_j)$ that the key faces are recognized as p_j, i.e.

$$N(p_j) = \sum_{i=0}^{i=N} \delta(p(f_i), p_j) \tag{8}$$

where $\delta(x, y)$ is an indicator function which is 1 when its two arguments are the same, and 0 otherwise. And the average recognition confidence of p_j is $\bar{C}(p_j)$, i.e.

$$\bar{C}(p_j) = \frac{1}{N} \sum_{i=0}^{i=N} \delta(p(f_i), p_j) * c(f_i) \tag{9}$$

Then, the person with the maximum $N(p_j)$ and $\bar{C}(p_j)$ is recognized as the identity of this cluster. A classifier is also built on $\max(N(p_j))$ and $\max(\bar{C}(p_j))$ to decide whether recognition is confident enough or not, otherwise the identity of this cluster is unknown.

5. Experiments and Results

5.1. Automatic Model Learning

We present results of the proposed method evaluated on a subset of web pages amounting to about 37 million images

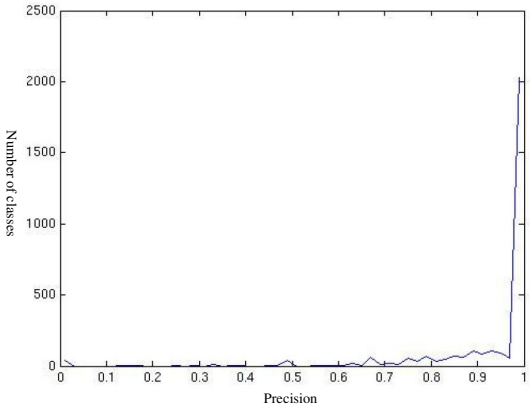

Figure 4. Distribution of Average Precision for the 6000 classes

(pure image data is used here, no snapshots from video). We use weak text correlations to get a hypothesis set for a person name. About six thousand of these are put together to run consistency determination experiments as outlined above. Towards the end we retain a set of 200k models for the 6k hypothesis classes put together. Detailed experimental setup is as follows :

- Collect a set of hypothesis images for each person name using text correlations from web pages.
- Perform face detection on each image.
- For images with faces, extract a face signature. Our face recognition system of choice is the Neven Vision Face Recognition SDK, which was one of the top performing engines in the most recent Face Recognition Vendor Test in 2006 [4]. The face recognition engine represents faces using local features based on the responses of a Gabor wavelet transform and has its roots in the Elastic Graph Matching approach.
- Run consistency learning experiment on the set of all hypothesis for all classes, using Nearest Neighbor classifier and face signature similarity as the similarity metric.
- Prune samples based on precision histogram at $\mu - 2\sigma$.
- Retain high confidence models for recognition.

Figure 4 shows the distribution of the average precision over the 6k classes. It is evident that we can find good models for the majority of them with accuracy larger than 96%. The mean classification rate for all the classes is 92.68% in a forced choice mode.

5.2. Video Face Recognition

Our testing set contains about 1500 hours videos randomly selected from web crawling and user uploading, and contains 44,453 face tracks. Video face recognition is performed with about 2k accurate models selected from the learned 6k models. We set up three experiments:

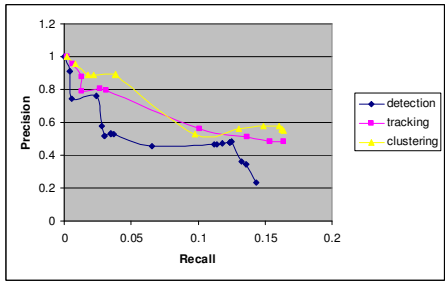

Figure 5. The precision recall distribution of difference schemes

- detection-based recognition: do face detection periodically for each shot, and recognize the detected faces. This is the base line.
- tracking based recognition: recognition on the face tracks with key frame selection.
- clustering based recognition: cluster face tracks, and recognize them.

We manually check the average precision of 20 persons. The results for are shown in Table 1. Please note that, the precision is not high because some people do not have enough videos. For example, Jacques Chirac does not appear in any video, so, it's precision and recall are zero for all of them.

To get the precision and recall distribution, we also manually label about 10 hours of videos. They are selected to contain 10 persons: George Bush, Condoleezza Rice, Tony Blair, Bill Clinton, Kofi Annan, Sanddam Hussein, Bill Gates, Steve Nash, Adolf Hitler, and Bin Laden. The precision and recall curve for three experiments are shown are shown in Figure 5. We can see that although the precision is pretty good, the recall is still low. This is because there are lots of faces which are non-frontal, and/or small. In such cases our face recognition engine's accuracy starts to decline significantly.

Name	top 5	top 10	top 20
Clustering	0.80	0.63	0.5
Tracking	0.77	0.60	0.44
Detection	0.70	0.43	0.37

Table 1. precision of top results

In our experiments, we also find that our algorithm is very robust for many sorts of variations including: pose (Figure 6), illumination (Figure 7), expression (Figure 8), age(Figure 9), occlusion (Figure 10), and quality (Figure 11). The recognized names are shown on the images in Figure 6-11, in which all the faces are correctly recognized even presented with these variations.

Figure 6. pose variations

Figure 7. illumination variations

Figure 8. expression variations

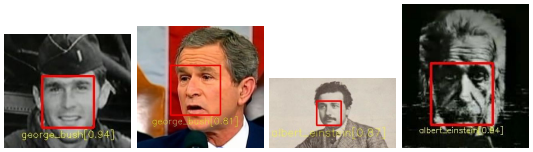

Figure 9. age variations

Figure 10. occlusion variations

Figure 11. quality variations

6. Conclusion and Future Work

A system for large scale recognition of faces in web videos are proposed in this paper. The face models of celebrities are automatically learned from the web images by consistency learning. The names of celebrities are mined from the web text by name entity detection. A highly accurate and efficient face detection and tracking algorithm is applied to extract faces. Key face selection and face track clustering is performed to get fast and robust recognition. A combination of majority voting and probabilistic voting is used to get high precise recognition. The experiments shows that this method can produce results with high precision (80% top-5-precision), which is preferred for retrieval.

There are amazing active learning possibilities in improving the recognizer to grow across age variations while incrementally adding high confidence samples to the training set. Another direction would be how to combine high-precision face-based retrieval and high-recall text-based retrieval.

References

[1] Learning object categories from Google's image search. Fergus R, Fei-Fei L, Perona P, Zisserman A. ICCV 2005. 1

[2] A visual category filter for Google images. R Fergus, P Perona, A Zisserman - Proc. ECCV, 2004. 1

[3] Face Recognition: A Literature Survey W Zhao, R Chellappa, A Rosenfeld, P Phillips . ACM Computing Surveys, 2003.

[4] FRVT 2006 and ICE 2006 Large-Scale Results. P. Jonathon Phillips, W. Todd Scruggs, Alice J. O'Toole, Patrick J. Flynn, Kevin W. Bowyer, Cathy L. Schott, Matthew Sharpe. www.frvt.org. 6

[5] Names and faces in the news. Tamara L. Berg, Alexander C. Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller and D.A. Forsyth. CVPR 2004. 1

[6] On Affine Invariant Clustering and Automatic Cast Listing in Movies. A.W. Fitzgibbon, A. Zisserman. European Conference on Computer Vision, 2002 1

[7] Finding Faces in Cluttered Scenes using Random Labelled Graph Matching. T. Leung, M.C. Burl, and P. Perona. Int. Conf Computer Vision, 1995. 1

[8] Matching Words and Pictures. K. Barnard,P. Duygulu, N. de Freitas, D.A. Forsyth, D. Blei, M.I. Jordan. Journal of Machine Learning Research, Vol 3, pp. 1107-1135, 2003. 1

[9] Words and Pictures in the News. J. Edwards, R. White, D.A. Forsyth. Workshop on Learning Word Meaning from Non-Linguistic Data, 2003. 1

[10] D. Ramanan, S. Baker, and S. Kakade, Leveraging archival video for building face datasets, Int'l Conf. on Computer Vision, Oct. 2007. 1

[11] J. Stallkamp, H.K. Ekenel, and R. Stiefelhagen, Video-based Face Recognition on Real-World Data International Conference on Computer Vision (ICCV'07), Rio de Jenario, Brasil, October 2007. 2

[12] P. Viola and M. Jones. Robust real time object detection. In *IEEE ICCV Workshop on Statistical and Computational Theories of Vision*, Vancouver, Canada, July 13 2001. 2, 4

[13] M. Everingham, J. Sivic, and A. Zisserman, "Hello! my name is... buffy - automatic naming of characters in tv video," in Proceedings of the British Machine Vision Conference, 2006. 2

[14] Sivic, J., Everingham, M. and Zisserman, A. Person spotting: video shot retrieval for face sets International Conference on Image and Video Retrieval (CIVR 2005). 2

[15] Dou Shen and Toby Walkery and Zijian Zhengy and Qiang Yangz and Ying Li, Personal name classification in web queries, WSDM '08: Proceedings of the international conference on Web search and web data mining, 2008, 149–158. 2

[16] A. Hadid and M. Pietikainen, "From still image to video-based face recognition: an experimental analysis", IEEE International Conference on Automatic Face and Gesture Recognition, pages:813–818, 2004. 5

[17] T. Maurer et al, "wavelet-based facial motion capture for avatar animation", U.S. Patent 6272231. 4