ViFaI: A trained video face indexing scheme

Harsh Nayyar
hnayyar@stanford.edu

Audrey Wei
awei1001@stanford.edu

1. Introduction

With the increasing prominence of inexpensive video recording devices (e.g., digital camcorders and video recording smartphones), the average user's video collection today is increasing rapidly. With this development, there arises a natural desire to rapidly access a subset of one's collection of videos. The solution to this problem requires an effective video indexing scheme. In particular, we must be able to easily process a video to extract such indexes.

Today, there also exist large sets of labeled (tagged) face images. One important example is an individual's Facebook profile. Such a set of of tagged images of one's self, family, friends, and colleagues represents an extremely valuable potential training set.

In this work, we explore how to leverage the aforementioned training set to solve the video indexing problem.

2. Problem Statement

Use a labeled (tagged) training set of face images to extract relevant indexes from a collection of videos, and use these indexes to answer boolean queries of the form: "videos with 'Person 1' OP1 'Person 2' OP2 ... OP(N-1) 'Person N' ", where 'Person N' corresponds to a training label (tag) and OPN is a boolean operand such as AND, OR, NOT, XOR, and so on.

3. Proposed Scheme

In this section, we outline our proposed scheme to address the problem we postulate in the previous section. We provide further details about the system implementation in Section 4.

At a high level, we subdivide the problem into two key phases: the first "off-line" executed once, and the second "on-line" phase instantiated upon each query.

For the purposes of this work, we define an *index* as follows: <video id, tag, frame #>.

3.1. The training phase

We first outline Phase 1 (the training or "off-line" phase):

1. Use the labeled training set plus an additional set of 'other' faces to compute the Fisher Linear Discriminant (FLD) [1].
2. Project the training data onto the space defined by the eigenvectors returned by the FLD, and train a classifier (first nearest neighbour, then SVM if required) using the training features.
3. Iterate through each frame of each video, detecting faces [2], classifying detected results, and add an index if the detected face corresponds to one of the labeled classes from the previous step.

3.2. The query phase

Now, we outline Phase 2 (the query or "on-line" phase):

1. Key the indexes on their video id.
2. For each video, evaluate the boolean query for the set of corresponding indexes.
3. Keep videos for which the boolean query evaluates true, and discard those for which it evaluates false.

4. Implementation Details

We are implementing the project in C++, leveraging the OpenCV v2.2 framework [4]. In this section, we will highlight some of the critical implementation details of our proposed system.

4.1. Training set acquisition

In order to obtain the training faces, we must extract faces from tagged Facebook data. This requires parsing through Facebook's Graph API. Essentially, for our purposes we have images, each with a set of tags. We first retrieve these images (using the wget utility), and then proceed to extract the tagged faces.

At this stage, we perform some outlier removal. Specifically, we run the OpenCV Viola-Jones based face detector in order to detect faces. If the detected faces correspond to tags, we accept the face. In our system, we reject regions with tags but no detected faces.

During the acquisition phase, we also resize all training samples to a standard size of 100px by 100px, and store the grayscale representation of the samples.

4.2. Computing the Fisher Linear Discriminant

For this stage of the training pipeline, we first perform PCA to reduce the dimensionality of our data to N-c where is N is the number of images in our training set, and c is the number of classes. We then perform the Fisher Linear Discriminant, minimizing the within-class scatter and maximizing the between class scatter.

We utilize the OpenCV implementation of singular value decomposition to perform the above tasks.

4.3. Classification

Having determined the optimal projection from the previous stage, we are left with a set of features for each training sample. We are investigating two approaches for classification: NN, and SVM.

First, we are implementing a simple nearest-neighbour approach. In this approach, we simply store the training features. For a query sample, we compute the euclidean distance to the nearest neighbour, and assign the closest class within a certain threshold. This threshold will be learned through training and cross-validation.

We expect that the nearest-neighbour approach is unlikely to provide robust performance when testing on general test data. For this reason, we will also be implementing an SVM based classification scheme.

We will begin use the one vs. all (OVA) scheme, and train a set of two-class svm classifiers able to discriminate between a particular class, and the rest of the training set. Based on the set of results, we infer the winning class.

4.4. Indexing

In this stage, for each supplied test video, we iterate through each frame and perform face detection to extract faces. We resize the faces to 100px by 100px, project the sample to obtain a feature vector, and this then serves as the input to our classifiers.

If our classification output determines that the query sample is one of our learned classes, we have an index and record it in the form of the index we present above.

4.5. Query evaluation

Based on the previous progression of tasks, our problem setup makes the query evaluation relatively trivial. For a given query, we simple evaluate the boolean expression for each video, based on the recorded indexes for that video (which are obtained according to the previous section).

5. Evaluation Methodology

Our first requirement is a labeled (tagged) set face images. We have obtained this using the data of three Facebook users. From this training set, we will partition the set to select 60% of the dataset as the training set and 40% as the test set to evaluate the classification subsystem performance.

Our second major requirement is a video collection. We have obtained an evaluative one using the popular iPhone 4 as a representative video capture device. We will obtain a set of videos that will allow us to effectively test our query space. For example, we will collect a video with Person A and Person B, another with Person A and Person C, another with only Person A, and so on. This will allow us to evaluate queries such as A & B, A, or NOT(C). We have also collected scenes with largely side faces. We also have video with illumination variation as well as video samples with largely side faces, with some frontal face occurances.

We are obtaining the ground truth for the videos in a relatively straight-forward mechanical manner by simply watching the video and indicating the individuals that occur in the videos.

We will specifically evaluate the following aspects of the system:

1. The classification error, evaluated using the test set for both NN, and OVA-SVM.
2. The system performance as the number of training samples is increased.
3. The performance of video with predominantly side faces, and minimal frontal faces.
4. Performance comparison between illumination changes in scenes.

In our evaluation, we will track false positives, false negatives, true positives, and true negatives. This will allow us to develop precision-recall curves for the aforementioned experiments.

6. Preliminary Results

To date, we have the initial system implementation and datasets and intend to begin the system integration and evaluation phases shortly.

References

[1] P.N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. Fisherfaces: Recognition using class specic linear projection," *IEEE Trans. Patt. Anal. Mach. Intell* 19, 711 720, 1997

[2] P. Viola and M. Jones, "Robust Real Time Object Detection, *IEEE ICCV Workshop Statistical and Computational Theories of Vision*, July 2001.

[3] Face Recognition Homepage. http://www.face-rec.org/databases/

[4] OpenCV. http://opencv.willowgarage.com/