ScavengeIT - a face verification based photo database scavenger

I.C.T.M Speek*

Computer Vision
Delft University of Technology

July 3, 2014

Abstract

1 Introduction

Since the emergence of social media websites, finding images of friends has been made easier. By manually adding tags to photos, websites as Facebook are able to identify your friends in your online images. This report focuses on developing an application in Matlab to scavenge your local file system for any image containing a queried friend.

2 Problem definition

The challenge of scavenging a database of images for photos of a particular person require the need to verify one or multiple input images vs the faces in the image and return a list of images in which the input face is verified. This process can be divided into several parts:

- Processing the database images
- Detecting faces
- Extract discriminant features from input and detected faces
- Classify the similarities
- Decide whether faces are the same
- Return the verified faces

3 Solution

Say something more about V&J

To detect the faces in the images we can use the Viola & Jones cascaded classifiers. These cascaded classifiers are particularly convenient when speed is an issue and its unsure whether or not a face is present in the images. The cascaded weak learner decision stumps return a negative output as soon as a single classifier does so. This way negative samples are processed significantly faster and only detected faces are processed by all classifiers. As there are already pretrained cascaded classifiers available online from OpenCV, there is no need to train these as that process can take a week time.

When the faces are detected, discriminant features can be extracted. As described in [3] there a 3 levels of features by which persons can be identified. Level 1 represents the global appearance of the person and distinguishes the size of the face, gender and racial discriminant features and distuingishes the persons as a first glance. Level 2 contains the most discrimative features and represents a shape and appearance model for the face and smaller local patches within the face region. Level 3 contains the most specific features such as birthmarks, wrinkles and more detailed specifications and will be left out in this report.

The Intraface Facial Feature Detection & Tracking matlab library can be used to fit a shape model onto a detected face. This model annotates 49 feature landmarks in the face as can be seen in the upper left image in figure 3. This shape model can be used to verify face images against one another. The easiest and least robust way to distuinguish between faces would be to measure the person specific ratio of the length between the eyes and the length of the nose. By setting a threshold on the inputted

*Electronic address: i.c.t.m.speek@student.tudelft.nl; Corresponding author

face image ratio a decision can be made whether or not the found faces belong to the same person. This method works relatively well when only dealing with frontal faces. Because the scavenge application should be scavenging a broad ranges of images, the face discriminator should be invariant to pose, lighting and occlusion of features. This causes the need for another solution. [2] presents normalization techniques that normalize the input faces to a shape mean (a frontal view). By doing so, the faces can be verified more reliable as we are not dealing with any shortening of the distances in between features as the effect of different poses is removed from the face.

3.1 Active appearance models for face verification

Active Appearance models as described in [1] try to model the face appearance as complete as possible. This justified approximation of a case can be used to understand the interface (between different face identities) and intraface (between the same identity) variability. As the face approximation describes an optimization problem which is the same for every face, generality is solved offline to achieve rapid convergence online. To achieve this, active appearance models are generated by combining a level 2 shape variation model with a shape-normalized level 1 appearance model.

Add active appaearance image of appaearance as well

These variation models are trained using a training set of labelled images where landmarks mark key positions on each face. These points are alligned in a common co-ordinate frame and represented by a vector. A *Principal Component Analysis* (PCA) generated a statistical model of this shape variation where equation 1 describes an example face.

$$x = \bar{x} + P_s b_s \quad (1)$$

Where x described the example face, $\bar{x}$ the mean model and P_s is a set of orthogonal modes of shape variation and b_s is a set of shape parameters [1]. For each input face a set of shape and gray-level parameters is extracted which varies in identity, expression, pose and lighting. To efficiently match faces, the mahalonobix distance measure can be used which enhances the effect of inter-class variation (identitation) and surpresses the effect of intra-class variation(pose, lighting and expression) when the intra-class covariance matrix is available. This would mean that multiple input images should be provided of the same person in order to determine the intra-class covariance matrix. Assuming this is not available a *Linear Discriminant Analysis* (LDA) can be used to seperate the inter-class variability from the intra-class variability assuming the intra-class variation is similar for each individual. This way the mean coveriance matrix can be used to estimate the variance.

4 Implementation

The project's approach was done using the same order as presented in section 2. This section will present the approach in chronological order, starting with the libraries that are used and explaining the additional steps that had to be made to solve the face verification problem.

4.1 Used libraries and data models

As recommended the state-of-the-art facial feature detection model IntraFace was used. This matlab library provides functionality for fitting a feature model using 49 landmarks on detected faces within an image. Figure **??** displays the initial performance of the facial feature detection functionality using the supplied 'auto' OpenCV face detection method. However when using the 'interactive' mode that allows for selecting a bounding box as face detection all but 1 face allows for fitting the shape model as can be seen in figure 2. This indicates that there is need for a better face detection algorithm.

Figure 1: IntraFace demo facial feature detection 'auto' mode

Figure 2: IntraFace demo facial feature detection 'interactive' mode

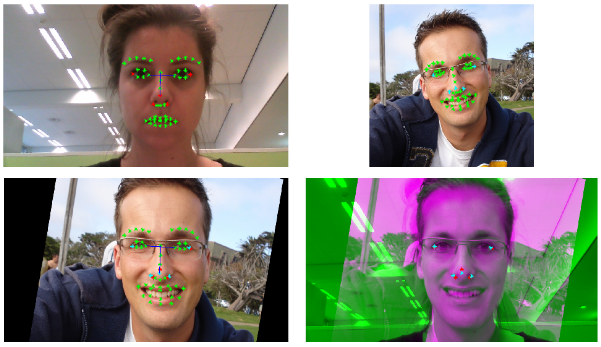

Figure 3: Normalized face and ratio

4.2 Own work

4.2.1 Face detection

4.2.2 Normalizing the faces

4.2.3 Feature Extraction

4.2.4 Classification

4.2.5 Decision model

5 Experiments

6 Results

7 Future Works

By learning the positive faces whilst doing the linear discriminant analysis, the faces can be used to determine the intra-class covariance matric and thus the Mahalonobis distance can be used to identify the individual.

Add references

References

[1] T.F. Cootes G.J. Edwards and C.J. Taylor. Face recognition using active appearance models. 1998.

[2] H.K. Ekenel H. Gao and R. Stiefelhagen. Pose normalization for local appearance-based face recognition. 2009.

[3] B. Klare and A.K. Jain. On a taxonomy of facial features. 2010.