Automatic Face Tagging

MOTIVATION

The use of smart phones has seen a tremendous rise in the last few years. People have these phones with them most of the times and whenever they feel like capturing a moment, they use the in-built camera of their phones as the quality of camera has improved significantly.

But management of these images later becomes a cumbersome task. Also, while sharing the image with others, not everyone knows all the people present in an image. One thing which comes to the rescue in this case is tagging. It sure helps a lot if the people present in an image are tagged with their names. One obvious example which comes quickly to mind is the Facebook tagging.

So I decided to develop an Android based Application which would automatically tag the people in an image as my mini project under the guidance of Prof. Subhashis Banerjee.

ALGORITHMS USED

Any face tagging task consists of two steps: first is to detect faces in an image and second is to identify them. I have used the following algorithms for each of the tasks.

1. **Face Detection:**
   I used Viola-Jones framework for face detection. It consists of two steps, first step is training in which a series of weak classifiers are trained on images of faces as well as non faces. This step is done only one time. After this in the second step which is detection step these weak classifiers are applied in cascade way to the image where the faces are to be detected. In my case I got an accuracy of 95-96% on face detection. This is the major time consuming task in the application.

2. **Face Recognition:**
   Images are very huge dimensional vector, for example if an image is of dimension N*M then it is treated as vector of dimension NM. So to reduce this dimension Principal Component Analysis (PCA) is used. The face space (vector space spanned by the principal components of faces) is further divided into the corresponding face class (images of a particular person) cluster using Linear Discriminant Analysis (LDA). Face recognition is a two step process – training and recognition. The face recognition system is first trained using PCA and LDA. Then in the second step of recognition, when a new face image comes, it is first projected onto the space obtained after PCA. Then its
distance from all the clusters is calculated. If the smallest distance is less than some threshold then it is label according to the cluster otherwise it is labeled as new.

**ANDROID FACE TAGGING APPLICATION**

The application can be divided into two parts: frontend and backend. The frontend contains the user interface and its task is to call the backend whenever required. It is written in Java. The backend is the place where all the actual processing like loading the database, detection of faces and recognizing them, happens. All of the backend tasks are heavy and require a lot of processing power, so they are written in C++ using OpenCV library. Both the parts of application are glued together using “Java Native Interface”.

The application has four options (see fig. 1)

1. **Load database**
   This option is the pre-requisite of all operations in this application. It basically loads the existing database of people. This database contains the principal components of face images used in training and the clustering information of different individuals (see fig. 2)

2. **Load image**
   There are two ways to get an image in that application for processing either load it from the memory capture image using the mobile camera. This option allows user to load an image using the gallery application. This option calls the gallery application so that user can find the required image or image folder.

3. **Capture image**
   This is another option of getting an image in application. This option calls the camera application in phone so that the user can capture image and directly load in the application without saving it on SD Card.

4. **Recognize**
   This option processes the loaded image. It first detects faces selected image. Then it searches if that image is of a person already present in the database or not. If the person is present it takes the name of the person but if not then it asks for the name of that person and adds that person to the database. Finally, it shows the image with the people in it tagged with their name (see fig. 3 and 4)

This application was tested on “Samsung Young” phone with “Android 2.3.3”. The accuracy on frontal faces was found to be 95% but the performance of side faces was not as expected. To increase the accuracy I used the cluster center to find the distance from image rather than using each image in dataset. This improved the results on non frontal images by 5%-6%. This project was my first experience with android platform and face recognition so I learned a lot from it. I am working on its GUI so that I can publish this application in market.
SOME SNAPSHOTs OF THE APplication

Fig 1 - First page of the application

Fig 2 – Database is loaded

Fig 3 – New face added in the database

Fig 4 – A tagged image