BUILT-IN FACE RECOGNITION FOR SMART PHOTO SHARING IN MOBILE DEVICES

Esteban Vazquez-Fernandez, Hugo Garcia-Pardo, Daniel Gonzalez-Jimenez, Luis Perez-Freire

GRADIANT, Galician R&D Center in Advanced Telecommunications, Spain
evazquez@gradiant.org

ABSTRACT

This paper presents a smart photo sharing application for mobile devices based on an efficient yet powerful face recognition engine. The system allows registering subjects either from live video (mobile camera) or from still images, linking the obtained biometric information to the already existing phone contacts. The photo sharing application allows the user to select images which will be automatically processed with the *built-in face recognition module* based on Local Binary Patterns: already registered contacts will be identified, so that the current picture can be sent to them in a smart, user-friendly manner. The application also allows to tag unknown subjects and store (or update) their biometric profiles. All the processing is done in the mobile device.

The current application runs on Android O.S., having been tested in two mobile platforms: HTC Desire and Samsung Galaxy Tab. The computational burden for each of the processing modules is provided in both devices, demonstrating the feasibility of the proposed approach for mobile platforms.

Index Terms— Face recognition, computer vision, mobile computing, Android, smartphone

1. INTRODUCTION

The classical architecture of computer vision systems usually comprises the following modules:

- Sensor: frequently a digital camera based in a CCD or CMOS device.
- Acquisition interface for data transfer: specific frame grabbers, CameraLink, IEEE1394, USB or the more recent GigE standard.
- Processing Unit: Usually a computer, either general purpose, specific or embedded. It needs libraries of signal and image processing functions.
- User interfaces: usually a screen for visualization and a mouse or a keyboard for the input.
- Other peripherals, ranging from sensors to communication devices.

Over the last years, image acquisition, visualization and communication capabilities have been integrated in mobile phones, while computational power was a limitation. However, this situation is changing with the emerging market of smartphones. On one hand, the hardware of these devices is increasing in processing power and storage capacity, reducing the gap with desktop computers. On the other hand, the popularity of operating systems oriented to third party applications (Android, iOS, etc.) brings to developers access to SDKs and NDKs for native applications development (e.g. C++) [1]. In addition, the rapid growth of application markets such as AppStore or Android Market, is attracting much attention from both academic and industrial forums [2].

This situation opens numerous possibilities for implementation of existing computer vision applications in these devices, as well as new ones, taking advantage of its portability and integration of sensors. One of these fields is face recognition. Some examples of its applications can be: biometric login, face recognition based auto focus or social applications that allow users to interact and share data.

A smartphone based face recognition system does not only suffer from the same problems of a computer based system, such as illumination, occlusion and pose variations, but is also limited by other factors: limited computing power, lower quality cameras and memory [3]. Although many works have aimed at developing robust face recognition algorithms, relatively few works have focused on adapting the technology to the domain of mobile computing. In some systems, mobile devices are used only to sense video and the processing tasks are performed on a server via a wireless network [4] [5] [6]. More recent works show a trend to make all the computations in the mobile device [7] [8] [9].

In this work, we propose a simple, but robust and efficient face recognition algorithm for implementation in mobile devices. It is based on Local Binary Patterns (LBP) features, which have shown to be a powerful and computationally efficient feature extractor. The algorithm constitutes the engine of a new face recognition based photo sharing application, which is proposed in this paper. It allows users to link biometric and contact information, which will be used to automatic

This work was partially funded by Xunta de Galicia under projects VISAGE (10TIC008CT) and VIXIA (09TIC026E)

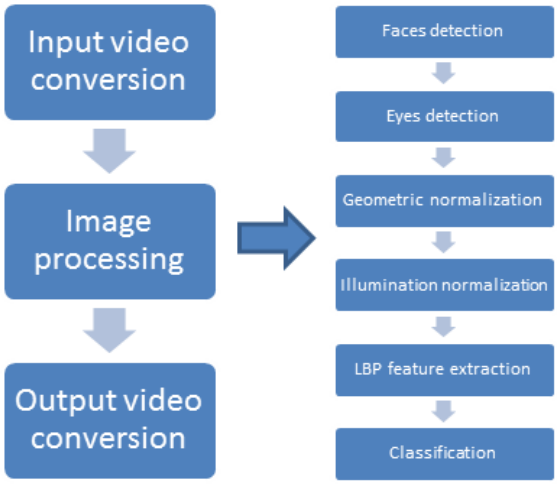


Fig. 1. Image processing chain

tag the faces in a picture. Then, the option of sending automatically the picture to the identified contacts is available by simply dragging the selected contacts to a "Send" button. The implementation is made for the Google Android platform, using OpenCV libraries for image processing. It is important to take into account that all the processing is done in the mobile domain, without the need of any external computation.

The paper is organized as follows. The implemented face recognition algorithm is presented in Section 2. In Section 3 we make an analysis of the computational cost of the recognition process. The photo sharing application is described in Section 4. Finally, some conclusions are given in Section 5.

2. FACE RECOGNITION ALGORITHM

In this section we will briefly explain the different parts of the processing chain. A block diagram of the global process is shown in Figure 1. There are two mainstreams for the computing process. One is in the Java domain (Android SDK), which is intended for the GUI, communication, camera access and video conversion. The other is in the C++ domain (Android NDK), which is intended for the image processing operations.

First, we need to convert the video format from the camera to a suitable format to be processed in OpenCV libraries. The conversion is made from a YUV12 video format (8 bit Y plane followed by 8 bit 2×2 subsampled V and U planes) to a RGB image matrix structure. This is not a free cost operation and accounts for 20% to 25% of the total computational cost, as can be seen in Table 1. After performing the face processing operations, the frame is converted again to a video stream suitable for being displayed in the mobile device using the Android SDK (RGB format, not YUV). This is performed in the Java mainstream.

Two methods have been used for face detection: Viola & Jones Haar based [10] and LBP-based face detector [9] (OpenCV implementations). Preliminary tests have shown that the LBP detector has a lower computational cost (see Table 1 for more details), but suffers from detection problems in non-frontal faces. To achieve a trade-off between detection performance and computational cost, LBPs are used in the enrollment stage, while Haar detector is used in identification (automatic face tagging in pictures). In the first one, the objective is to perform the detection in real time video. The user can point the camera to the face of the person to be registered, so LBP frontal face detectors seem adequate for the purpose. In the second one, face poses can not be controlled and the processing is not needed to be performed in strictly real time, so the Viola & Jones detector is used (due to its slightly broader range of detection).

One of the most important steps in a face recognition algorithm is face normalization. To prevent errors due to misalignments, rotation or scale differences, a warping process is applied. The normalization is based on the eye positions located in the previously detected face, using Viola & Jones method. They are used to align and rescale the image to a fixed interocular distance. Rotation is also corrected, placing the eyes in an horizontal line. The final size of the normalized image is 105×120.

Illumination normalization is one of the major challenges for robust face recognition, degrading considerably the performance. We adopted the method proposed in [11], which models the layers of the retina. The model consists on a non-linear light adaptation and a Difference of Gaussian (DoG) filter to remove illumination variation. Finally, a truncation is used to enhance the global image contrast.

Local Binary Patterns (LBP) [12] have shown to be a robust descriptor for face recognition in uncontrolled conditions [13]. We use the simple form of the original uniform LBP (3×3 neighborhood and 59 bins) coded in 8 bit words. The size of LPB cells are 15×15, which are placed in a face grid of 7×8 blocks. An histogram is computed and normalized to unit length for each block. Values higher than 0.2 are truncated and then the histograms are normalized again and concatenated in a unique feature vector of $59 \times 7 \times 8$ characteristics ($\approx$ 13KB).

Several faces can be registered for each user in order to capture different poses and facial expressions. The classification is based on L2 distance measurements.

3. PERFORMANCE ANALYSIS

Results presented in [11] show that the combination of Retina illumination normalization and LBP features obtains good recognition rates in FERET database and YaleB database (98% and 100% respectively). Furthermore, this method obtains similar results than Local Gabor Binary Pattern Histogram Sequence (LGBPHS) [14], but with much lower com-

plexity (computation time), so it seems a good starting point for mobile face recognition implementation.

We also show a performance analysis of the implemented algorithms for two different mobile platforms in Table 1. An HTC Desire, with a Qualcomm QSD 8250 1 GHz processor and 576 MB RAM (768×432 video resolution), and a Samsung Galaxy Tab with a 1 GHz ARM Cortex A8 processor and 512 MB RAM (800×600 video resolution) are used for the tests. Apart from the time consuming video conversions (already discussed in Section 2) from this Table we can highlight the following points[1]:

- The Samsung device is slightly slower than the HTC platform, albeit having similar technical specifications.
- LBP-based face detection is notably faster than the Haar-based approach. However, preliminary experiments on real data seem to demonstrate that the OpenCV version of the latter performs better (in terms of detection rates) than the OpenCV version of the former. Further experiments are needed to compare both systems on standard datasets.
- Face and eye detection are computationally intensive tasks ($\approx 63\%$ and $\approx 42\%$ of the global face processing burden with Haar-based and LBP-based face detection respectively). Geometric warping is also a computationally demanding step.
- Both illumination normalization and LBP feature extraction are efficient and suitable for mobile platforms (less than $\approx 7\%$ of the whole computation load).
- The computational load of identification is negligible for a typical number of stored contacts (computation time is given for 50 registered contacts with 4 stored faces per contact)

Table 1. Average computation time for each of the different modules in two mobile platforms

Operation	HTC Desire	Samsung Galaxy Tab
1) Input video conversion	50.1381 ms	72.5963 ms
2) Face detection:		
-Haar features	131.976 ms	163.176 ms
-LBP features	31.8782 ms	36.3714 ms
3) Eyes detection	39.1852 ms	49.2257 ms
4) Geometric normaliz.	40.1932 ms	51.1157 ms
5) Illumination normaliz.	21.0278 ms	56.5148 ms
6) LBP feature extraction	3.69485 ms	7.00171 ms
7) Classification (not in enrollment)	6.9471 ms	10.3688 ms
8) Output video conversion	26.7675 ms	24.0356 ms

4. PHOTO SHARING APPLICATION

Based on the face recognition system described above, we propose an embedded smart photo sharing application. The aim of the application is to share images with a contact at the touch of a finger. The application has two stages: enrollment and identification (automatic face tagging in pictures). In the enrollment stage, LBP templates are extracted from face images and linked to already existing phone contacts. The identification stage is used to detect faces in a selected picture and identify already stored contacts, so that the current picture can be sent to the detected subjects in a smart, user-friendly manner.

Enrollment of users can be performed both from video input (i.e. video sequence captured by the embedded camera) or from stored images. In the first case, the system locates the face in the video sequence, extracts the information and asks the user to link the face image and biometric template to a contact previously stored in the phone (see Figure 2). This information is stored in a database and loaded in the identification stage.

For the identification stage, the application allows the user to select an image which will be automatically processed with the built-in face recognition module. The faces in the picture will be detected and compared against the (biometric) database of contacts. As a result, the contacts in the picture will be identified (ordered list of potential candidates). When finished, we only need to drag and drop the tagged faces to the mail icon (see Figure 2) and and they will be sent to the email addresses linked to the identified contacts. The application also allows the user to tag unknown subjects appearing in the pictures and storing (or updating) their biometric profiles.

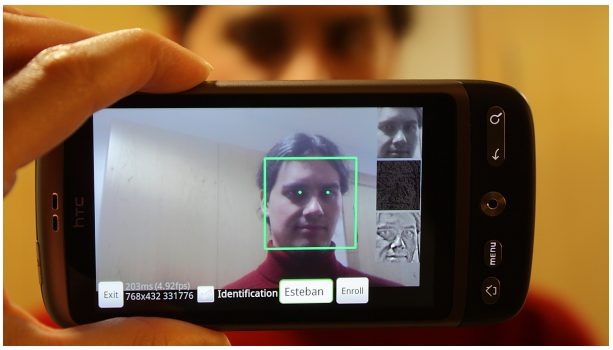
The proposed application is very useful for quick image sharing. It combines several phone SDK capabilities to show the versatility of the device as an integrated computer vision system, managing acquisition, processing, communication and presentation tasks.

5. CONCLUSIONS AND FURTHER WORK

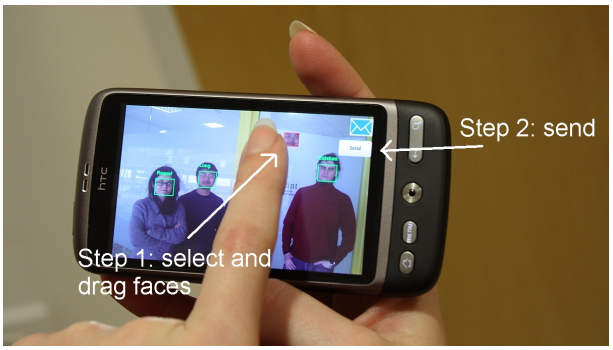
Mobile phones and smartphones have been increasing their computational power in the last years. These new devices bring together imaging, processing, communication and displaying capabilities. The traditional computer vision paradigm is moving towards new ways of implementation. A the same time, this new way of mobile computer vision implementation brings new possibilities of application.

An image sharing application based on face recognition has been developed. It is a useful tool for fast image sharing among phone contacts. The key function is to link the contact list of the phone to the extracted biometric information of the registered users. This information is used by a face recognizer

[1]Detection and processing load is provided for one face



(a) Enrollment stage



(b) Identification stage

Fig. 2. Captures of the photo sharing application

module to automatically tag a picture at the time of capturing or either by selecting an already stored image. The tagged images can be sent directly to the registered contact via email.

We have implemented a face recognition scheme suited to work efficiently in Android mobile devices. It is based on LBP features and adaptive illumination normalization to obtain robust identification. A comparison of computing times for the different stages of the recognition chain is shown. The key point is that all the processing is done in the mobile device, without the need of any external computation or database server. The implementation is done for Android platform, so it can be easily ported to other devices: tablets, Set-top Boxes, etc.

Some further work is proposed: As previously shown, video format conversion is an intensive time consuming task. It will be interesting to test the performance of processing in the luminance channel of YUV video (Y channel). This should reduce the computation time at the cost of not displaying the color image in real time processing (enrollment stage).

Another point is to study the use of clustering algorithms for global picture searching and tagging. This can be a useful tool for searching all the stored pictures where a selected contact appears.

6. REFERENCES

[1] OpenCV - Android Website, `http://opencv.willowgarage.com/wiki/Android`

[2] A. Holzer and J. Ondrus, "Mobile application market: A developer's perspective," *Telematics and informatics*, vol. 28, no. 1, pp. 22-31, Febr. 2011

[3] K. Choi, K. A. Toh and H. Byun, "Realtime training on mobile devices for face recognition applications," *Pattern recognition* vol. 44, no. 2, pp. 386-400, Febr. 2011.

[4] E. Weinstein, P. Ho, B. Heisele, T. Poggio, K. Steele, A. Agarwal, "Handheld face identification technology in a pervasive computing environment," *Short Paper Proceedings, Pervasive 2002*, pp. 48-54, 2002.

[5] T. Hazen, E. Weinstein, R. Kabir, A. Park and B. Heisele, "Multi-modal face and speaker identification on a handheld device," *Workshop on Multimodal User Authentication*, SantaBarbara, CA, pp.113-120, 2003.

[6] S. Mukherjee, Z. Chen, A. Gangopadhyay and S. Russell, "A secure face recognition system for mobile devices without the need of decryption," *Workshop on Secure Knowledge Management*, 2008.

[7] S. Jung, Y. Chung, J. Yoo, K. Moon, "Real-time face verification for mobile platforms," *Proc. of International Symposium on Advances in Visual Computing*, Part II, pp. 832, 2008.

[8] C. Doukas, I. Maglogiannis, "A fast mobile face recognition system for android OS based on Eigenfaces decomposition," *Proc. of Artificial Intelligence Applications and Innovations*, vol. 339 AICT, pp. 295-302, 2010.

[9] S. Marcel, C. Cool, C. Atanasoaei, F. Tarsetti, J. Pesn, P. Matejka, J. Cernocky, M. Helistekangas and M. Turtinen, "MOBIO: Mobile biometric face and speaker authentication," *CVPR*, 2010.

[10] P. Viola, and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," *CVPR*, vol. 1, pp. 511-518, 2001.

[11] N. S. Vu, A. Caplier, "Illumination-robust face recognition using retina modeling," *ICIP*, pp.3289-3292, Nov. 2009.

[12] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. PAMI*, vol.24, no.7, pp.971-987, Jul 2002.

[13] L. Wolf, T. Hassner and Y. Taigman, "Descriptor Based Methods in the Wild," *Faces in Real-Life Images workshop at the European Conference on Computer Vision (ECCV)*, Oct 2008.

[14] W. Zhang, S. Shan, W. Gao, X. Chen and H. Zhang, "Local Gabor binary pattern histogram sequence (LGBPHS): a novel non-statistical model for face representation and recognition," *ICCV*, vol. 1, pp.786-791, 2005.