AN EMBEDDED SYSTEM FOR face recognition BASED on raspberry pi AND SUPPORT VECTOR MACHINEs

hệ thống nhận dạng khuôn mặt với máy vector hỗ trợ thực thi trên máy tính nhúng raspberry pi

Hồ Anh Trang, Hồ Phước Tiến, Huỳnh Việt Thắng

Danang University of Science and Technology, The University of Danang, VIETNAM

Email: [anhtrangdn89@gmail.com](mailto:anhtrangdn89@gmail.com); [hptien@dut.udn.vn](mailto:hptien@dut.udn.vn); [thanghv@dut.udn.vn](mailto:thanghv@dut.udn.vn)

**Tóm tắt –** Nhận dạng khuôn mặt người là một bài toán quan trọng của thị giác máy tính và có nhiều ứng dụng trong an ninh, bảo mật, cũng như mạng xã hội. Các hệ thống nhận dạng mặt người yêu cầu độ tin cậy, tính chính xác, thời gian xử lý nhanh và giá thành thấp. Trong bài báo này, chúng tôi thiết kế một hệ thống nhận dạng khuôn mặt người hướng đến đáp ứng các yêu cầu nêu trên. Chúng tôi thực thi và đánh giá hệ thống này trên board mạch nhúng Raspberry Pi 2. Phương pháp phân tích thành phần chính (PCA) được sử dụng để trích chọn thuộc tính từ dữ liệu ảnh đầu vào, và sau đó máy vector hỗ trợ (SVM) được sử dụng để nhận dạng các khuôn mặt người khác nhau trong tập cơ sở dữ liệu được lựa chọn. Hệ thống cho kết quả nhận dạng cao với thời gian nhận dạng nhanh.

**Từ khóa –** nhận dạng khuôn mặt người; phân tích thành phần chính; máy vector hỗ trợ; Raspberry Pi, hệ thống nhúng;

**Abstract –** Face recognition is an important topic of computer vision and has many applications in security, verification, and social networks. Reliability, accuracy, fast processing and low-cost are desirable in commercial human face recognition systems. The Raspberry Pi 2 based system proposed in this paper tries to meet some requirements such as resources optimization, low cost, low complexity and improved quality as well as processing speed. The system utilizes Principal Component Analysis (PCA) to represent face images in lower dimension and then recognize different persons using Support Vector Machine (SVM). The system shows a high recognition rate (on average) as well as fast processing time.

**Key words -** face recognition; PCA; SVM; Raspberry Pi; embedded system;

# Introduction

As security has now become a big concern in our society, security systems have attracted much interest of researchers, as well as industry. Among many biometrics-based recognition methods, for example iris and fingerprint recognition, face has several advantages since it can be captured at distance and in a covert manner. Recently, face recognition has witnessed great success in human recognition, promising interesting applications in surveillance systems, verification systems, and social networks. Such a success is a result of multidisciplinary researches including computer vision, applied mathematics, machine learning, neuroscience, and psychology.

Face recognition has very often been implemented in desktop computers, as it requires a great deal of computation. Generally, face recognition concerns two main modules: feature extraction and recognition. Feature extraction is to obtain pertinent characteristics that are capable to distinguish a human face from another. Based on such features, the recognition module will determine the identity of a presented face. A large number of methods have been proposed for the problem of human face recognition: they include well-known approaches such as eigenfaces and neural network, as well as the currently famous deep learning method [1].

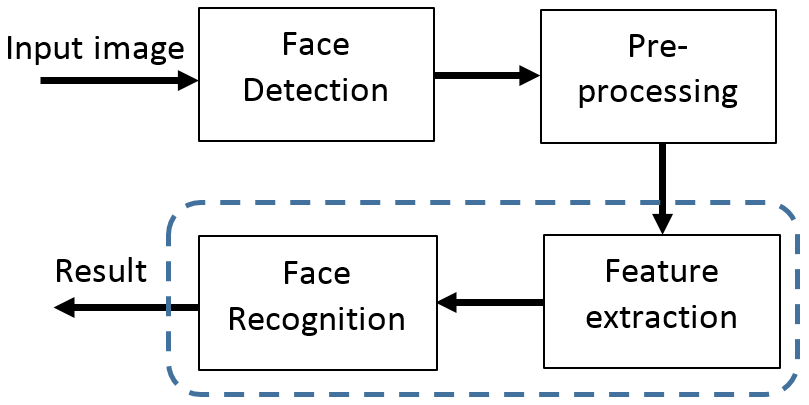
Since the above systems often need very powerful computers, they are not appropriate for applications that give priority to mobility and low cost, for example, for person recognition in a class or in a small shop. For this purpose, we propose to use a Raspberry Pi embedded board, with an upgraded ARMv7 multicore processor and 1 Gigabyte of RAM, in order to implement a human face recognition system. To be compatible with an embedded system, our proposed method contains “light” algorithms that are Principal Component Analysis (PCA) and Support Vector Machine (SVM). Indeed, eigenfaces through PCA assume feature extraction, followed by SVM that is used for recognition. We note that our face recognition system was carried out entirely on the Raspberry Pi board. Its testing provided promising results.

The remains of this paper are as follows: Section 2 introduces the proposed face recognition system, PCA, and SVM. In Section 3, we describe a face recognition system on a Raspberry Pi 2 embedded board. Section 4 presents experimental results of the proposed system. Finally, conclusion and future work are presented in Section 5.

# Background

## System overview

Figure 1 presents the block diagram of the proposed face recogntion system. This system consists of four modules: face detection, preprocessing, feature extraction and face recognition. Given an input image (from camera or computer memory), face detection module identifies regions that contain human faces. The resulting human face images will then be preprocessed or enhanced, normally with 2D filtering or histogram equalizer for better quality images. We choose to perform face preprocessing after face detection since the detected face images typically have smaller sizes than the original input images, thereby leading to less computation for the preprocessing module. After preprocessing, feature extraction is used to extract the most important information (or features) from the raw image data. The input of the Feature extraction block is image intensity. This step reduces data redundancy as well as computational complexity in the following recognition module. Since, in this work, we deal with the database mainly containing already cropped face images, we will focus exclusively on the last two modules using PCA and SVM techniques, the face detection (which can be carried out by AdaBoost Algorithm) and preprocessing modules will not be included in this work.



*Fig. 1. Block diagram of face recognition system*

## Feature extraction with PCA

Principal Component Analysis (PCA) is a useful technique data compression, i.e. dimensionality reduction. From a usual tool for data analysis, it has been successfully applied to the face recognition problem and has gained much popularity in this domain [2].

Generally, the PCA technique starts with a covariance matrix. In the case of face recognition, this matrix is obtained from the training images. Then, the principal components are the principal eigenvectors of the covariance matrix. These eigenvectors are selected according to the largest corresponding eigenvalues. The selected eigenvectors serve approximately as the basis of the face space. Consequently, a given face image (training or testing image) can be represented by a set of coefficients, which are results of the projection of this image on the basis (or selected eigenvectors).

Since the number of selected eigenvectors is much smaller than the dimension of an image, the above projection ensures dimensionality reduction. While PCA is widely known, here we only summarize its main idea. Readers that want more details about PCA can easily find them in the literature.

## Recognition

In early works concerning face recognition, template matching was a preferred recognition method [3][4]. Accoding to this approach, both training and testing images are represented by coordinates in the space generated by eigenvectors. Then, a distance, very often Euclidean distance, is used to evaluate the similarity between two human face images, which are now represented by two sets of coefficients, and determine whether these two faces are the same. Despite its simplicity, template matching reveals some drawbacks in face recognition: the main problem may relate to the difficulty of finding an appropariate distance that is capable of learning a manifold.

Neural network is another useful method for face recognition, especially with the recent success of deep learning [1]. However, its computational complexity seems not to be suitable for an embedded board.

In this work, we opted for Support Vector Machine (SVM) for the recognition step. The SVM technique has showed high efficiency in classification and recognition; besides, SVM is also “light” enough for an implementation on an embedded board.

Originally, SVM is a binary classifier and a supervised learning method. We consider a training dataset: is a set of data points, is a corresponding set of labels. In the case of binary classification, is either 1 or -1.

The training process seeks a vector w that allows a clear separation of two data groups. In other words, w is the solution to the following optimization problem:

In the testing phase, the category of each data point is determined by projecting this point on w. It is important to note that the max-margin hyperplane to separate the two categories depends only on the data points *xi* that are the nearest to it. These *xi* are called support vectors and hence explain the name of the method [5].

In this work, as we want to recognize many persons this problem is hence related to a multiple classes recognizer. However, the binary SVM can easily be extended to handle multiple classes [6]. In fact, the multiple classes SVM combines all possible binary classifers, and each unseen example is classified to the class that “wins” most binary classifications.

# Experimental setup

## Raspberry Pi 2 embedded computer

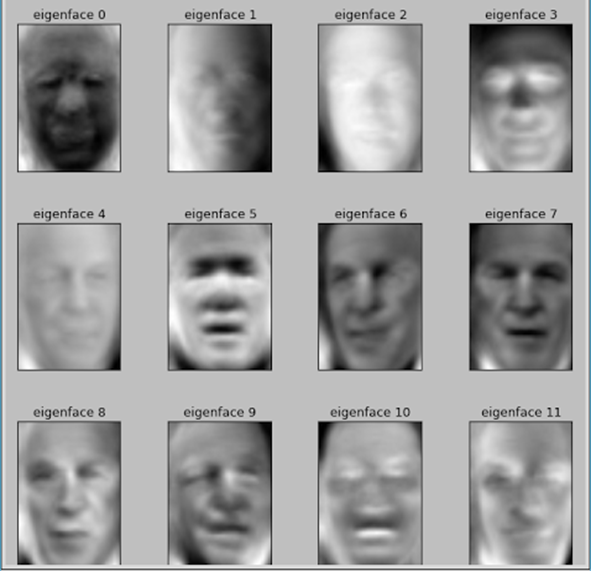
In this section, we present the implementation of the proposed face recognition system. We employed a Raspberry Pi 2 [7] embedded computer. This embedded computer features the ARM Cortex A7 (32 bit) mircroprocessor, operates at a clock frequency of 900 MHz and has 1 GB of RAM. The Raspberry Pi 2 board runs under Raspbian (Linux-based) operating system. The OpenCV [8] and Sklearn [9] libraries are both installed into the board to perform all detection and recognition tasks in the system.

## Databases

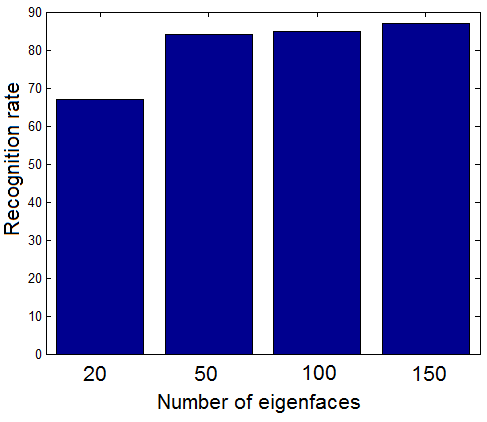
The dataset for evaluating the proposed system is extracted from the “Labeled Faces in the Wild” face database [10]. The dataset consists of many folders: each folder presents one person and his/her faces in different conditions. Taking into account both the computational resources of the proposed embedded system and its realistic ability of human recognition, we selected seven sets of human faces corresponding to seven persons from the dataset for testing; each person has more than 70 face images of 250250 pixels (62500 pixels in one image). These persons are well-known politicians: Ariel Sharon, Colin Powel, Donald Rumsfeld, George W Bush, Gerhard Schroeder, Hugo Chavez, and Tony Blair (denoted as person A to G). This design choice results in a dataset of 1288 face images in total. Figure 2 illustrates sample images (extracted from the dataset) of seven persons selected for face recognition in the system.



*Fig. 2. Seven persons (denoted as Person A to G) selected from the “Labeled Faces in the Wild” database.*



*Fig. 3. The first 12 eigenfaces obtained from the training images.*



*Fig.4. Average recognition rate (%) according to various number of eigenfaces*

For experiments, we divide the dataset into training set and testing set as follows: 75% of the dataset was used for training, the remaining 25% for testing. Both the training and testing sets contain the seven persons.

For the feature extraction, we use PCA and employ only 20, 50, 100 and 150 principal components (from 62500 raw image pixels) to build corresponding eigenfaces for recognition task. Then, in the recognition module, we employ 21 binary SVMs with radial basis function (RBF) kernel to perform the classification of seven persons.

## Performance criteria

Firstly, we use recognition rate (or precision) in order to evaluate the recognition performance of the system.

Secondly, we are interested in processing time to have a rough estimation about the real-time processing ability of the proposed Raspberry-Pi based embedded sytem.

# Results and Evaluation

## Eigenfaces through PCA

As described above, we used the training images to establish a covariance matrix, from which eigenvectors or eigenfaces are computed. Figure 3 illustrates the first 12 eigenfaces. We can see that these eigenfaces represent some persons in the dataset. It is worth noting that a given face image is a combination of such eigenfaces. In the next section we will consider the influence of the number of eigenfaces on the recognition performance.

## Face recognition using PCA and SVM

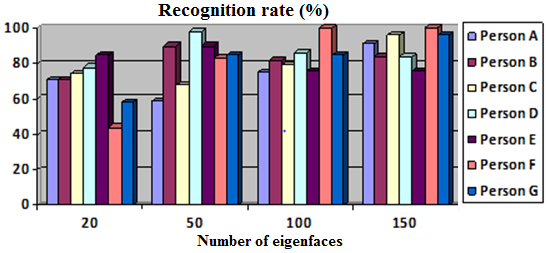
Combining eigenfaces and SVM, we obtained promising recognition results. Generally, the recognition rate increases with the number of eigenfaces employed. Figure 4 shows the recognition rate averaged over all persons. It seems that increasing the number of eigenfaces will improve the recognition performance. This result can be explained by the fact that more eigenfaces represent more features of each face image and, hence, increase the ability of distinguishing different human faces. Besides, fixing an appropriate number of eigenfaces is an interesting issue. From Figure 4, using 50 eigenfaces gives a clearly better result than 20 eigenfaces (i.e., 84% with 50 eigenfaces compared to 67% with 20 eigenfaces). However, there is only a slight improvement from 50 to 100 to 150 eigenfaces, corresponding to recognition rates of 85% and 87% when using 100 and 150 eigenfaces, respectively. It is important to remind that employing more eigenfaces will require more computation, this effect is still more significant in an embedded system.

Figure 5 presents recognition rates for different persons. It is interesting to see that the influence of the number of eigenfaces varies according to particular persons. For instance, person E seems require a small number of eigenfaces. In fact, using more eigenfaces to represent this person does not lead to a significant improvement in recognition rate, or even worse results (for example in the cases of 100 and 150 eigenfaces). The other persons generally can be better recognized with more eigenfaces (e.g. persons F, G).

Concerning recognition time, Figure 6 shows an increase in time according to the number of eigenfaces for the recognition task. The recognition times are 0.20, 0.26, 0.53 and 0.84 (second) when using 20, 50, 100 and 150 eigenfaces, respectively. We can observe from Figure 6 that the recognition time increases almost linearly from 20 to 150 eigenfaces, while it is not the case for the recognition rate. Hence, the number of eigenfaces must be carefully considered when we want to implement a face recognition method on embedded systems. We also remind that the training time is much higher, but it can be carried out only once before being used for all test images.

## Remark

This work is focused mainly on SVM for the recognition task. Yet, we implemented the template matching method for the purpose of comparison. Our experiment showed that while also using PCA as feature extraction, the template matching method did not provide acceptable results with the chosen database (“Labeled Faces in the Wild”) used in the present paper. The corresponding average recognition rate was just about 30%, which is much lower than the results with SVM.



*Fig. 5. Recognition rates (%) for different persons under the impact of varied number of eigenfaces*

*Fig. 6. Processing time (in second) for recognition task under the impact of varied number of eigenfaces*

Another inportant observation from the template matching method is its very high processing time. This is due to the nature of template matching: we need to compute the distance of a test image to every image in the training database; and for large database this computation is therefore very time consuming and would degrade the overall processing performance of the recognition system.

We note that the SVM-based face recognition system implemented in this work is one of our attempts to investigate the capability of the Raspberry Pi computer in embedded security-like applications, and we, therefore, considered only a small dataset of seven persons with 1288 face images. It is also our interest to expand the system with larger dataset (more people for the recognition task) and study the corresponding recognition rate and excution performance of the system, for which it would take more computing resouces (more memory, execution time, number of SVMs employed). This expansion, however, will be our future study.

# Conclusion

We present in this paper a practical system of face recognition on the Raspberry Pi 2 embedded computer with PCA and SVM techinques. The studied system showed a high recognition rate (on average) as well as fast recognition times. Our face recognition system reveals itself as a reliable, low-power and low-cost solution for practical embedded applications, especially in the context of Internet-of-Things (IoT); examples include face recognition in surveillance systems, verification systems, and social networks. For future work, we will expand the system with more data as well as study other recognition techniques for improving recognition performance.

References

1. O. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” BMVC 2015, pp. 1-12.
2. M. Turk and A. Pentland, “Eigenfaces for recognition,” Journal of Cognitive Neuroscience, 3(1), 1991.
3. H. Zhao, X-J Liang, and P. Yang, “Research on Face Recognition Based on Embedded System,” Mathematical Problems in Engineering, 2013.
4. Trương Văn Trương, Huỳnh Việt Thắng, “Nhận dạng khuôn mặt trên máy tính nhúng Raspberry Pi“, Tạp chí Khoa học Công nghệ ĐHĐN. Số 1(98), 2016.
5. C. Cortes and V. Vapnik, “Support-Vector Networks,” Machine Learning, vol. 20 (3), pp. 273-297, 1995.
6. C. W. Hsu and C. J. Lin, “A comparison of methods for multiclass support vector machines,” IEEE Trans. Neural Netw., 13(2), pp. 415-25, 2002.
7. <http://www.raspberry.org> (last accessed 29.09.2016)
8. <http://opencv.org/> (last accessed 29.09.2016)
9. <http://scikit-learn.org/> (last accessed 29.09.2016)
10. <http://vis-www.cs.umass.edu/lfw/> (last accessed 29.09.2016)

**Phần trả lời và giải trình các ý kiến của phản biện**

Nhóm tác giả đã điều chỉnh nội dung bài báo theo ý kiến của Phản biện. Dưới đây là giải trình chi tiết. Trân trọng cảm ơn Phản biện và Ban biên tập Tạp chí.

|  |  |  |
| --- | --- | --- |
| **TT** | **Ý kiến phản biện** | **Trả lời của nhóm tác giả** |
| 1 | Phương pháp dùng đề phát hiện khuôn mặt trước khi trích thông tin độ sáng? Kỹ thuật loại bỏ vùng ảnh ở các góc (không thuộc khuôn mặt) trước khi đưa vào xử lý PCA. | Như đã trình bày, bài báo không thực hiện việc phát hiện khuôn mặt. Tuy nhiên, có nhiều cách để phát hiện khuôn mặt, ví dụ giải thuật AdaBoost (tác giả đã thêm vào trong bài báo). Và do đó, bài báo cũng không tập trung vào việc xử lý các đối tượng không liên quan có thể xuất hiện trong ảnh chứa khuôn mặt. Trong bài báo này, các ảnh sử dụng là ảnh đã được cắt sẵn, bao gồm khuôn mặt và nền ở phía sau. |

**Thông tin về tác giả**

|  |  |
| --- | --- |
|  | **Hồ Anh Trang**  - Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu): Tốt nghiệp Đại học ngành Điện Tử - Viễn Thông tháng 06/2012; Tốt nghiệp Thạc sỹ chuyên ngành Kỹ thuật Điện Tử tại Đại học Đà Nẵng tháng 06/2016;  - Tóm tắt công việc hiện tại (chức vụ, cơ quan): Kỹ sư thiết kế; Công ty eSilicon Đà Nẵng.  - Lĩnh vực quan tâm: Vi mạch và Hệ thống nhúng; Xử lý tín hiệu;  - Điện thoại: +84-(0)902.200.164  - Email: [*anhtrangdn89@gmail.com*](mailto:anhtrangdn89@gmail.com) |
| D:\PERSONAL\Photo\Iden_Tien.jpg | **Hồ Phước Tiến**  - Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu): Tốt nghiệp Đại học ngành Viễn Thông (PFIEV) tại Đại học Bách Khoa TPHCM năm 2004; Cao học và Tiến sĩ ngành Xử lý tín hiệu, hình ảnh, tiếng nói, viễn thông (SIPT) tại Đại học Grenoble lần lượt vào các năm 2006 và 2010.  - Tóm tắt công việc hiện tại (chức vụ, cơ quan): Giảng viên, Khoa Điện tử - Viễn thông, Trường Đại học Bách Khoa, Đại học Đà Nẵng.  - Lĩnh vực quan tâm: Xử lý tín hiệu, xử lý ảnh, thị giác máy tính.  - Điện thoại: +84-(0)905.493.457  - Email: [hptien@dut.udn.vn](mailto:hptien@dut.udn.vn) |
| E:\CONGVIEC\DoAnTotNghiep\09DT\UART_I2C_Viet_Ngoc_Nga\photoThang.jpg | **Huỳnh Việt Thắng**:  - Tóm tắt quá trình đào tạo, nghiên cứu (thời điểm tốt nghiệp và chương trình đào tạo, nghiên cứu): Tốt nghiệp Đại học ngành Điện Tử - Viễn Thông tháng 06/2003; Thạc sỹ Kỹ thuật Điện Tử tại Đại học Đà Nẵng tháng 12/2007; Tiến sĩ Kỹ thuật chuyên ngành Điện Tử tại Đại học Kỹ thuật Graz – Cộng hòa Áo, tháng 07/2012.  - Tóm tắt công việc hiện tại (chức vụ, cơ quan): Giảng viên, Khoa Điện tử - Viễn thông, Trường Đại học Bách Khoa, Đại học Đà Nẵng.  - Lĩnh vực quan tâm: Tính toán cấu hình lại (FPGA); Vi mạch và Hệ thống nhúng; Xử lý tín hiệu và Học máy.  - Điện thoại: +84-(0)963.100.174  - Email: [thanghv@dut.udn.vn](mailto:thanghv@dut.udn.vn) |