Implement of Photo Sticker Compound System

Siou-Hong Liou #1, Cheng-Yuan Chang #2

Department Of Electrical Engineering, National United University
2, Lienda, Miaoli 36063, Taiwan, R.O.C.

[1] nni520@gmail.com

[2] chengyuan@nuu.edu.tw

***Abstract*—The photo sticker machine has been seen in many public places, such as the photo gallery, the art center, the supermarket, the playground and so on. According to the function, these photo sticker machines can be mainly classified into two categories. One is to take a picture for the official sticker and the other is to have the sticker with the entertainments, fun and novelty-effect emoticons. Especially for the latter, the purpose of the machine is not only to attract the more number of the customers, but also to increase the rate of consumers staying at the store. No matter the former or the latter, there are two main problems in both photo sticker machines, that is, bulky appearance and higher manufacturing cost. Therefore, this paper presents a photo sticker compound system by using Raspberry Pi and Python program to improve above both issues.**

***Keywords*—photo sticker, image processing, Raspberry Pi, Python.**

1. INTRODUCTION

In everyday life, people often take pictures by utilizing the photo sticker machines due to the requirements of businesses, official applications, and entertainments, and so on. In general, these photo sticker machines can be simply classified into two categories in accordance with the function. One is to support the requirements of official stickers and the other is for the requirements of entertainments, fun and novelty-effect emoticons. However, no matter which photo sticker machines are, there are two main issues of bulky appearance and higher manufacturing cost in both machines.

With the advancement of sciences and technologies, not only the network communication services are becoming increasingly popular, but also the hardware of many systems is toward the miniaturization development. In this paper, a photo sticker compound system is proposed and implemented to improve above both issues. The hardware of the proposed system consists of a small Raspberry computer and a digital charge-coupled device (CCD) camera. The real-time image for photo sticker can be extracted via Python program, face detection technique and CCD camera. The extracted image will be further performed two steps of image processing. The extracted image can be filtered with the blurred operation, contour operation, edge enhancement operation or embossed operation. Then, the filtered image will be composited with another background which the user selects from the special effects of image options. Finally, the completed photo sticker can be sent to the user via e-mail [1]-[3].

Face detection which is the front step of face recognition can be obtained through face feature extraction. In this paper, the users do not need to further modify the size of the extracted image due to the use of the face detection. Filter functions, such as gray-scale, fog, contour, edge enhancement and relief image processing techniques, are one of the indispensable functions for modern image processing software or image applications (APPs), thus many software or APP developers have provided a lot of interesting filters to human beings for increasing the convenience when they modify the original images. The proposed system mainly improve the shortcomings of the conventional photo sticker machine, such as bulky appearance and higher manufacturing cost by simply using Raspberry computer, digital CCD camera and Python program.

2. SYSTEM ENVIRONMENT

The hardware of the proposed system mainly consists of a small Raspberry computer and a digital CCD camera. Moreover, the real-time image for the photo sticker is extracted via Python program, face detection technique and CCD camera simultaneously. Therefore, the Raspberry Pi and Python program will be briefly introduced in this section.

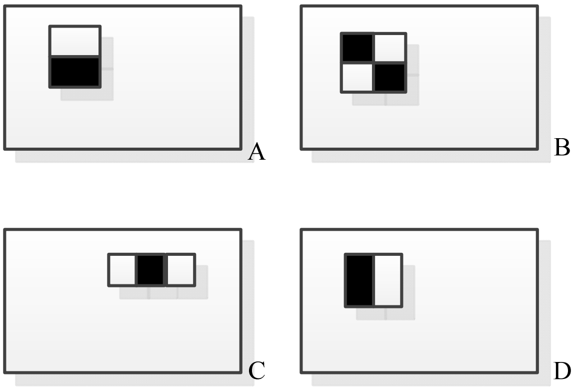


Fig. 1 Haar feature [7]

2.1. Raspberry Pi

In recent years, Raspberry Pi is widely applied in many fields, such as self-propelled vehicles, servers, personal micro-computers, smart appliances, intelligent monitoring systems, and internet of things (IoTs). Raspberry Pi which is a single computer product is proposed by the British Raspberry Pi Foundation [4]. The emergence of this product is to mainly make the students more easily obtain and learn the related computer-science knowledge and techniques with lower cost. In general, the operating system which Raspberry Pi uses is Linux, but it can be also performed under win 10 in the revision of Raspberry Pi [4]. The size of Raspberry Pi is about equal to that of a credit card, thus it can be much easier and more convenient to carry the device of Raspberry Pi than a computer. Moreover, the price of Raspberry Pi does not change as its version changes. That is, the official prices of all versions are all equal to $35, even through the hardware specification has greatly improved in each version. Especially, the latest version (i.e., Raspberry Pi 3) has the capabilities of wireless fidelity (WiFi) and Bluetooth already.

2.2. Python

Python is a literal, object-oriented computer programming language. It mainly uses the indentation instead of braces to define blocks of sentences, with the principle of simple syntax and clarity. The standard library of Python is very complete because of having a community of third-party libraries (called modules) [5], [6]. Therefore, Python is often used as a support language for many fields and applications, such as game fabrication, web page designs, data bases, image processing techniques, science computing facilities, big data and so on.

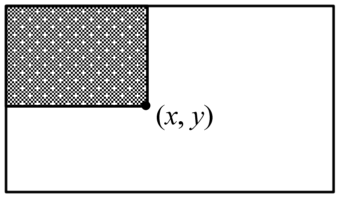


Fig. 2 Integral image [7]

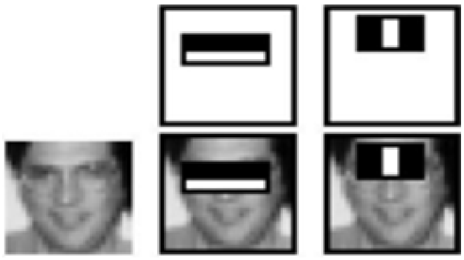


Fig. 3 Comparison of face features [7]

3. SYSTEM THEORY AND RESULTS

In this section, the principles and techniques which the proposed system adopts, such as face detection, feature extraction, cascade detection, image filter and image synthesis, are briefly introduced. Moreover, not only the flow chart of system design, but also the sample results of photo stickers are illustrated and presented.

3.1. Face Detection

In order to complete the face detection, the proposed system uses Haar cascade classifier [7] which OpenCV [8], [9] provides. Basically, face detection mainly consists of two steps: feature extraction and cascade detection. Both steps will be also interpreted as follows.

3.1.1. Feature Extraction

In [7], Viola *et al.* proposed an algorithm which can quickly carry out real-time object detection. Based on the algorithm, the proposed system uses the Haar feature shown in Fig. 1 to perform the feature extraction, where the Haar feature can be obtained from the integral image shown in Fig. 2. In Fig. 2, the value of the point (x, y) is the sum of the values of all pixels within the gray square in the upper-left corner. Fig. 3 shows the comparison of Haar features for the face feature of face detection.

The classifier can be obtained by training the positive samples (i.e., face image) and negative samples (i.e., non-face image). After above-mentioned training process, an iterative learning algorithm (called "Adaboost" algorithm) is used to combine the weak classifier and the feature

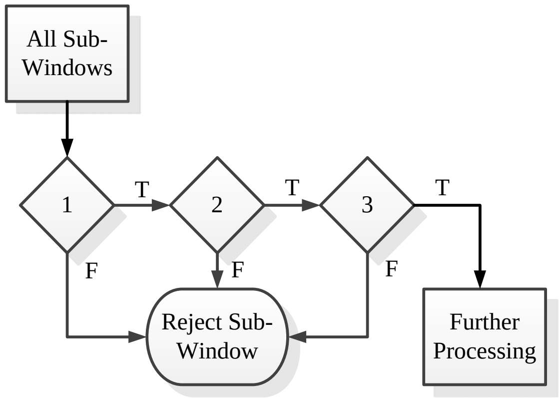


Fig. 4 Flow chart of cascade detection [7]

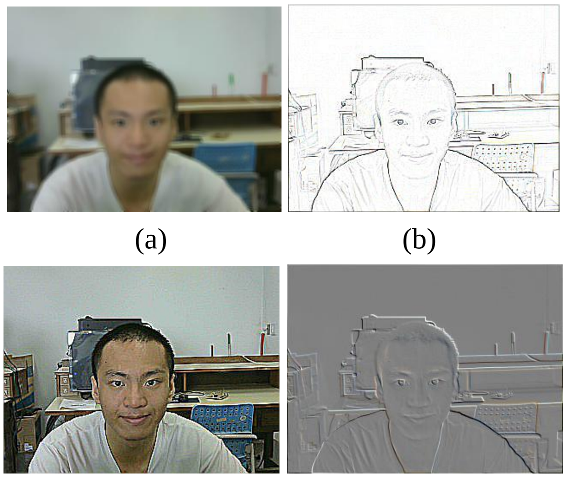
with high correct probability. Thus, it can construct a strong classifier. Finally, a cascade classifier can be obtained by combining several strong classifiers which are constructed in accordance with the above-mentioned process.

3.1.2. Cascade Detection

In order to detect whether there is a human face in the image, the image will be cut into several equal partitions first. Then, they are compared with the features via the cascade classifier. When all cascade classifiers are detected correctly, it will confirm that the human face is in the image actually.

Take three cascade classifiers as an example, the process of the detection can be modelled as a flow chart, as illustrated in Fig. 4 [7]. Each of these partitions cut from the image is inputted into the first classifier to confirm whether it is detected correctly by comparing with the features. If all partitions are detected correctly, they will be further inputted into next classifier (i.e., the second classifier) to confirm whether it is detected correctly by comparing with the features. Similarly, if all partitions are detected correctly, they will be also inputted into next classifier (i.e., the third classifier). Then, it will confirm that the human face is in the image actually until all cascade classifiers are detected correctly, which the cascade detection means.

Fig. 5 Original image



(a) (b)

(c) (d)

Fig. 6: (a) Filtered image with blurring effect; (b) Filtered image with contour effect; (c) Filtered image with edge enhancement effect; (d) Filtered image with embossing effect.

3.2. Image Filter

Image filter is usually to use an $n \times n$ square matrix (in general, 3x3 or 5x5) as the mask to apply the convolution operation to the original image, thus it can obtain the new image with the blurring, contour, edge extraction or embossing effects, based on the following equations (1)-(4), respectively [10].

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{1}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{4}$$

For example, Fig. 5 shows the original image. The blurring filter uses the mask (i.e., (1)) to apply the convolution operation to the original image, thus the new image with the blurring effect can be obtained as shown in Fig. 6 (a). Similarly, the original image also can be applied the convolution operation with the use of masks (i.e., based on (2)-(4)) to obtain the new images with contour, edge extraction or embossing effects as shown in Fig. 6 (b)-(d), respectively.

3.3. Image Synthesis

Image synthesis is the process of creating new images by replacing the certain range or certain region of the original image with the desired synthesis image [11]. In general, the region is called as region of interest (ROI). That is, ROI is the region in which we have to perform the computing operations in the image. If the position of ROI is known, the new image can be simply obtained by replacing ROI of the original image with the image matrix. However, if the position of ROI is unknown or the range of ROI is very large, the image will be painted in blue or a specific color first, and then it is performed with the same process of replacing the ROI of the original image with the image matrix. This method is very common in the filmmaking. The actors are usually photographed in the studio with a special color background, thus the computer-generated producer can simply perform the image synthesis of film effects.

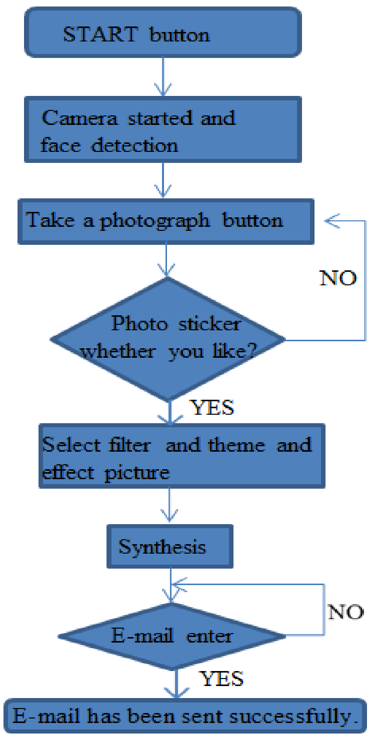


Fig. 7 Flow chart of system design

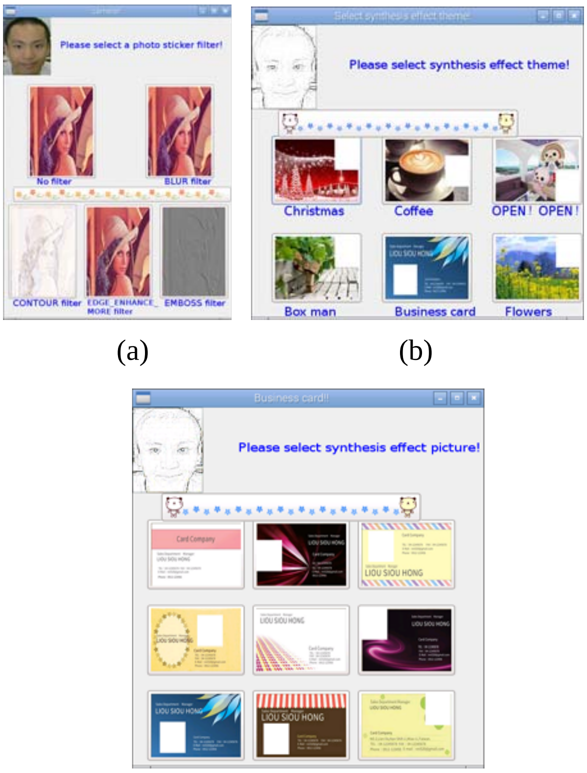
Fig. 8 Main interface of the proposed system

Fig. 9 Face detection via the camera

3.4. Flow Chart of System Design and Experiment Results

Shown in Fig. 7 is the flow char of system design of the proposed system. First, a user can press the “START” button in the main interface of the proposed system shown in Fig. 8 to start the system, thus the camera will be started and the process of face detection will be also performed automatically via the camera, shown in Fig. 9. Then, the user only has to press the “Take a photograph” button, the image of human face will be extracted. Moreover, the user can perform the extraction process by pressing the button again until he or she has been satisfied with the extracted image already.

In the next step, the user can select the category of the filter which is used for the process of the image filter in accordance with his or her preference. Fig. 10 (a) shows four filters which are used commonly for the image filter, including the blurring, contour, edge extraction and embossing filers, respectively. Then, the user can select theme background which is used for the synthesis process, shown in Fig. 10 (b), such as Christmas, coffee, business card, flower styles and so on. Shown in Fig. 10 (c) are the various types of samples in the business card background. For example, the extracted original face image is performed by contour filter first, and then the filtered image is further compound with the theme background (i.e., the business card backgrounds shown in Fig. 10 (b) and (c)) which user selects. Fig. 11 shows a business card which is composed with the complete synthesis image. Finally, in order to make the user collect the complete synthesis productions conveniently, he or she can upload them directly via the interface of entering the e-mail shown in Figs. 11 and 12, and enter his or her email simply.



(a) (b)

(c)

Fig. 10 (a) Various types of filters; (b) Various types of theme backgrounds; (c) Various types of samples in the business card background

Fig. 11 Synthesis image with the business card background

Fig. 12 Interface of entering e-mail

4. Conclusions

This paper proposed a photo sticker com pound system which is implemented with the use of Raspberry Pi 2 and Python programs, thus it can improve both design issues of product size and manufacturing cost. However, the internet function of the proposed system is only supported via wired internet at present. Therefore, the proposed system will be implemented by using Raspberry Pi 3 in the next version, because WiFi function will be built in Raspberry Pi 3 directly to increase the convenience in the use of internet function and without additional purchase of wireless card. Moreover, the proposed system can further use the flat screen instead of the traditional screen, thus this system can become more portable and more miniaturization.

Acknowledgment

This work was supported by the Ministry of Science and Technology of Republic of China under Grant MOST 105-2221-E-239-003.

References

[1] Matt Richardson, Shawn Wallace, *Getting Started with Raspberry Pi*, Maker Media Inc, 2012.

[2] Eben Upton, Gareth Halfacree, *Raspberry Pi User Guide*, 2nd ed, Wiley, 2014.

[3] Maik Schmidt, *Raspberry Pi A Quick-Start Guide*, Pragmatic Bookshelf, 2012.

[4] Peter Membrey, David Hows, *Learn Raspberry Pi with Linux*, Apress, 2013.

[5] Mark Lutz, *Programming Python*, O'Reilly Media, 4th ed, 2010.

[6] Mark Lutz, *Learning Python*, O'Reilly Media, 5th ed, 2013.

[7] Paul Viola, Michael J. Jones, “Robust Real-Time Face Detection,” *International Journal of Computer Vision*, vol. 57, no. 2, pp.137-154, 2004.

[8] Robert Laganiere, *OpenCV Computer Vision Application Programming Cookbook*, Packt Publishing, 2nd ed, 2014.

[9] Joe Minichino, Joseph Howse, *Learning OpenCV3 Computer Vision with Python*, Packt Publishing, 2nd ed, 2015.

[10] Jan Erik Solem, *Programming Computer Vision with Python*, O'Reilly, 1st ed, 2012.

[11] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, *Digital Image Processing Using MATLAB*, New Jersey: Pearson Prentice Hall, 2004.