

OFFICE ENTRANCE CONTROL WITH FACE RECOGNITION

¹ Yun-Che Tsai (蔡昀哲), ² Chiou-Shann Fuh (傅楸善)

¹ Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan
E-mail: jpm9ie8c@gmail.com

² Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan
E-mail: fuh@csie.ntu.edu.tw

ABSTRACT

A method of face recognition and a door access control system is proposed in this thesis. We combine uniform local binary pattern and Adaboost for face recognition. We achieve acceptable recognition rate and computation time for door access control system. A door control system uses several pieces of simple hardware so that everyone can build the system easily. By using horizontal Sobel operator, a simple eyeglasses detection method is also proposed to judge if tested person wears eyeglass or not. It may apply to other applications.

Keywords Face Recognition; Eyeglasses Detection; Adaboost; Uniform Local Binary Pattern;

1. INTRODUCTION

Over the several decades, face recognition has got wide attention from either science or industry due to its various applications such as surveillance, access control, and identity verification. Typically, face recognition system contains three steps: face detection which extracts faces in an image, feature extraction which applies feature extraction schemes such as Local Binary Pattern [14] or Gabor Wavelet [3] to face image to further enhance details on facial region, and classification which judges if two face images belong to the same person.

There are three issues of face recognition system to be considered: pose, illumination variation, occlusion, and execution time. Pose issue means that the poses of faces in images may vary. The same person may look quite different under various poses. Similarly, the same person may look quite different under various illumination. The most difficult of illumination problems is direction of light source, because different directions of light source projected on faces cause various shadows. These shadows may make a person look different. Occlusion problem means that one may wear accessories such as eyeglasses or gauze masks. This will lose much

information of faces. Execution time is the time consumed by face recognition system. In some cases such as surveillance and access control system, execution time is a critical issue. Much time consumed by face recognition system may make this kind of systems impractical.

Several classical methods are introduced. Local feature-based methods [1, 7] apply feature extraction schemes to describe the faces and further classify them by various methods. Sparse representation methods [15, 17] utilize sparsity in face feature space. Faces of the same person in feature will be very similar. If the feature space is too large, faces of the same person are far away from each other. Linear regression methods [13] consider that features of the same person lie on roughly the same subspace. They use least square estimation to estimate where probe faces belong. These methods are one kind of Nearest Subspace Classification.

The remaining of this thesis is organized as follows: Chapter 2 introduces several techniques related to face recognition. Chapter 3 introduces our method to build our face recognition system. Chapter 4 illustrates the system setup of our entrance control system including how we control the door lock. Chapter 5 presents some experimental data we have conducted to illustrate which methods are suitable. Chapter 6 contains the summary of our work and experimental results and the future work. The last chapter lists references.

2. RELATED WORKS

2.1. Local Binary Pattern

Local Binary Pattern (LBP) was first introduced by Ojala in 1994 [14]. It is widely used in computer vision area for texture representation. Not only for face recognition but also other tasks such as age and gender estimation, LBP is adopted to represent face and achieves good results. How LBP works is shown in Fig. 1. Central pixel is compared with eight neighbor pixels around it. If value

of one neighboring pixel is greater than the value of central pixel, we set its value as 1. Otherwise, we set its value as zero. After applying such operation to eight neighboring pixels, we transform values of neighboring pixels ranging from 0 to 255, which is the pixel value of 8-bit gray image, to binary values (0 or 1). We can view binary values around center pixel as 8-bit binary sequence, starting from top-left pixel, going through the remaining 7 pixels clockwise. Finally, we transform 8-bit binary sequence into decimal number and assign this value to central pixel.

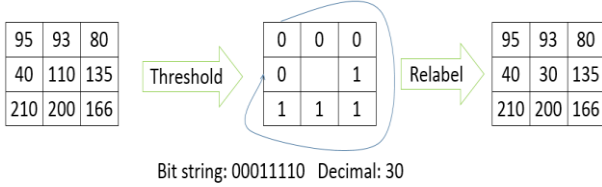


Fig. 1: The process of LBP operator.

One extended form of LBP is uniform Local Binary Pattern [1]. It is inspired by the fact that some output values of LBP operator appear much less frequently than the others. A local binary pattern is called uniform if the output binary string of LBP operator contains at most two bitwise transitions from 0 to 1 or from 1 to 0. For instance, binary strings 00000000, 00111000, and 00011111 are uniform because they have at most two bitwise for 0 to 1 or from 1 to 0. Binary strings 00110011, 01010101, and 11010010 are not uniform because they have more than two bitwise transitions. The number of output values of uniform LBP is 59. It contains uniform and non-uniform binary strings. The number of uniform binary strings is 58 (0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 38, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126, 127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 241, 249, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254, 255), and the rest of numbers are non-uniform strings. The uniform LBP assigns 58 uniform strings to value ranging from 0 to 57, and assigns all the non-uniform binary string to value 58. Thus the total number of uniform LBP's value is 59. There are two advantages of uniform LBP. One is that it reduces the number and thus dimension of LBP values from 256 to 59. The length of vector after applying histogram is much less compared with original LBP. Another advantage is that it resists noise to a certain degree, because some random noises have more than two bitwise transitions.

2.2. Adaptive Boosting (Adaboost)

Adaboost is a machine learning meta-algorithm proposed by Yoav Freund and Robert Schapire [2]. In some problems, it can be less susceptible to the overfitting problem than other learning algorithms.

The basic concept is to combine a sequence of other learning algorithm (referred to as weak classifier), and the decision of weak classifiers is viewed as the result of strong classifier. The subsequent weak classifiers favor instances misclassified by previous classifiers. The individual weak classifier can be very weak as long as the

result of the weak classifier is better than random guessing.

First, we have training samples defined as (x_i, y_i) , $i = 1, \dots, l$, where $x_i \in R^n$ as descriptors vector and $y_i \in \{-1, 1\}$ as labels of descriptors. Initially, we weigh every sample as equal. We denote weight as D .

$$D_t(i) = \frac{1}{m}, i = 1, \dots, m$$

where m is the number of training samples, and t is the index of weak classifier. Next, Boosting procedure starts: for $(t=1, \dots, T)$ { // T is the number of weak classifier for $(j=1, \dots, J)$ {

calculate error $\epsilon_j = \sum_{i=1}^m D_t(i)$, where $y_i \neq h_j(x_i)$
// $h_j()$ is weak classifiers.

$$h_t = \underset{h_j \in H}{\operatorname{argmin}} \epsilon_j$$

$$\alpha_t = 1/2 \ln \frac{1-\epsilon_t}{\epsilon_t}$$

$$D_{t+1}(i) = \frac{D_t \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

}
At testing stage, the strong classifier is defined:

$$H(x) = \operatorname{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

where x is testing sample

2.3. Histogram Comparison

There are several ways to calculate the similarity of two vectors. In terms of histogram comparison, we compare two histograms and measure the similarity of two histograms. There are four common ways of histogram comparison: cosine distance, correlation, chi-square distance, and histogram intersection.

Cosine distance measures the similarity of two vectors based on inner product of two vectors. Cosine distance is formulated as:

$$\operatorname{cosDist}(H_1, H_2) = \frac{H_1 \cdot H_2}{\|H_1\| \|H_2\|} = \frac{\sum_{i=1}^n H_{1i} * V_{2i}}{\sqrt{\sum_{i=1}^n H_{1i}^2} * \sqrt{\sum_{i=1}^n H_{2i}^2}}$$

where H_{ki} is element of vector (or histogram); n is the length (number of bins) of vector (histogram).

Correlation is formulated as:

$$\operatorname{correlation}(H_1, H_2) = \frac{\sum_{i=1}^n (H_{1i} - \bar{H}_1) * (H_{2i} - \bar{H}_2)}{\sqrt{\sum_{i=1}^n (H_{1i} - \bar{H}_1)^2 * \sum_{i=1}^n (H_{2i} - \bar{H}_2)^2}}$$

where H_{ki} is element of vector (or histogram); n is the length (number of bins) of vector (histogram); and \bar{H}_k is mean of H_k and is defined as $\bar{H}_k = \frac{1}{n} \sum_{i=1}^n H_{ki}$.

Chi-squared distance is formulated as:

$$\operatorname{ChiSquared}(H_1, H_2) = \sum_{i=1}^n \frac{(H_{1i} - H_{2i})^2}{H_{1i} + H_{2i}}$$

where H_{ki} is element of vector (or histogram); n is the length (number of bins) of vector (histogram).

Histogram intersection calculates similarity of two histograms (vectors) by comparing corresponding bins of two histograms and saving the bin whose value is less than the other:

$$HistogramIntersect(H_1, H_2) = \sum_{i=1}^n \min(H_{1i}, H_{2i})$$

where H_{ki} is element of vector (or histogram); n is the length (number of bins) of vector (histogram).

2.4. Eyeglasses Detection

We found that eyeglasses will influence results of face recognition dramatically, because eyeglasses become strong features of faces, especially thick trimmed eyeglass, and specular reflection of light tube may show up at different locations depending on head pose. It means that two different persons with the same thick trimmed glass may be judged as the same person. For an access control system, we have to avoid this situation automatically, so we propose a quite simple way to detect eyeglasses.

We can see that if a person wears eyeglasses, there are strong edges between after applying horizontal Sobel operator ((D), (E), and (F) in Fig. 2). By calculating the intensity of pixels between two eyes, we know if the person wears eyeglass or not.

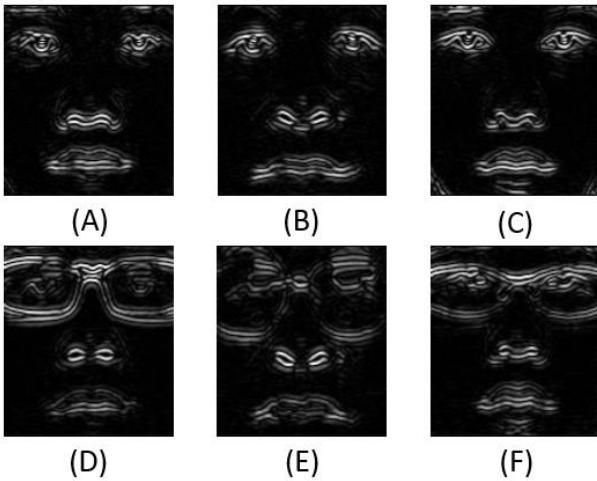


Fig. 2: (A) and (D) are the same person. (B) and (E) are the same person. (C) and (F) are the same person. (A), (B), and (C) are different persons without eyeglass. (D), (E), and (F) are different persons wearing eyeglasses.

3. METHODOLOGY

Our proposed methodology for office access control by face recognition contains certain parts including cropping a face from image, illumination normalization, uniform LBP operation, histogram, and Adaboost (Fig. 3). The system first crops face from input image and normalizes its position to standard coordinate. The cropped face is processed by a series of illumination normalization procedure to reduce the influence of illumination variation. The normalized image is processed by uniform local binary pattern and histogram to turn the image into descriptor vector. At training stage, the descriptor vector is compared with another descriptor vector produced by other people. If the two descriptor vectors belong to the same people, we call it intra-face. Otherwise, we call it

extra-face. The intra-faces and extra-faces are trained by Adaboost, and we get trained model of testing stage. At testing stage, two descriptor vectors are compared. One descriptor vector is produced by the input image and the other is produced by images in database. We put the comparison result into Adaboost to estimate if these two persons are the same or not.

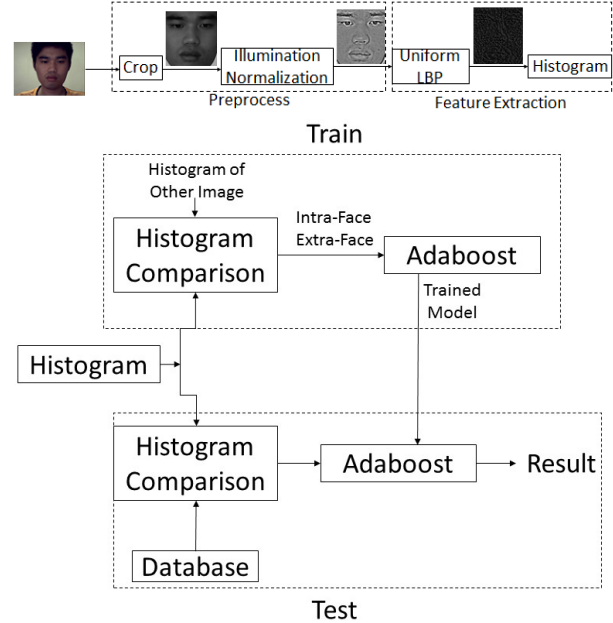


Fig. 3: Flowchart of our face recognition system.

3.1. Crop Face from Image

The first step of face recognition is to crop face from image and normalize its position to the standard coordinate. This step contains face detection, face landmark, and moving face in image to standard coordinate.

Face detection aims to find where the face is in an image. Face landmarking is to find out specific points on face, such as nose, eyes, eyebrows, and so on. We do face detection and landmarking by a C++ library called IntraFace [18].

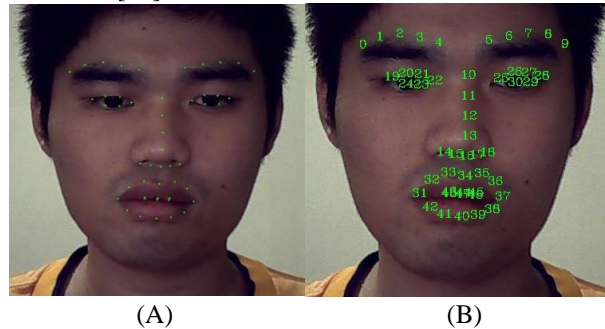


Fig. 4: (A) All the points the IntraFace C++ library marked [18]. (B) All points have unique ID, we can refer to 2D coordinates by unique IDs.

Once we obtain the 2D coordinates, there are two ways to align face in an image to standard coordinates. The first is to align image by rotation, scaling, and translation. The second is to warp the image. The advantage of first

method is its speed. The advantage of second method is that it can align all the points precisely. By research of [19], it suggests that we should perform rotation, translation, and scaling to align the face instead of warping. Although warping can align feature points on face to standard face precisely, it may lose some information. For example, the distance between mouse and nose is different from one to another. After applying warping, the distance between mouse and nose for all faces becomes the same.

We align an image using rotation, translation, and scaling by two points, where their IDs are 19 and 26 in Fig. 4(B). These two points represent two external canthi. Our target image is 190 pixels in height and 160 pixels in width and two coordinates of external canthi are (20, 45), (140, 45) in form of (x, y), not (r, c).

After applying rotation, translation, and scaling by two external canthi, we resize the height of target image in Fig. 5. The bottom line of target image is 10 pixels down below the bottom of mouth (ID = 40 in Fig. 4). If we skip this step and the face pose of the person in image heads down, the target image will contain many regions which do not belong to face, such as neck.

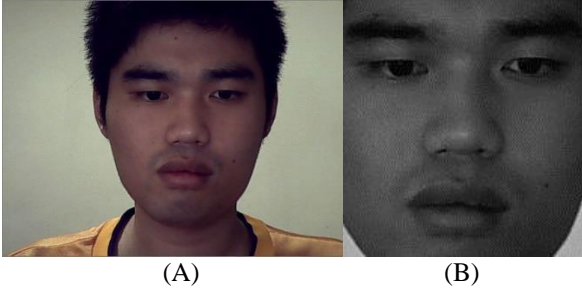


Fig. 5: (A) Input color image. (B) Cropped gray-scale image by rotation, translation, and scaling.

3.2. Illumination Normalization

Faces illuminated by various light conditions may look quite different. Before we recognize the image, we have to reduce the influence of different light conditions.

We follow the steps introduced by [16] which contains three parts: gamma correction, difference of Gaussian (DoG), and contrast equalization.

Gamma correction is a non-linear image gray-level transformation. It can be formulated as:

$$I'(x, y) = I(x, y)^\gamma$$

where $I(x, y)$ is intensity of image at (x, y) .

If γ is less than 1 and greater than 0, details in bright regions are compressed while enhancing details in dark regions. The γ value in [16] is 0.2.

After gamma correction, difference of Gaussian is applied. DoG is first applied two Gaussian filters with different σ to the same image. The results are two images containing different frequency information. Simply subtract two Gaussian images, we can get DoG image. The DoG is formulated as:

$$f(u, v, \sigma_1, \sigma_2) = \frac{1}{2\pi\sigma_1^2} \exp^{-\frac{(u^2+v^2)}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} \exp^{-\frac{(u^2+v^2)}{2\sigma_2^2}}$$

We use σ_1 and σ_2 of 1.0 and 2.0.

Finally, contrast equalization contains three actions:

$$I(x, y) = \frac{I(x, y)}{(\text{mean}(|I(x', y')|^\alpha))^{1/\alpha}}$$

$$I(x, y) = \frac{I(x, y)}{(\text{mean}(\min(\tau, |I(x', y')|^\alpha))^{1/\alpha}}$$

$$I(x, y) = \tau * \tanh(I(x, y)/\tau)$$

where $\text{mean}(|I(x', y')|^\alpha)$ comprises doing $I(x, y) = |I(x', y')|^\alpha$ first, then calculating the mean of whole image pixels.



Fig. 6: Flowchart of illumination normalization by [16]. DoG operator generates abnormal lines on four sides of image due to Gaussian filter on the margin of image. We follow the recommendation of [16] to use constant padding when applying Gaussian filter. We can see that it generates abnormal lines (about 5 pixels in width) on four sides of image in Fig. 7. We do not want abnormal lines to influence the result so we remove them. After illumination normalization, we resize the image into 144 pixels in height and 128 pixels in width.

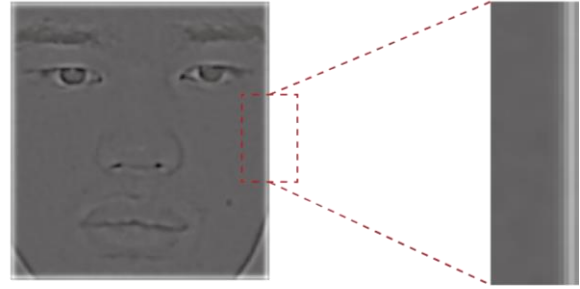
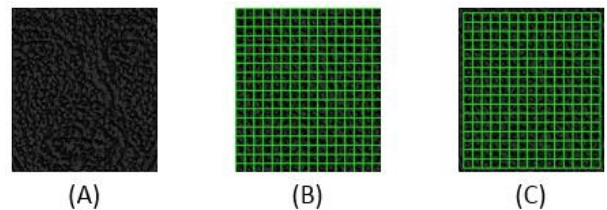


Fig. 7: After DoG operator, four abnormal lines on the sides of image are generated.

3.3. Uniform Local Binary Pattern and Histogram

After illumination normalization, we need to apply some operators to enhance representation for original image. We choose uniform local binary pattern due to its lower dimension compared with other operators.

After uniform LBP, we partition image into rectangular non-overlapping regions. We use two different types of partition to gain more information in Fig. 8. For each non-overlapping region, we calculate its histogram so that the region is transformed into a vector which is 59 in dimension. Each grid in Fig. 8 is 8x8 pixels and the number of total grids is 543. At last, we merge all the vectors into a descriptor vector. The descriptor vector is 543*59=32,037 in dimension.



(A)

(B)

(C)

Fig. 8: (A) Image after applying uniform LBP operator. (B) and (C) Two different partitions of the images. Once they are partitioned, we calculate the histogram of each grid.

3.3. Classification

This section contains two parts: one is histogram comparison and the other is classifier.

Once we obtain histograms of two images, for each pair of corresponding histograms (histograms that have same geometry in two images), we apply histogram comparison methods to generate a similarity score (a floating number) in Fig. 9. We use cosine distance as our histogram comparison method. Results generated by all pairs of corresponding histograms are collected to form a descriptor. Since we partition image into 543 non-overlapping regions, the descriptor is 543 in dimension.

At training, if the descriptor is made from the same person, inspired by [錯誤! 找不到參照來源。](#), we call it intra-face, otherwise, we call it extra-face. The descriptors of intra-faces and extra-faces are trained using Adaboost to learn the characteristics of intra-faces and extra-faces.

At test stage, the descriptor is judged by trained Adaboost to verify who the descriptor represents, intra-faces or extra-faces. If the result is intra-faces, it means the two images forming the descriptor are the same person, otherwise, they are different persons.

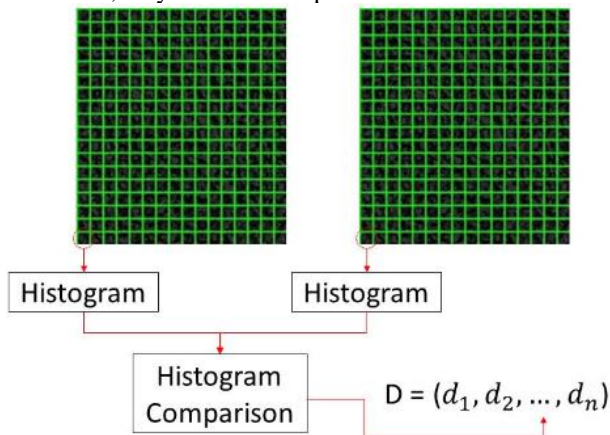


Fig. 9: Corresponding regions of two images are extracted. We calculate histogram for each region. Two histograms are compared with Histogram comparison methods with cosine distance. Results (d_k) generated by all pairs of corresponding histogram are collected to form descriptor D ; n is the length of descriptor $D=543$ in our case.

4. SYSTEM SETUP

Face recognition technology helps identify whether people belong to this office or not. Other devices are needed for door access control. Table 1 lists all devices to control access of our laboratory.

Table 1: Devices for door access control.

Notebook	
•	ASUS X455LD
•	CPU: Intel i5-4210U (1.7GHz-2.7GHz)
•	RAM: 4G
•	OS: Windows 7
Raspberry Pi Rev 2 Model B	
•	CPU: Broadcom BCM2835 700 MHZ
•	RAM: 512MB SDRAM
•	OS: RASPBIAN
Keyes 2-Channel Relay Module	
Electric Power Bolt Lock DA-62A/D	

Raspberry Pi is a series of credit card-sized single board computers developed in the UK by the Raspberry Pi Foundation. It is intended to help people learn more about programming and computer hardware. We use Raspberry Pi Rev 2 Model B due to low cost, high performance, and versatility. Fig 10 shows the hardware on Raspberry Pi Rev 2 Model B. The key component on Raspberry Pi is General-Purpose Input Output (GPIO), which gives order to relay to unlock the door and detects status of the door (open or closed).

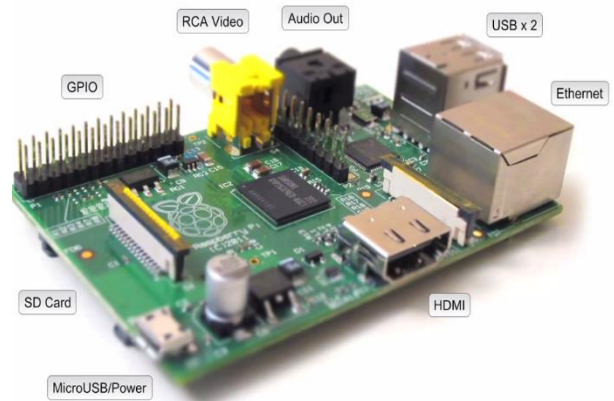


Fig. 10: Hardware on Raspberry Pi Rev 2 Model B. HDMI: High-Definition Multimedia Interface. USB: Universal Serial Bus. RCA: Radio Corporation of America. GPIO: General-Purpose Input/Output. SD card: Secure Digital card.

General purpose IO (GPIO) is a row of pins with various functions (Fig. 11). It can supply power to relay with 5V and GND pins. It can also change status of relay by switching voltage of pins of GPIO.

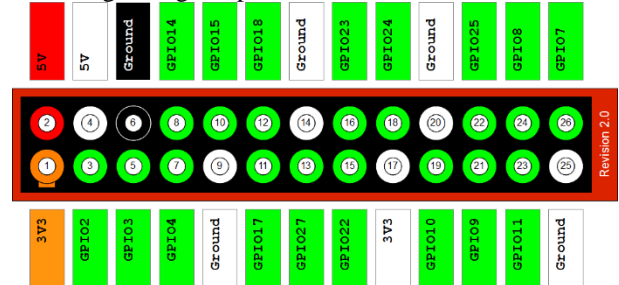


Fig. 11: Definition of GPIO on Raspberry Pi Rev 2 Model B. Pin 2 can supply 5V current to other devices, such as relay. Pin 12 can change status of relay by switching its voltage.

A relay is an electrically operated switch. Relays are used to control a circuit by low-power signal. In our proposed

system, low-power signal is generated by GPIO which controls the behavior of lock via a relay.

Typically, relay module has three entries, COM (common), NO (normally open), and NC (normally close). These three entries are connected to controlled circuit. COM connects to either NC or NO. When not applying voltage to relay, COM is connected to NC. When applying voltage to relay, COM is connected to NO.



Fig. 12: Keyes 2-module relay.

Electric power bolt lock in Fig. 13 has three interfaces which are white, red, and blue in color respectively. Each of the interfaces has two pins. The white one is connected to a Direct Current (DC) power supply. The red one allows to detect the status of the lock (open or closed). The blue one is for control of the lock. If we activate the circuit of blue interface to be on (off), the lock is open (closed).

We use six interfaces on the relay in Fig. 13 to control the lock (NC, COM, NO) and receive order from Raspberry Pi (IN1, GND, VCC). VCC and GND are connected to Pins 4 and 2 of Raspberry Pi. It is for Raspberry Pi to supply current to relay. IN1 is connected to Pin12. If we apply voltage on IN1 through raising voltage level of pin 12, COM-NC are disconnected while COM-NO are connected. Two wires on the lock of blue interface are connected to COM-NC. Normally, we apply high voltage to IN1 so that COM-NC are disconnected. If face recognition system gives order on opening the lock, Raspberry Pi turns off the voltage of Pin 12 leading to COM-NC to be connected, so that the lock is opened. We connect two wires of blue interface on the lock to COM-NC, because we want to make sure if Raspberry Pi shuts down accidentally, the voltage of pin 12 is 0 and the lock is opened. If we connect two wires to COM-NO, the lock will be closed forever when Raspberry Pi shuts down.

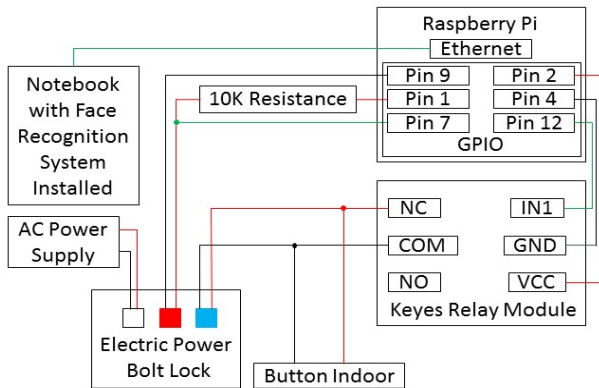


Fig. 13: The structure of the whole access control system. Red lines are hot wires, and black lines are ground wires. Green wires are cables of no special meanings. Button indoor is a button set inside the door. If we want to go outside the office, we have to push the button to unlock the door.

5. EXPERIMENTAL RESULTS

This chapter shows some experimental results. Moreover, it is organized as follows: Section 5-1 introduces face databases we use to train models for recognition. Section 5-2 evaluates four histogram comparisons methods introduced in Section 2-3. Section 5-3 discusses whether we should keep eyebrows while cropping faces. Section 5-4 evaluates the simple way to detect whether a person wears eyeglasses or not in Section 2-4. Section 5-5 evaluates our access control system and compares with a business package called Luxand.

5.1. Face Databases

We collect face images from 8 well-known face databases for training including AR [12], BioID, Caltech Faces [5], CIE Biometrics 錯誤! 找不到參照來源。 , CK+ (The Extended Cohn-Kanade Dataset) [9], EYFDB (The Extended Yale Face Database B) [4], FEI, and JAFFE (The Japanese Female Facial Expression Database) [11]. We collect images which contain frontal faces with little variation in head pose. All images are without eyeglasses. There are total 7,860 images and 500 subjects. They can produce 126,506 intra-faces. As for extra-faces, we tried different amount of extra-faces to get best results. Too many extra-faces may lead to high false rejection rate. Too few extra-faces may lead to high false acceptance rate.

5.2. Evaluation about Histogram Comparison Methods

There are four histogram comparison methods introduced in Section 2-5, including cosine distance, correlation, chi-square, and intersect. We evaluate the performance of these four methods by linear SVM and Adaboost. SVM has many kernel types. We use linear kernel because of system speed. Table 2 uses 126,506 intra-faces and 123,097 extra-faces for training. We train 80% of training samples and 20% for cross validation for both Adaboost and linear SVM.

Table 2: Accuracy of four histogram comparison methods under Adaboost and linear SVM with eyebrow. There are 600 weak classifiers and the maximum depth is 5 when we train Adaboost. We can observe that cosine distance and correlation methods are superior to chi-square and Intersect methods both in the cases of Adaboost and linear SVM.

	Cosine Distance	Correlation	Chi-Square	Intersect
Adaboost	98.87%	99.37%	97.24%	97.24%
Linear SVM	91.50%	92.06%	83.78%	90.94%

Table 3: Accuracy of four histogram comparison methods under Adaboost and linear SVM without eyebrow.

	Cosine Distance	Correlation	Chi-Square	Intersect
Adaboost	98.67%	98.39%	95.91%	98.27%
Linear SVM	90.42%	91.32%	80.18%	87.87%

We choose Adaboost, because it wins over linear SVM in accuracy of our case. Besides, Adaboost is really quite fast method which is suitable for our access control system.

We choose 126,506 intra-faces and 2,478,657 extra-faces as our training sample of Adaboost. The training error is 0% and the testing error is 0.24%. We use 600 weak classifiers and the maximum depth is 5 to be parameters of Adaboost.

5.3. Retaining Eyebrows or Not

Retaining eyebrow while cropping faces from images may contain more information, but it may contain more noise due to hair. The training setup of Table 3 is the same as Table 2, except that Table 3 uses face images which do not contain eyebrow. We observe that the accuracy without eyebrow decreases about 1% to 3% in both cases, Adaboost and Linear SVM, compared with Table 2, except for the case of Intersect with Adaboost. By the experimental result, we should keep eyebrow while cropping faces.

5.4. Performance on Eyeglasses Detection

Wearing eyeglasses may significantly influence results of face recognition. Considering access control system, we cannot hire a person to ask people to take off their glasses while testing, so we have to develop a mechanism to automatically detect whether people being tested wear eyeglasses or not.

We find that if someone wears eyeglasses, there is high intensity value between two eyes after horizontal Sobel operator in Section 2-4.

The inputs of horizontal Sobel operator are images after cropping and illumination normalization. Note that the output image of horizontal Sobel is 144 pixels in height and 128 pixels in width due to our cropping method in Section 3-1. The range of pixels we sample are 10 to 40 pixels in height and 63 to 64 pixels in width, so we sampled $(40-10+1)*2=62$ pixels. At training, we randomly choose 30 images for each case, with and without eyeglass. The result is as follows:

Table 4: Intensity sum of pixels after horizontal Sobel operator. Images with glasses have higher gradient magnitude due to much stronger horizontal edge caused by glasses.

	Maximum	Minimum	Mean	Standard Deviation
With Glasses	19519	4491	10733.2	2895.25
Without Glasses	2901	363	990.52	341.546

We think that the purpose of eyeglasses detection is for whom wearing eyeglasses, so we set threshold as 2,700. If the intensity sum of gradient magnitude is above 2,700,

we think the person being tested wears eyeglasses and we will ask the person to take glasses off.

At testing, there are 1,499 images being tested. The false rejection rate (judging person without eyeglasses as one with eyeglasses) is 0.4% and the false positive rate (judging person with eyeglasses as one without eyeglasses) is 0.0667%.

5.5. Comparison with Luxand

Luxand is a private hi-tech company formed in 2005 [10]. Luxand research activities began with Artificial Intelligence and biometric identification technologies, allowing the company to develop a complete set of tools and libraries to perform fully automatic recognition of human faces and facial features.

We test our system and Luxand business package for about 90 days. There are 15 people registered as our laboratory members. There are totally 957 visits (about 4,785 sec.) to our system and there are 69 unknown persons and 888 laboratory member visits among them. Some of visitors are difficult for the system due to slanted face, smiling or sad expression, and open mouth. Without glasses, False Acceptance Rate (FAR) of our system is 0% and False Rejection Rate (FRR) is 9.50%. Without glasses, false acceptance rate of Luxand is 0% and false rejection rate is 6.66%. The average operation time of our system is 104.64 ms. The average operation time of Luxand is 212.67 ms. The time is measured by Intel I7-4790k at 4.5GHz.

6. CONCLUSION AND FEATURE WORKS

In this thesis, we present a face recognition system and office door access control system. There are two considerations of door access control system: speed and recognition rate. By combining uniform local binary pattern for describing faces and Adaboost for recognition, we achieve the same zero FAR and slightly higher FRR but much faster speed of judging one person than Luxand. With glasses, false acceptance rate of our system is 4.28% and false rejection rate is 14.66%. With glasses, false acceptance rate of Luxand is 0.66% and false rejection rate is 5.08%. The average operation time of our system is 81.96 ms. The average operation time of Luxand is 194.59 ms. The time is measured by Intel I7-4790. Although our system is still faster, we need to improve our FAR and FRR in the future.

Recognition on people wearing glasses is difficult in face recognition, because eyeglasses become strong features on faces. Due to limited databases of people with eyeglasses, we cannot deal with people with eyeglasses effectively. Therefore, we introduce a quite easy way to judge whether tested person wears eyeglasses or not. If tested person wears eyeglasses, our system does not recognize him and asks visitor to remove glasses. It improves security of our system. Although performance of our system is not superior to Luxand in FRR, we

consumes less time about a half than Luxand does and the false accept rate is the same.

The major problem to be solved is recognition with people wearing eyeglass. As a door control system, asking visitor to take off their eyeglasses may be annoying. There are barely any databases contain sufficient images containing people wearing eyeglasses. We try to collect faces with eyeglasses, there are 293 subjects only, which means it have no sufficient intra-faces and extra-faces. Although we have 500 different people as our training samples, we think it is still too few. If we can obtain more training samples, we can achieve better performance.

ACKNOWLEDGEMENT

This research was supported by the Ministry of Science and Technology of Taiwan, R.O.C. and under Grants MOST 103-2221-E-002-188.

REFERENCES

- [1] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face Recognition with Local Binary Patterns," *Lecture Notes in Computer Science*, Vol. 3021, pp. 469-481, 2004.
- [2] Y. Freund and R. E. Schapire, "A Desicion-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Proceedings of European Conference on Computational Learning Theory*, Barcelona, Spain, pp. 23-37, 1995.
- [3] D. Gabor, "Theory of communication," *Transactions on Institution of Electrical Engineers*, Vol 93, No 26, pp. 429-441, 1946.
- [4] A. S. Georghiades, P. N. Belhumeur, and D. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 6, pp. 643-660, 2001.
- [5] O. Jesorsky, K. Kirchberg, and R. Frischolz, "Robust Face Detection Using the Hausdorff Distance," In J. Bigun and F. Smeraldi, editors, *Audio and Video Based Person Authentication*, pp. 90-95, Springer, 2001.
- [6] A. Kasiński and A. Florek, A, "The Put Face Database," *Schmidt Image Processing & Communications*, Vol. 13, No. 3-4, pp. 59-64, 2008.
- [7] Z. Lei, D. Yi, and S. Z. Li, "Discriminant Image Filter Learning for Face Recognition with Local Binary Pattern Like Representation," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, pp. 1063-6919, 2012.
- [8] S. Z. Li, R. F. Chu, S. G. Liao, and L. Zhang, "Illumination Invariant Face Recognition Using Near-Infrared Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 4, pp. 627-639, 2007.
- [9] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The Extended Cohn-Kanade Dataset (CK+): A Complete Dataset for Action Unit and Emotion-Specified Expression," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, San Francisco, CA, pp. 94-101, 2010.
- [10] Luxand, "About Luxand," <https://www.luxand.com/>, 2015.
- [11] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba, "Coding Facial Expressions with Gabor Wavelets," *Proceedings of IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, pp. 200-205, 1998.
- [12] A. M. Martinez and R. Benavente, "The AR Face Database," CVC Tech. Report #24, 1998.
- [13] I. Naseem, R. Togneri, and M. Bennamoun, "Linear Regression for Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, No. 11, pp. 2106-2112, 2010.
- [14] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions," *Proceedings of IAPR International Conference on Pattern Recognition*, Stockholm, Sweden, Vol. 1, pp. 582 - 585, 1994.
- [15] E. G. Ortiz, A. Wright, and M. Shah, "Face Recognition in Movie Trailers via Mean Sequence Sparse Representation-Based Classification," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Portland, Oregon, pp. 3531-3538, 2013.
- [16] X. Y. Tan and B. Triggs, "Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions," *IEEE Transactions on Image Processing*, Vol. 19, No. 6, pp. 1635-1650, 2010.
- [17] A. Wagner, J. Wright, A. Ganesh, Z. H. Zhou, H. Mobahi, and Y. Ma, "Toward a Practical Face Recognition System: Robust Alignment and Illumination by Sparse Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 2, pp. 372-386, 2012.
- [18] X. Xiong and F. De la Torre, "Supervised Descent Method and Its Application to Face Alignment," *Proceedings of IEEE International Conference on Computer Vision*, Sydney, NSW, pp. 532 - 539, 2013.
- [19] J. Zou, Q. Ji, and G. Nagy, "A Comparative Study of Local Matching Approach for Face Recognition," *IEEE Transactions on Image Processing*, Vol. 16, No. 10, pp. 2617-2628, 2007.