Development of Intelligent Humanoid Robot with Face Recognition Features

Christian Tarunajaya[1], Oey Kevin Wijaya[1], Reinard Lazuardi Kuwandy[1], Heri Ngarianto[1], Alexander A S Gunawan[1], and Widodo Budiharto[1]

Abstract—**In this paper, we would like to present the development of intelligent humanoid robot system. Our previous research have designed humanoid robot which could receive command and do speech recognition based on Bioloid GP robot and Raspberry Pi 2 as controller. In here, the robot system is extended to be able to detect and recognize human face. Based on OpenCV and Python, Viola-Jones method is used for face detection and PCA (Principal Component Analysis) for face recognition. Viola-Jones method provides a fast and robust framework for extracting and detecting face features. While, PCA or frequently called as eigenfaces is one of the algorithms to recognize face by reducing the image dimension using its eigen values. Generally, face recognition is influenced by many variables like light intensity, poses, and expression. By controlling these variables, the result of face recognition accuracy is about 93% (28 correct recognition of 30 experiments). Based on the recognized face, our humanoid robot gives the response action.**

Keywords— **humanoid robot, Bioloid GP, Raspberry Pi 2, Viola-Jones, PCA, face detection, face recognition.**

Abstrak— **Dalam makalah ini, kita akan mengembangkan sistem cerdas dari robot humanoid. Dalam riset sebelumnya, telah didesain robot humanoid yang dapat menerima perintah dan melakukan pengenalan ucapan berbasis robot Bioloid GP dan pengendali Raspberry Pi 2. Pada penelitian kali ini, sistem robot tersebut diperluas agar mampu mendeteksi dan mengenali wajah manusia. Berdasarkan OpenCV dan Python, metode Viola-Jones digunakan untuk mendeteksi wajah dan metode PCA untuk pengenalan wajah. Metode Viola-Jones menyediakan suatu cara untuk mendeteksi objek dalam citra secara cepat dan kokoh. Sedangkan PCA atau yang sering disebut sebagai eigenfaces adalah salah satu algoritma untuk mengenali wajah dengan mereduksi dimensi dari citra dengan menggunakan nilai eigennya. Secara umum pengenalan wajah dipengaruhi oleh berbagai macam variabel seperti intersitas cahaya, posisi wajah dan ekspresinya. Dengan mengendalikan variabel ini, pengenalan wajah dari robot humanoid yang dikembangkan mencapai sekitar sekitar 93% (28 pengenalan benar dari 30 eksperimen). Berdasarkan wajah yang telah dikenali ini, robot humanoid akan memberikan respon tindakan.**

Kata Kunci—***humanoid robot, Bioloid GP, Raspberry Pi 2, Viola-Jones, PCA, deteksi wajah, pengenalan wajah.***

I. Introduction

Recently, robot technologies are growing larger and become common in many area. Many current robotics researches focus in building a robot that can do activities and behave like human being. The typical examples of these researches are robot for delivering an order to customer and robot for demonstrating certain activities. In humanoid robot [1], vision systems are one of the main sources for environment information. The popular research in robotic vision system [2] is for detecting whether the target object is a human or not. For object detection, robot is expected to scan an image and detect target object which have certain specific features, such as human face. Another widespread research is for recognizing human face, just like a human does. For face recognition, the process begins by collecting face images which would like to be recognized and save them in a database. Those images are then resized and extracted their features by using PCA (Principle Component Analysis) [3]. PCA projects face images onto a feature space that spans the known face images in database. The significant features are called as "eigenfaces," because they are the eigenvectors (principal components) of the set of faces. Then individual face is characterizes by a weighted sum of the eigenface features. Finally, we compare these weights to those of known faces in database to recognize a specific face using Eulidean distance. Several methods can be used in the comparison process, for example neural network [4]. For our experiments, it is used 10 images of eigenfaces for each individual to capture the variations in an object. Generally, face recognition is done in the controlled variables, such as light intensity, poses, and expression. Uncontrolled variables would decrease the performance of face recognition [5]. Thus by controlling these variables, the result of face recognition accuracy is about 93% (28 correct recognition of 30 experiments).

The remainder of this paper is composed as follows: first we discuss robot architecture in section 2, and then is followed by the description of face detection and recognition system in section 3. In section 4, we report the experiment result in real time face recognition. Finally, we conclude our work with notes on future research in section 5.

II. Method

A. Architecture

The humanoid robot is designed to be able detect human face and recognize the face of people in front and then display his name. After that, the humanoid robot waves its hand and welcomes him with his name. The main components in robot architecture are Bioloid GP robot and Raspberry Pi 2 model B as controller. In

[1]Christian Tarunajaya, Oey Kevin Wijaya, Reinard Lazuardi Kuwandy, Heri Ngarianto, Alexander A S Gunawan, and Widodo Budiharto are with Departement of Computer Science, Universitas Bina Nusantara, Jakarta, Indonesia. Email: wbudiharto@binus.edu.

Figure 1, we descript our robot architecture in three modules that are: controller using Rasberry Pi 2, audio and visual sensors, and humanoid robot using Bioloid GP.

In robot architecture, camera is used to capture image in front of robot. Then to handle high speed image processing of the captured image, it is used Raspberry Pi 2, which based on recent ARM processor. The output of image processing in the main controller thus delivers to the CM 530 controller. Finally CM 530 will drive the actuators of the humanoid robot.

B. Raspberry Pi 2

Raspberry Pi 2 is used to process data from sensor and then control the robot based on the data. In this paper, we focus only on the visual sensor using low-cost USB webcam. Raspberry Pi 2 [1] which can be seen in Figure 2 has specification as follow:

a. Broadcom BCM2836 Arm7 Quad Core Processor powered Single Board Computer running at 900MHz
b. 1GB RAM
c. 40pin extended GPIO
d. 4 x USB 2 ports
e. 4 pole Stereo output and Composite video port
f. Full size HDMI
g. CSI camera port for connecting the Raspberry Pi camera
h. DSI display port for connecting the Raspberry Pi touch screen display
i. Micro SD port for loading your operating system and storing data
j. Micro USB power source

For our research, it is used the Linux-based Raspian as operating system of Raspberry Pi 2. For image processing, OpenCV 2.4.10 library [7] has to install over Raspian for Python 2.7 which is default programming language in Raspberry Pi.

C. Bioloid GP

The Bioloid GP [8] is a humanoid robot designed for robotics education. It contains high quality motors, lots of sensors, a controller and aluminum structural frames. The main benefit of Bioloid GP as programmable humanoid robot is easy to command though its CM 530 controller. Its gripper hands also allow it to manipulate objects. The robot also has several sensors, such as infrared distance sensor to avoid walls and gyroscopic sensor to maintain its balance. It is shipped with a wireless infrared remote control to respond your commands in certain distance range.

Programming is done through the RoboPlus software, included with the robot. It allows to record movements and to automate actions in a simple and intuitive manner. The program is then transferred on the CM-530 controller and the Bioloid GP programmable humanoid robot can act autonomously. In Figure 3, it can be seen Bioloid GP robot together with its components.

The side view appearance of our robot architecture including humanoid robot, embedded controller and visual sensor can be seen in Figure 4. The robot architecture has been designed in our previous research [9], which focuses on receiving command and doing speech recognition.

III. FACE DETECTION AND RECOGNITION SYSTEM

A. Image Preprocessing using Histogram Equalization

Histogram equalization [10] is a method to impove lightning in an image processing by adjusting its contrast using the image's histogram. This method increases the global contrast of an image, by distributing the intensities based on the histogram. Thus the areas with lower local contrast can gain a higher contrast. It can be done by spreading out the most frequent intensity values.

Histogram equalization is a basic technique from spatial domain of image processing. For implementing histogram equalization, it can be thought as palette change. Given L as maximum gray scale, histogram from digital image with gray scale span [0, L-1] is a discrete function:

$$h(r_k) = n_k \tag{1}$$

where r_k is k[th] gray scale value, and n_k is the number of pixel in image that have r_k's gray scale value [2]. In Figure 5 can be seen histogram of sample image. If the distance from 0 to L-1 divided into 2 parts with X_T as the threshold intensity, then this separation produces 2 histograms. The first histogram has scope from 0 to X_T, and the second histogram has scope from X_T to L-1.

Given an image with M x N pixels and L shows the maximum gray scale value, histogram equalization transformation T mapped input value r_k (where k=0,1,2,…,L-1) to output value S_k as follow:

$$S_k = T(r_k) = (L-1)\sum_{j=0}^{k} P_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j$$

$$k = 0, 1, 2, \ldots, L-1 \tag{2}$$

Probability $P_r(r_j)$ is the total pixel with intensity value r_j in input image n_j divided by total pixels in the image M x N. With the histogram equalization transformation, the lightning in the image can be corrected effectively.

B. Face Detection

Face detection and recognition system can be divided into four steps, that is:

1. Face Detection
2. Face Alignment
3. Feature Extraction
4. Feature Matching

Face detection is the first process which starts by scanning face in each image frame. The chosen detection method is Viola-Jones [11] which provides a complete framework for extracting and recognizing image features. A combination of several algorithms in Viola-Jones method including Haar-like features, integral image, boosting algorithm, and cascade classifier provides a robust and fast detector for the human face detection.

C. Face Recognition

After robot detects human face in first step, then we have to de face alignment. The second step, face alignment, is procedure for identifying the geometric structure of human faces in an image by doing localization and normalization. It can be done by using face location and scale estimation value from face detection step.

Face recognition is the process where computer analyze the detected face image to identify it by comparing to the known face image in database. Face recognition process can be divided into two steps: feature extraction and feature matching. In feature extraction, we extract unique features from the detected face image. In here, it is used PCA (Principle Component Analysis) for this task. While feature matching is proceed by matching the extracted features of detected face image to a set of face image data in database. We use Eulidean distance method for feature matching step.

According to Shakhnarovich and Moghaddam [12], there are some difficulties in face recognition that is:

a. To handle high dimension, especially recognition context based on similarity and matching is expensive based on computation.
b. For parametric method, total of parametric that needed grow in exponential based on dimensionality.
c. For non-parametric method, complexity of sample is high enough.

Therefore, dimension reduction technique is needed to build face recognition system. One of the fast and reliable algorithms for dimension reduction is PCA (Principal Component Analysis). Thus PCA have two roles in face recognition that is: for extracting image feature and also in reducing image dimension.

D. PCA

Principal Component Analysis is a standard technique that is used in statistical pattern recognition and signal processing for data reduction and extraction features [4]. Every principal component is a representation for linear combination for every face images in database that have been substracted by image mean. In PCA, the enquiried and known images must be the same size. Therefore, a normalization is needed to lineup the eyes and the mouths across all images. Each image is treated as one vector.

All images of the training set are stored in a single matrix T and each row in the matrix represents an image. The average image has to be calculated and then subtracted from each original image in T. Then calculate the eigenvectors and eigenvalues of the covariance matrix S. These eigenvectors are called eigenfaces. The eigenfaces is the result of the reduction in dimensions which removes the useless information and decomposes the face structure into the uncorrelated components (eigenfaces).

The training database consists of M images which is same size. The images are normalized by converting each image matrix to equivalent image vector ri. The training set matrix r is the set of image vectors, that is:

Training set r=[r_1,r_2,…r_M]

The mean face (ψ) is the arithmetic average vector as given by:

$$\psi = \frac{1}{M}\Sigma_{i=1}^{M}\Gamma_i \quad (3)$$

The deviation vector for each image Φi is given by:

$$\Phi_i = \Gamma_i - \psi \quad (4)$$

$$i = 1, 2, \ldots M$$

Consider a difference matrix A= [Φ_1, Φ_2, ……. Φ_M] which keeps only the distinguishing features for face images and removes the common features. Then eigenfaces are calculated by find the Covariance matrix C of the training image vectors by:

$$C = A.A^T \quad (5)$$

Due to large dimension of matrix C, we consider matrix L of size (M_t x M_t) which gives the same effect with reduced dimension. The eigenvectors of C (Matrix U) can be obtained by using the eigenvectors of L (Matrix V) as given by:

$$U_i = AV_i \quad (6)$$

The eigenfaces are:

eigenface $= [U_1, U_2, U_3, \ldots U_M]$

Instead of using M eigenfaces, the certain value $m' <= M$ is chosen as the eigenspace. Then the weight of each eigenvector ωi to represent the image in the eigenface space, as given by:

$$\omega i = U_i^T(\Gamma - \Psi), i = 1, 2, \ldots m' \quad (7)$$

Weight matrix $\Omega = [\omega_1, \omega_2, \ldots \omega_{m'}]^T$

Average class projection $\Omega_\Psi = \frac{1}{X_i}\Sigma_{i=1}^{X_i}\Omega_i$ (8)

Finally the euclidean distance δi (8) is used to find out the distance between two face keys vectors and is given by:

$$\delta_i = \left|\left|\Omega - \Omega_{\Psi i}\right|\right| = \Sigma_{k=1}^{M}(\Omega_k - \Omega_{\Psi iK}) \quad (9)$$

Euclidian distance is one of the methods that can be used to match new face image to face image in database. Smaller value means the image is more similar to the one in database.

IV. Results and Discussion

The result of our research is implemented on Bioloid GP robot by using OpenCV 2.4.10. There are two main processes in our research, one is offline collecting known face images in database, and another is online face recognition process. Firstly, we need to capture known faces to be stored in database, for example in Figure 6.

For our experiment, it is used 10 known faces images for each person and then stored in database. Next step is image preprocessing by converting them to grayscale images and doing histogram equalization. The histogram equalization will make the images more contrast. In collecting process, we select only the appropriate images and eliminate those that can’t be used. The appropriate known faces are face images which have significant different positions or expressions. By selecting the images appropriately, we can increase the accuracy of the face recognition.

After 1st step is done, the known face images are stored in a database folder that had been created before. The process of selecting and saving the images are done offline. It means the processes are done while the robot is turned off. Furthermore, we start the online face recognition step. We can divide the developed face recognition system into different stages, which assure the robot can detect and recognize any faces smoothly.

First, when the robot is turned on, the camera will stream the image frames to controller, which then search any faces in them. When a face is detected, robot will mark it as "Unknown" because the process identifying has not started yet. To do face recognition process, we need to push the button at its back, and the robot will run the code of face recognition. The first stage in the face recognition algorithm is to capture an RGB image, as follow Figure 7.

Then, converting the RGB image into grayscale image, to reduce computation in the image processing. The grayscale image is illustrated in Figure 8.

The next step, the lightning in grayscale image will be corrected using histogram equalization method. The resulted image will be more contrast and clearer, which can be seen in Figure 9.

Next, the face in the image is cropped (see Figure 10), because only the important part of the image will be used in the recognition process. The cropping process is done by calculating the area's pixel value in form [x_1, y_1, x_2, y_2] array which is obtained during detection process.

The cropped image is then stored in the certain folder. By accessing that image in that folder and the known faces in database, we can compare the similarity between them. The result of this matching process is a value which called as Euclidean distance. The smaller the value, the closer the enquiried face to certain known face in database. By chosen a threshold value, we can infer that two images have similar features. When the detected face can be recognized well, robot then respond by saying hello and calling the recognized person's name.

To verify the developed system, we did 30 experiments based on three known person images. The next Table 1-3 show the experiment results.

In this research, there is still a false recognition due to by causes, such as the quality known images which too dark, less varied, or the blur enquiried image. The accuracy of the experiment results is about 93% (28 correct recognitions out of 30). Eleyan et al [13] which using PCA and Neural Netwok as classifiers reach the accuracy about 95%. Comparing to this result, our developed system which uses simple Eulidean distance as classifier has comparable performance.

In this research, there is still a false recognition due to by causes, such as the quality known images which too dark, less varied, or the blur enquiried image. The accuracy of the experiment results is about 93% (28 correct recognitions out of 30). Eleyan et al **Invalid source specified.** which using PCA and Neural Netwok as classifiers reach the accuracy about 95%. Comparing to this result, our developed system which uses simple Eulidean distance as classifier has comparable performance.

CONCLUSION

In this paper, we would like to explain how to recognize face using PCA algorithm by using Raspberry Pi 2 based on OpenCV 2.4.10 and Python 2.7. RaspberryPi as controller is then integrated to humanoid robot Bioloid GP from Robotis in order to implement face recognition in humanoid robot system.

In the experiment result, our approach which uses PCA algorithm and Euclidean distance as classifier reaches the accuration about 93%. For future work, it is planned to build humanoid robot which can detect and recognize multiple faces and make respond to everyone persons.

ACKNOWLEDGMENT

We thank to Indonesian government which supports this research by granting Hibah Ristek 2015.

REFERENCES

[1] I. Ha, Y. Tamura and H. Asama, "Development of Open Humanoid Platform DARwIn-OP," in *SICE Annual Conference (SICE)*, Tokyo, 2011.

[2] R. Siegwart, I. R. Nourbakhsh and D. Scaramuzza, Introduction to Autonomous Mobile Robots, 3 ed., Cambridge: The MIT Press, 2011, p. 976.

[3] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience,* vol. 3, no. 1, pp. 71-86, 1991.

[4] A. Eleyan and H. Demirel, "PCA and LDA Based Face Recognition using Feedforward Neural Network Classifier," in *International Workshop Multimedia Content Representation, Classification and Security (MRCS)*, Istanbul, 2006.

[5] C. Conde, I. M. d. Diego and E. Cabello, "Face Recognition in Uncontrolled Environments, Experiments in an Airport," in *E-Business and Telecommunications*, Springer Berlin Heidelberg, 2012, pp. 20-32.

[6] RASPBERRY PI FOUNDATION, "RASPBERRY PI 2 MODEL B," Credit Card–Sized Single Board Circuit, 2015. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-2-model-b/. [Accessed January 2016].

[7] OpenCV, "OpenCV 2.4.10," Open Source Computer Vision, 02 10 2014. [Online]. Available: http://opencv.org/documentation/opencv-2-4-10.html. [Accessed January 2016].

[8] Robotis, "Robotis Inc," Robot Production, [Online]. Available: http://en.robotis.com/index/product.php?cate_code=121510. [Accessed January 2016].

[9] W. Budiharto, A. A. S. Gunawan, A. C. Sari and H. Ngarianto, "Designing of Humanoid Robot with Voice Recognition Capability," in *The 3rd IIAE International Conference on Intelligent Systems and Image Processing 2015 (ICISIP2015)*, Fukuoka, 2015.

[10] R. C. Gonzalez and R. E. Woods, Digital Image Processing (3rd Edition), New Jersey: Prentice Hall, 2007.

[11] "B. Senjaya, A. A. S. Gunawan and J. P. Hakim, "Pendeteksian Bagian Tubuh Manusia untuk Filter Pornografi dengan Metode Viola-Jones," Jurnal ComTech, vol. 3, no. 1, pp. 482-489, 2012".

[12] G. Shakhnarovich and B. Moghaddam, "Face Recognition in Subspaces," in *Handbook of Face Recognition*, London, Springer-Verlag London, 2004, pp. 141-168.

[13] A. Eleyan and H. Demirel, "PCA and LDA based Neural Networks for Human Face Recognition," in *Face Recognition*, Vienna, I-Tech Education and Publishing, 2007, pp. 93-106.

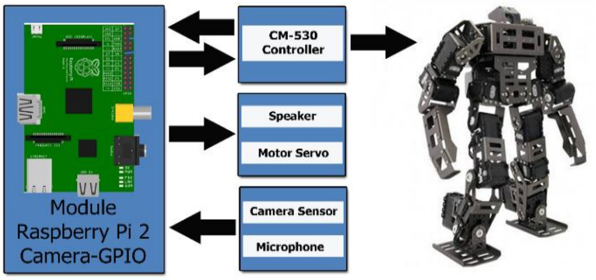

Figure 1. Architecture of our humanoid robot.

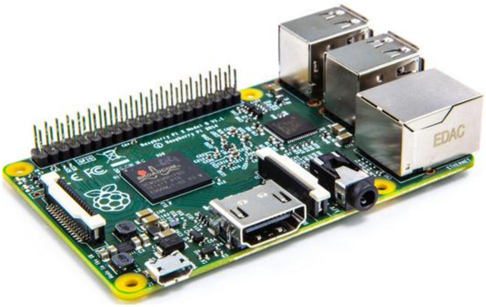

Figure 2. Raspberry Pi 2 Model **B**

Figure 3. Bioloid GP robot

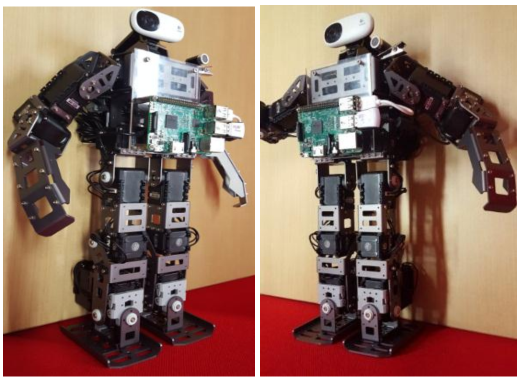

Figure 4. Side view appearance

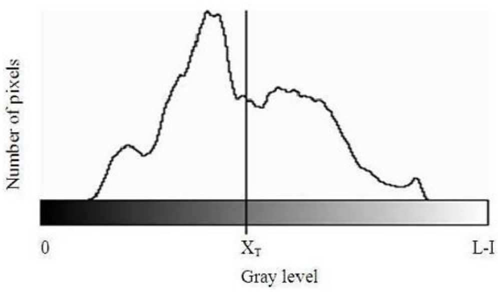

Figure 5. Histogram of sample image

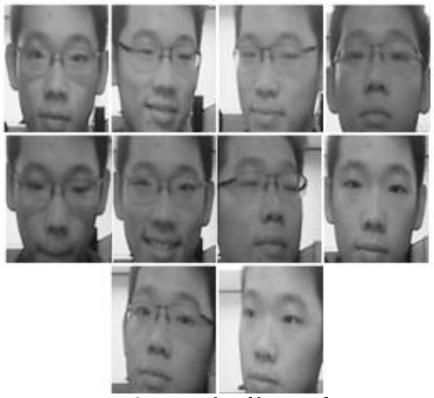

Figure 6. Example of known faces

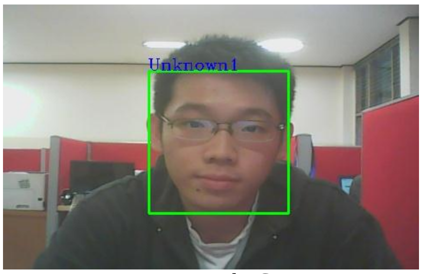

Figure 7. Detected RGB image

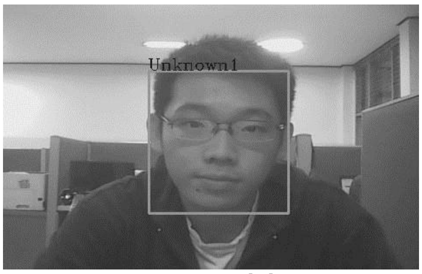

Figure 8. Gray scaled image

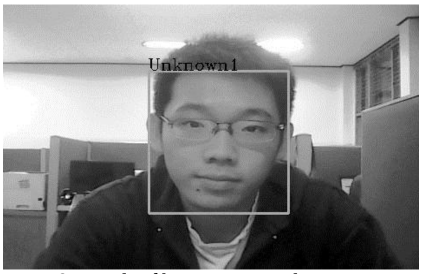

Figure 9. Result of histogram equalization process

Figure 10. Cropped image

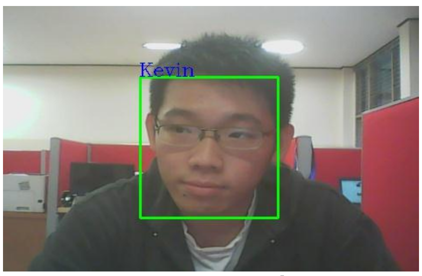

Figure 11. Recognized Face

TABLE 1.
THE EXPERIMENT RESULTS KEVIN'S TEST FACES

Kevin's Test Faces		
Test Face	Result	Distance
	Kevin (true)	2.1166
	Kevin (true)	2.1209
	Kevin (true)	1.7125
	Kevin (true)	1.6568
	Kevin (true)	2.0869
	Kevin (true)	2.4114
	Kevin (true)	1.6606
	Kevin (true)	2.2442
	Reinard (false)	2.1411
	Christian (false)	1.8994

TABLE 2.
THE EXPERIMENT RESULTS CHRISTIAN'S TEST FACES

Christian's Test Faces		
Test Face	Result	Distance
	Christian (true)	2.0498
	Christian (true)	1.2895
	Christian (true)	1.6999
	Christian (true)	1.5994
	Christian (true)	1.7049
	Christian (true)	2.0217
	Christian (true)	2.2439
	Christian (true)	1.4606
	Christian (true)	1.7815
	Christian (true)	2.1652

TABLE 3.
THE EXPERIMENT RESULTS REINARD'S TEST FACES

Reinard's Test Faces		
Test Face	Result	Distance
	Reinard (true)	1.0657
	Reinard (true)	2.2486
	Reinard (true)	1.6651
	Reinard (true)	1.9914
	Reinard (true)	1.7181
	Reinard (true)	1.7201
	Reinard (true)	1.9038
	Reinard (true)	1.5883
	Reinard (true)	1.5212
	Reinard (true)	2.1116