Cognitive-based Online Face Matching System

Teik-Toe Teoh[1], Kiat-Han Kok[2], Oruganti Venkata Ramana Murthy[2], Siu-Yeung Cho[3], Yok-Yen Nguwi[1]
[1]School of Business (IT), James Cook University, Singapore
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]Division of Engineering, The University of Nottingham Ningbo, China
e-mail: Nguwi.yokyen@jcu.edu.sg

***Abstract*—Social network service provides means for users to interact over the Internet. Early social networking on Internet began in the form of generalized online communities such as Geocities (1994) and Tripod.com (1995). Social network makes it possible to connect people who share interests and activities across geographical borders. The most successful social network today is facebook.com, which ties together 750 million active users around the world. The recent Google+ attempts to dethrone Facebook provide similar functionalities and search capability. The current searching capability in social network is limited to search via name and email. The significance of this work is that we propose a new search method that opens new frontiers to existing search functions. We propose a unique FaceSearch system that allows user to search people from faces appear in a photo. Once similar faces are found or suggested from database, contact details of the person can be retrieved. FaceSearch uses a face recognition engine that reads Gabor features. Gabor kernels have its root on high similarity to human visual V1 processing. The Euclidean distance is used to show closest match. We developed a GUI that runs on Androids. The simple and yet effective design warrants reasonable processing speed and good recognition rate of about 80%.**

Keywords- facebook, face match, face search, gabor

I. INTRODUCTION

In the year of 2010, Facebook.com overtook Google.com as the most popular website [1]. The aims of social network are for interpersonal communication, to meet new people and find old friends. Some researchers like Lampe, Ellison and Steinfield [2] supposed that Facebook is largely used to upkeep or reinforce on-going offline relationships, rather than creating new friendships online. The other researcher Joinson [3] concluded that social network is primarily used for "social searching", i.e. to get to know more about people they have met briefly, whether through mutual friends or through shared experiences. Facebook searches relying largely on mutual friends, names, email address. This static text based search function only works if the names are remembered. There are situations where acquaintances were met briefly, taken pictures together but can only vaguely remember this person and hard to recall the name.

This work developed an android phone application which uses a Face Recognition Algorithm to find faces of high correlation with the key picture. We investigated how the different face recognition algorithms affect the computation time and accuracy to search similar/familiar faces from his/her existing Facebook's friends list using photos. This application enables fast and live recognition of faces has the potential of enhancing the overall socialization experience.

II. LITERATURE REVIEW

FaceSearch is based on three main strategies to identify persons in a photo: the use of Euclidean distance, the use of cropping images and an adaptation of image representation using 2D Gabor Wavelets. Euclidean distance is used as the basic measurement unit measuring the length of line segment connecting two points.

In a similar project on facial recognition by Guillaume Dave, Xing Chao and Kishore Sriadibhatla [4], facial recognition serves as a security function that will help enhancing the accessibility of the android phone for its user by minimizing time spent on negotiating through security functions which utilizes user's personal photograph. They adopted methods such as color segmentation and template matching. The significance of their research is the conclusion on how varying light intensities and background images tend to limit the accuracy of the recognition process. In our proposed FaceSearch, we filter background noise to lower and eliminate the interference of these variables and also help to optimize performance by minimizing the total Euclidean distance that needs to be covered per photograph. Moreover, the primary difference lies in the function and purpose of the facial recognition process, which is targeted at overall social level rather than a personal security level.

Research on the use of Gabor Wavelets for the purpose of image representation in 2D tend to focus on improvements to the efficiency of the Gabor Wavelets or to propose feasible alternatives such as the work by Gareth Loy [7]. Gareth approximates the wavelet cross-section with a function that can be further separated into a convolution involving a sparse component. He proposed to use an adapted version of Gabor Wavelet, which is called "mGabor" or "mini-Gabor", to reduce the number of Gabor Wavelets to 1000 features per cropped photograph. The rationale of such is to optimize speed whilst minimizing compromise on accuracy. The details of FaceSearch structure is introduced in the following section.

III. COMPUTATIONAL FRAMEWORK

There are three stages involved in FaceSearch: Preliminary phase, Secondary phase, and Final phase. Preliminary phase prepares for image collection. Secondary phase queries pictures from database based on the same set of features. Final phase presents the results of similarity ranking. The details are set out as follows:

A. *Preliminary Phase*

Preliminary phase prepares for image collection. In this phase, users store the relevant images in his smart phone. The images are then passed through the feature extraction system to generate feature vector.

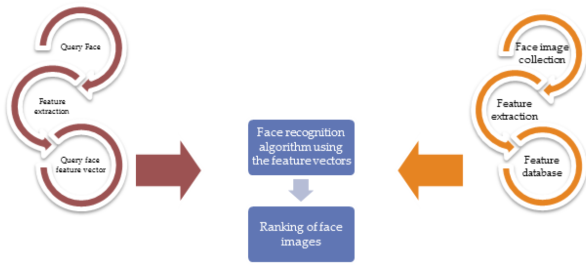


Figure 1 Technology Overview

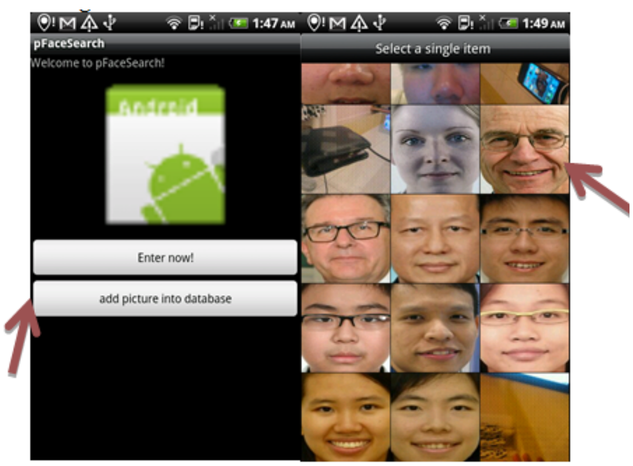


Figure 2: Image collection selection screen using manual method

Feature vector is a variable used to store essential values that are used to calculate statistical image information for face [8]. It contains a range of normalized integer values.

We use two approaches for image processing: manual method or Facebook method. The manual method is illustrated in Figure 2; user manually clicks on ‘add picture into database’ button to add pictures from the photo library into database. The Facebook method is illustrated in Figure 3, user clicks on the button to enable collection of the Facebook friends’ profile photo. The Graph API developed by Facebook [9] provides that authentication of Facebook user will be done first, followed by the authorization of Facebook application. Lastly, the list of photos added into the database will be shown to user.

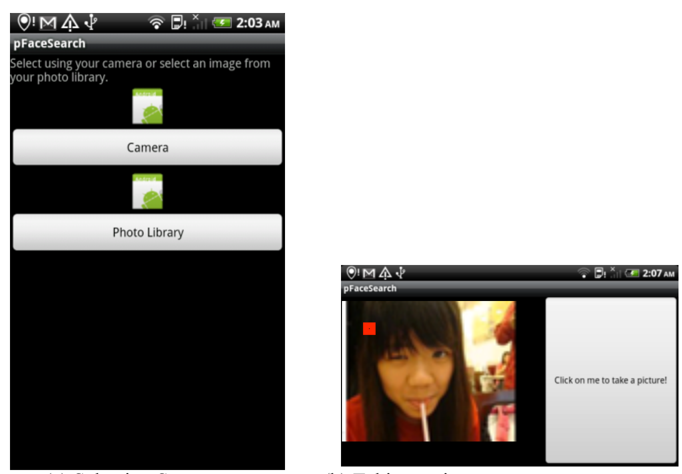
The process of adding photos onto the database activates feature extractions. Upon successful storing of essential feature values, subsequent phases can be started.

Figure 3: Image collection process using Facebook method

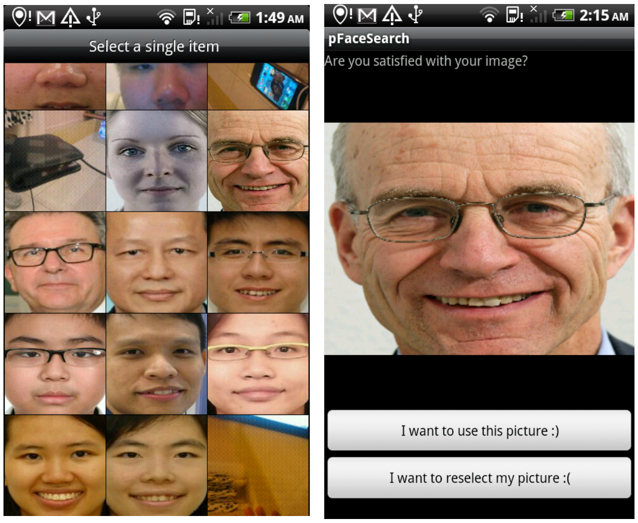
B. *Secondary Phase*

Secondary phase queries pictures from database based on the same set of features. User first specifies the image for querying. The query image will then undergo the same feature extraction process as set out in Section A to retrieve its unique set of feature vector.

There are two ways in the process of querying image collection. The process can start by selecting an image from the library or by taking a picture as shown in Figure 4. After selection of the picture, the face from the image is detected using the FaceDetector Class provided in Android library [10]. The face is then extracted for the recognition process. A confirmation screen will then be prompted to confirm the selection of the user’s photo.



(a) Selection Screen (b) Taking a picture



(c) Choosing from library (d) Confirmation Screen

Figure 4: Image Querying

C. Final Phase

The final phase consolidates efforts from prior phase and utilizes a face recognition algorithm to find similar faces. The face recognition algorithm analyzes feature vector that's stored in the database to find the closest match. The closest match is the one with the smallest difference in the feature vector's value. In cases where the difference exceeds a certain limit; we use threshold to control the readable values. The image is deemed as an invalid match if the difference is large. Figure 5 shows a sample view of the ranking of results from our prototype.

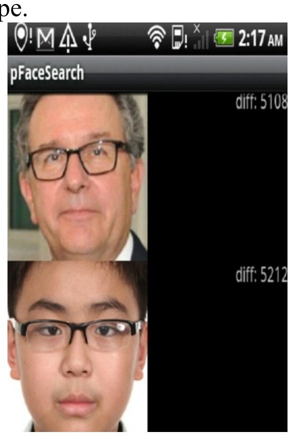


Figure 5: Ranking Results from Prototype

IV. FACE RECOGNITION ALGORITHMS

This section discusses the phases of face recognition algorithm used. An analysis study is also carried out to compare its performances based on accuracy and computation time taken.

A. Image Cropping

In order to enhance the performance of the system, some areas which do not contain meaningful information is deemed as noise and is removed from the original image. These areas include the background of the image, the upper face areas where colors of hairs and wearing of spectacles can lead to poor results. The feature vector of an image is computed using Euclidean distance [11].

Figure 6 illustrates the areas which are cropped. The rationales for cropping the images are: Firstly, we remove background noise and edges of the faces that may affect the results through cutting left/right/bottom side of the image. Secondly, we remove top side of the image because the subject may be wearing glasses, and the hair style may change, this cropping reduces approximately 50% of the image area.

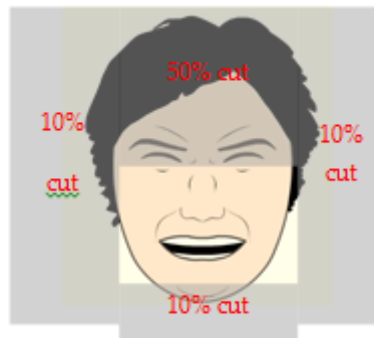


Figure 6: Image Cropping

B. Computation using Euclidean distance

We use a simple yet compute intensive algorithm that calculates the difference between each feature vectors as proposed by Calvin R. Maurer [12]. This method is defined as the baseline of all algorithms in terms of calculation speed and accuracy.

Ranking is based on computation of Euclidean distance between the feature vectors:

Length of byte array is calculated to be 153600 for a 320x240 image.

The mathematical formula [13] is as follows:

$$\text{Feature Vector }(x)=\{x_1,x_2,x_3,x_4,\ldots,x_{\text{length of byte array}}\}$$

$$\text{Feature Vector }(y)=\{y_1,y_2,y_3,y_4,\ldots,y_{\text{length of byte array}}\}$$

$$\text{Euclidean dist.}=\{(x_1-y_1)^2+(x_2-y_2)^2+\ldots + (x_{\text{length of byte array}}-y_{\text{length of byte array}})^2\}^{\frac{1}{2}}$$

C. "Mini-Gabor"

This step implements "Mini-Gabor" to get the convolved feature vector, followed by constructing the comparisons using Euclidean distance to get the difference in each image. The use of "Mini-Gabor" has greatly improved the computation speed and accuracy.

Gabor kernels are similar to the receptive field profiles found in cortical simple cells, which are characterized as localized, orientation selective, and frequency selective [14]. It is a linear filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination [15]. In spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave [16]. The Gabor filters are self-similar; all other filters can be generated from one mother wavelet by dilation and rotation [17]. According to Jones and Palmer [18], these receptive fields can be reproduced fairly well using Daugman's Gabor function.

J. G. Daugman [19] discovered that simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions. Thus, image analysis by the Gabor functions is similar to perception in the human visual system.

K. Etemad [5] also found out that different components of wavelet decomposition capture different visual aspects of a gray scale image. Therefore, we need to devise the correct components of wavelet decomposition to produce a strong result.

We used 5 scales and 8 orientations; a total of 40 Gabor functions are used as the default Gabor. The number of oscillations under the Gaussian envelope function is determined by $\delta = 2\pi$. Gabor convolution will retrieve the 3D orientation information [6]. The response of Gabor kernel is shown in Figure 7.

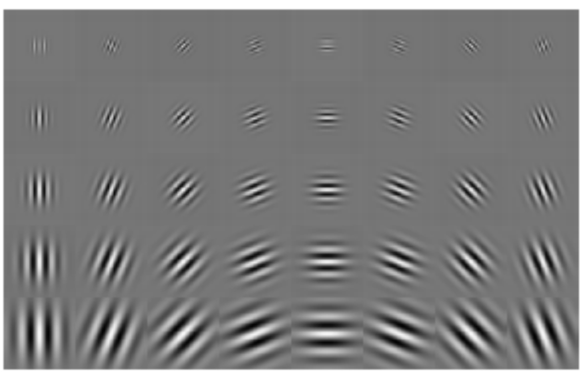


Figure 7: Response of Gabor Kernel using Different Orientations and Phases

In order to develop the application on a mobile platform, a simplified version of Gabor filter, named "Mini-Gabor" is used. It has 5 scales and 4 orientations, a total of 20 Gabor functions to reduce the computation complexity.

After taking the cropped image from the above step, "Mini-Gabor" is convolved with the image. A feature vector of maximum of 1000 values from the sum of 20 Gabor functions is computed and stored into the database.

D. Face Recognition Algorithm Analysis

To further prove that this algorithm works, we managed to gather a set of 50 sets samples from Facebook. This set of images is tested with the 3 algorithms, building upon one and another.

We analyze the results base on computation time and accuracy of results. However, better accuracy is always compensated with longer computation time and vice versa. Therefore, we attempted to balance the computation time and accuracy results. Table I shows the computation time in calculating ranking, adding an images and the number of values in feature vector.

Table II shows a significant improvement in using "Mini-Gabor" and is ranked the best in terms of computation time. The rationale for such can be traced back to the number of values in feature vector. The difference in feature vector affects the results greatly because the amount of loops that the processor needs to goes through affect the computation time.

We used different threshold values to analyze the best value to place in order to achieve the highest amount of positive results (i.e. the case where correct person is identified with the correct query).

Figure 8 shows the overall performance results in terms of computational time and accuracy. Blue color denotes results with using pure Euclidean Distance, red color shows results with area cropping is added to the step. Orange color shows the enhancement of Mini Gabor to the process. The computational time has decreased tremendously but accuracy has helped face recognition to be a lot more accurate. This greatly increases the potential for commercialization to the market.

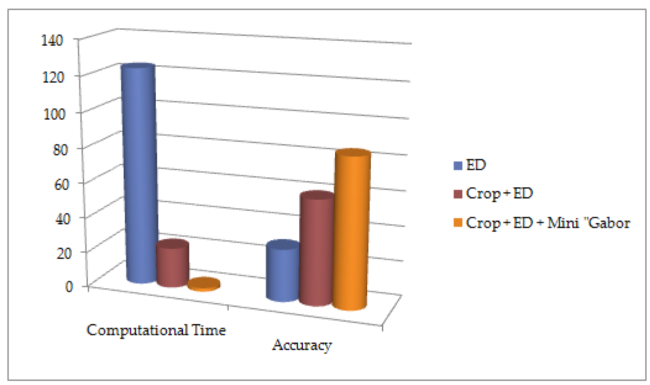


Figure 8: Performance Results

TABLE I. COMPUTATION TIME TABLE

Image No.	Computational Time (Secs)			Addition of images		
	Euclidean distance	*Image cropping, Euclidean Distance (ED)*	*Image cropping ED and "Mini-Gabor"*	*ED*	*Cropped Image, ED*	*Cropped Image, ED and "Mini-Gabor"*
1	3.323	0.161	0.0404	0.213	0.380	0.083
2	5.197	0.244	0.0611	0.324	0.565	0.127
3	7.436	0.339	0.0847	0.453	0.772	0.177
4	9.395	0.415	0.1037	0.559	0.933	0.218
5	10.235	0.439	0.1097	0.595	0.974	0.232
6	11.923	0.496	0.1241	0.678	1.089	0.265
7	13.105	0.531	0.1326	0.729	1.151	0.285
8	15.823	0.623	0.1558	0.861	1.339	0.336
9	17.234	0.661	0.1652	0.919	1.406	0.359
10	19.634	0.734	0.1834	1.025	1.547	0.400
11	20.122	0.733	0.1833	1.030	1.532	0.402
12	22.573	0.802	0.2006	1.132	1.663	0.442
13	22.765	0.790	0.1975	1.120	1.625	0.438
14	25.742	0.872	0.2181	1.243	1.782	0.485
15	27.983	0.927	0.2317	1.326	1.880	0.518
16	27.486	0.890	0.2225	1.279	1.794	0.499
17	29.573	0.937	0.2342	1.351	1.876	0.528
18	33.624	1.043	0.2606	1.509	2.076	0.590
19	32.963	1.001	0.2502	1.454	1.981	0.568
20	36.012	1.071	0.2678	1.562	2.109	0.610
21	38.035	1.109	0.2771	1.622	2.172	0.634
22	40.361	1.153	0.2883	1.693	2.249	0.661
23	39.326	1.102	0.2755	2.515	2.246	0.983
24	41.054	1.129	0.2822	2.562	2.287	1.001
25	45.923	1.239	0.3098	2.797	2.498	1.093
26	48.102	1.275	0.3186	2.862	2.555	1.118
27	53.126	1.383	0.3456	3.089	2.758	1.207
28	55.910	1.429	0.3574	3.179	2.838	1.242
29	58.124	1.460	0.3651	3.233	2.886	1.263
30	61.940	1.530	0.3825	3.372	3.010	1.317
31	63.427	1.541	0.3851	3.381	3.018	1.321
32	65.567	1.566	0.3916	3.423	3.328	1.337
33	67.372	1.584	0.3959	3.447	3.346	1.347
34	69.936	1.618	0.4044	3.508	3.399	1.370
35	71.326	1.624	0.4060	3.509	3.394	1.371
36	73.461	1.647	0.4118	3.546	3.425	1.385
37	75.243	1.661	0.4154	3.565	3.438	1.393
38	77.043	1.676	0.4189	3.285	3.451	1.283
39	81.305	1.742	0.4356	3.410	3.572	1.332
40	83.572	1.765	0.4413	3.449	3.602	1.347
41	85.236	1.775	0.4436	3.461	3.606	1.352
42	87.937	1.805	0.4512	3.515	3.652	1.373
43	90.329	1.828	0.4571	3.555	3.685	1.389
44	93.105	1.859	0.4647	3.609	3.731	1.410
45	95.387	1.879	0.4696	3.642	3.757	1.423
46	98.236	1.909	0.4772	3.696	3.803	1.444
47	101.847	1.953	0.4882	3.776	3.877	1.475
48	105.121	1.990	0.4974	3.842	3.936	1.501
49	110.332	2.011	0.5153	3.976	4.065	1.553
50	115.128	2.124	0.5309	4.091	4.174	1.598

TABLE II. ACCURACY TABLE

Threshold Value	No. of positive results (50)		
	Euclidean distance (ED)	*Image cropping, ED*	*Image cropping ED and "Mini-Gabor"*
1000	19	27	38
1100	25	27	45
1200	25	29	40
1300	20	23	42
1400	15	36	37
1500	25	23	44
1600	16	31	45
1700	23	24	42
1800	15	24	41
1900	22	28	41
2000	23	35	40
2100	21	24	44
2200	18	32	44
2300	20	29	37
2400	15	27	39
2500	22	29	44
2600	23	29	44
2700	24	34	45
2800	24	32	39
2900	16	27	35
3000	25	22	35
3100	22	34	41
3200	17	22	44
3300	15	34	36
3400	23	26	44
3500	19	27	38

V. CONCLUSION AND FUTURE WORKS

Continuing from the current progress, the integration of this prototype with Facebook can be followed up through the deployment on different smart phones. The profile images of the Facebook are downloadable onto the phone. Follow-up can be done to store these images into the database, and provides functionality to select the best image from the Facebook's friends' tagged photos.

On the other hand, we can also utilize this prototype for community event purposes. The other further works can be extended to use this prototype to be applied on criminal forensics. The relevant authorities and citizens can co-operate together facilitated by the use of technology to identify criminals easily and instantaneously using the smart phones, enhancing social security.

Alternatively, the community or other research groups may also choose to modify the prototype for recognition of non-facial features such as landscaping for physical geography purposes, or possibly for research in the life sciences pertaining to flora and fauna.

VI. TOOLS

Android SDK Revision 11, Android 2.2 Platform
HTC Desire HD Android Version 2.2
Eclipse Classic 3.6.2

Java™ SE Runtime Environment (build 1.7.0-ea-b82)

REFERENCES

[1] The Next Web, http://thenextweb.com/insider/2010/12/30/facebook-beat-google-as-the-most-visited-site-in-2010/

[2] Lampe, C., Ellison, N., & Steinfield, C. (2007). A familiar Face(book): Profile elements as signals in an online social network. Proceedings of Conference on Human Factors in Computing Systems. New York: ACM Press.

[3] Joinson, A. (2008). "Looking at", "Looking up" or "Keeping up with" people? Motives and uses of Facebook. CHI 2008 Proceedings-Online Social Networks. Florence, Italy.Elena Deza & Michel Marie Deza (2009) Encyclopedia of Distances, page 94, Springer.

[4] Guillaume Dave, Xing Chao, Kishore Sriadibhatla; Face Recognition in Mobile Phones. Stanford University.

[5] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images", J. Opt. Soc. Amer. A, vol. 14, pp. 1724 - 1733 , 1997.

[6] Joinson, A. (2008). "Looking at", "Looking up" or "Keeping up with" people? Motives and uses of Facebook. CHI 2008 Proceedings-Online Social Networks. Florence, Italy.Elena Deza & Michel Marie Deza (2009) Encyclopedia of Distances, page 94, Springer.

[7] Gareth Loy; Fast Computation of the Gabor Wavelet Transform. DICTA2002: Digital Image Computing Techniques and Applications, 21–22 January 2002, Melbourne, Australia

[8] Xiaoguang Lu; Image Analysis For Face Recognition. Michigan State University

[9] Facebook Graph API, http://developers.facebook.com/docs/reference/api/

[10] Android Developers, FaceDetector Class, http://developer.android.com/reference/android/media/FaceDetector.htm

[11] W.K. Teo, Liyanage C De Silva and Prahlad Vadakkepat; Facial Expression Detection and Recognition System. Journal of The Institution of Engineers, Singapore, Vol. 44 Issue 3 2004

[12] Calvin R. Maurer, Jr., Member, IEEE, Rensheng Qi,and Vijay Raghavan, Senior Member, IEEE; A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 2, Feburary 2003

[13] Per-Erik Danielsson; Euclidean distance mapping, 6 February 1980

[14] Linlin Shen, Li Bai. A review on Gabor wavelets for face recognition. Received: 6 December 2004 / Accepted: 17 May 2006 / Published online: 18 August 2006_ Springer-Verlag London Limited 2006

[15] Yi-Chun Lee, Chin-Hsing Chen, "Feature Extraction for Face Recognition Based on Gabor Filters and Two-Dimensional Locality Preserving Projections", Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference on, On page(s): 106 - 109, Volume: Issue: , 12-14 Sept. 2009

[16] Li-Men Cui, Yuan-Yan Tang, Fu-Cheng Liao, Xiu-Feng Du, "Face recognition based on wavelet packet decomposition and support vector machines", Wavelet Analysis and Pattern Recognition, 2007. ICWAPR '07. International Conference on, On page(s): 1437 - 1441, Volume: 3 Issue: , 2-4 Nov. 2007

[17] Yang, Xiaochao, Dai, Ji, Zhou, Yue, Yang, Jie, "Gabor-Based Discriminative Common Vectors for Gait Recognition", Image and Signal Processing, 2008. CISP '08. Congress on, On page(s): 191 - 195, Volume: 4 Issue: , 27-30 May 2008

[18] J.P. Jones, L.A. Palmer, "An evaluation of the Two-Dimensional Gabor Filter model of simple Receptive fields in cat striate cortex", *J.Neurophysiol.,* vol. 58 (6), pp. 1233-1258, 1987.

[19] J. G. Daugman, "Two-dimensional spectral analysis of cortical receptive field profiles", Vis. Res., vol. 20, pp.847 - 856 , 1980.