All Common Subsequences for Face Recognition

http://dx.doi.org/10.3991/ijoe.v9i4.2818

C.H. Wang[1], A.K. Hu[1] and F.L. Han[1]
[1] College of Shipbuilding Engineering, Harbin, China

***Abstract*—In recent years, face recognition has become one of the hottest research topics aimed at biometric applications. Comparing with other biometrics recognition, face recognition provides more natural means for perceptual interface. However, face recognition algorithms weakly perform under some common conditions, which include the variation of facial expressions or lightening conditions, the occlusion of faces like wearing glasses or mask, the low resolution or noises of input images, and the like. The other problem is the recognition efficiency, especially when the facial database is tremendous. This paper presents all common subsequences (ACS) as the kernel function (similarity method) to solve the time series problem. Experiments on 4 public face databases: Caltech, Jaffe, Orl and Yale databases, demonstrate that ACS can achieve higher recognition accuracy than some classic face recognition methods, e.g. 2DPCA and 2DLDA.These instructions give you basic guidelines for preparing camera-ready papers for conference proceedings.**

***Index Terms*—All common subsequences (ACS), Face recognition, kNN, SVM, 2DPCA, 2DLDA**

I. INTRODUCTION

Face recognition is one of the hottest research topics aimed at biometric applications such as robotics, visual surveillance, human-computer interfaces etc. Comparing with other biometrics recognition, e.g. fingerprint, eye iris recognition, face recognition provides more natural means for perceptual interface without special requirements for user actions while only makes use of a wide range of inexpensive consumer cameras. However, the face recognition technic for consumer applications still remains a difficult problem. The main problem is that most of face recognition algorithms weakly perform under the some common conditions, which include the variation of facial expressions or lightening conditions, the occlusion of faces like wearing glasses or mask, the low resolution or noises of input images, and the like. The other problem is the recognition efficiency, especially when the facial database is tremendous. A good recognition algorithm should react in real time. Thereby, it is a great challenge for a face recognition algorithm to achieve high robustness and computational efficiency. Currently, main methods for face recognition include Linear Subspace Method (e.g., PCA , LDA etc.), Nonlinear Subspace Method, i.e. Kernel Method (e.g., Kernel PCA etc.), Elastic Graph Matching (e.g., DLA etc.), Neural Network-Based Method (e.g., CNN etc.), Hidden Markov Model and so on. In this work we firstly present a novel kernel method - all common subsequences (ACS) - for face recognition. ACS [1] was designed for solving the time series problem. The main concept of ACS is to measure the similarity of two sequences by counting the number of all common subsequences of these two sequences. For the use of ACS for images, we need to translate images to sets of sequences, and then individually use ACS for each pairs of sequences to compute the similarity, in the end average the results as the similarity between images.

II. ALL COMMON SUBSEQUENCES

All common subsequences was firstly proposed in [1] for the purpose of measuring the similarity of time series. Then it was extensively studied in [2] as a problem of computer science. Later ACS was proved to be a valid kernel [3], thereby ACS could also be studied in kernel machine.

Let $\sum$ be a finite alphabet. An n-long sequence t is an ordered set $\{t_1, \cdots, t_n\}$, where $t_i \in \sum$ is the i-th element in sequence t, $1 \le i \le n$. An empty sequence is denoted by ε, whose length is 0. Let u be a sequence. If there exist indices $i = (i_1, \cdots, i_{|u|})$, with $1 \le i_1 < \cdots < i_{|u|} \le |t|$, such that $u_j = t_{i_j}$, for $j = 1, \cdots, |u|$, then we say u is a subsequence of t (denoted by $u \le t$ or $u = t(i)$). We denote by $\sum^n$ the set of all finite sequences of length n, and by $\sum^*$ the set of all sequences

$$\sum^* = \bigcup_{n=0}^{\infty} \sum^n$$

Formally, ACS could be described as

Definition 1: [All common subsequences] Let I be a feature (subsequence) space of sequences set S, lets, t be sequences, let

$$\phi_u(s) = \begin{cases} 1 & |\{i: u = s(i)\}| > 0, u \in I, \\ 0 & \text{otherwise.} \end{cases}$$

Then, ACS can be defined as an inner product of vectors of $\phi_u(s)$:

$$acs(s,t) = \langle \phi_u(s), \phi(t) \rangle = \sum_{u \in I} \phi_u(s)\phi_u(t) \quad (1)$$

Dynamic approach is adopted for calculating the number of ACS. Lemma 1 implies quadratic operations.

Lemma 1: Let s and t be finite, nonempty sequences over $\sum$ with lengths $|s| = m$ and $|t| = n$, respectively. For each $p \in \sum$, let $r(s,p) \coloneqq \max\{i : s_i = p\}$ with $r(s, b) \coloneqq 0$ if $p > s$. For brevity, we set $s^i \coloneqq s(1{:}i)$, $r_s \coloneqq r(s^{m-1}, s_m)$ and $r_t \coloneqq r(t^{n-1}, t_n)$. Then

$$\mathrm{acs}(s,t) = \begin{cases} \mathrm{acs}(s^{m-1},t) + \mathrm{acs}(s,t^{n-1}) - \mathrm{acs}(s^{m-1},t^{n-1}) \\ \quad \text{if } s_m \neq t_n, \\ \mathrm{acs}(s^{m-1},t^{n-1})\times 2 - \mathrm{acs}(s^{r_s-1},t^{r_t-1}) \\ \quad \text{if } s_m = t_n, 0 < r_s < m \text{ and } 0 < r_t < n, \\ \mathrm{acs}(s^{m-1},t^{n-1})\times 2 \text{ otherwise.} \end{cases} \quad (2)$$

All Common Subsequences for Images: The visual appearance of objects in computing science is represented by digital images, which have a finite set of digital values, called pixels. Let I be an image with m×n pixels. Since each pixel of the image carries a single integer value (0 1 for binary and 0 255 for grayscale image), the image could be described by an matrix with integer value entries, denoted also by I.

Let R^i, C^j be the i-th row, j-th column of the image I, respectively. Then I could be deemed as the orderly combination of its rows $I = [R^1; R^2; \cdots; R^m]$ or the orderly combination of its columns $I = [C^1; C^2; \cdots; C^n]$, where each row is a sequence with length |R| = n and each column with length |C| = m. Thereby we transformed an image to a set of sequences: $I \Rightarrow \{R^1, R^2, \cdots, R^m, C^1, C^2, \cdots, C^n\}$. For convenience, we denote R^i by $I^i, i = 1 \cdots m$, and denote C^j by $I^{m+j}, j = 1 \cdots n$. Then the sequences set of image I is

$$I \Rightarrow \bigcup_{i=1}^{m+n} I.$$

Definition 2: The similarity of two images I, J with the same size m×n is the sum of number of all common subsequences of corresponding rows and columns of images I and J:

$$\mathrm{acsi}(I,J) = \textstyle\sum_{i=1}^{m+n} \mathrm{acs}(I^i, j^i). \quad (3)$$

Formally, ACS should be normalized:

$$\widehat{acsi}(I,J) = \frac{1}{m+n} \sum_{i=1}^{m+n} \frac{acs(I^i,J^i)}{\sqrt{acs(I^i,I^i)acs(J^i,J^i)}}. \quad (4)$$

The computational efficiency of ACS of two sequences with length m and n is O(mn). So the computational efficiency of ACS of two images with size m × n is$O(m^2n + mn^2) = O(\max(m^2n, mn^2))$.

III. FACE RECOGNITION EXPERIMENTS

Before calculating the value of ACS of images, the original images should be preprocessed. Generally, image preprocessing consists of the following procedures:

- Cropping - Some images may contain wide background, which should be cropped off.
- Converting - Original images sometimes are color, while ACS only concern the luminance of images. So we need to convert color images to grayscale by eliminating the hue and saturation information while retaining the luminance.

Compensation and Equalization - Face under different lightening conditions shows appearances with different or unbalanced luminance values, while ACS directly compares the gray value of images. So we should make sure that the gray value of the same face under various lightening conditions is invariant. In this work we compensate the face illumination by the technic based on wavelet transform[4], and equalize the pixels of each gray value of images using histogram equalization.

Resizing - Images from datasets have high (spatial) resolution (e.g. 256 × 256 or higher). But in experiments, the time for computing ACS of images is proportionate to the cubic of image size. So we should make the resolution of images as low as possible.

Rescaling - Grayscale images often possess high gray-level resolution (e.g. 8-bit), which may cause ACS not robust, i.e. make ACS sensitive to the noise or slight changes of luminance. Hence it is required to adjust high gray-level resolution to lower. Experimental results suggest that under 1-bit gray-level the recognition performs best. For pixel with gray value p, we can decrease the gray-level resolution in this way: $\tilde{p} = [\frac{p}{\mu}]$, where $\tilde{p}$ is a new gray value and $\mu \in [100,140]$ is a rescaling factor.

Rescaling factor μ is a key parameter of ACS for face recognition, for the value of significantly affects the recognition accuracy. As for 2DPCA and 2DLDA, the rescaling step is not demanded.

A. Face recognition on public databases

We use ACS with SVM [5] and kNN for face recognition on 4 public databases: Caltech[6], Jaffe[7], Orl [8] and Yale [9] face databases. In order to make the experiments comparable, two classic face recognition methods are adopted in experiments: 2DPCA [10] and 2DLDA[11].

The penalty parameters of SVM is set to C = 10. The parameter k of kNN is tuned to the value which makes the recognition get the highest accuracy. The preprocessing steps, including cropping, converting, compensation, equalization, resizing, rescaling, are orderly carried out if necessary. The images are resized to 32 × 32. The rescaling factor μ is tuned carefully to achieve the best recognition results. All experiments, if not specialized, are performed with leave-one-out strategy.

From Caltech database we choose 395 images for face recognition experiment, which consist of 19 subjects (each one has 18 25 images). All images have various backgrounds. So image preprocessing steps include cropping, converting, equalization, resizing, rescaling.

Jaffe database contains 213 images of 7 facial expressions (anger, disgust, fear, happiness, neutral, sadness, surprise) posed by 10 Japanese female expressers. Image preprocessing steps include cropping, converting, equalization, resizing, rescaling.

Orl database consists of 40 distinct subjects, each of which has ten different images. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for slight side movement). The image preprocessing steps only include converting, resizing, rescaling.

Yale face database includes 15 subjects with 11 images per subject: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and winking. So the unbalanced light exists in this database. Hence the image preprocessing steps include cropping, converting, compensation, equalization, resizing, rescaling.

Fig. 1, Fig. 2, Fig. 3, Fig. 4 depict some faces from Caltech, Jaffe, Orl, Yale databases (the upper photos) and the preprocessed ones (the nether photos), respectively.

The experimental results of face recognition are shown on Table I. From the table we find that

- ACS with kNN outperforms other methods in most cases;
- Algorithms with kNN in some cases perform better than that with SVM.

In addition, we also made some robust tests, including faces with various backgrounds, faces with Gaussian noise, small training set, faces with partial occlusion.

Figure 1. Some faces from Caltech database. Upper: raw faces; Nether: preprocessed faces

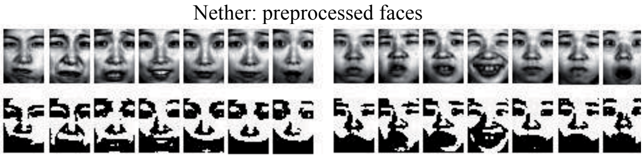

Figure 2. Some faces from Jaffe database. Upper: raw faces; Nether: preprocessed faces

Figure 3. Some faces from Orl database. Upper: raw faces; Nether: preprocessed faces

Figure 4. Some faces from Yale database. Upper: raw faces; Nether: preprocessed faces

TABLE I.
FACE RECOGNITION ACCURACY (%) ON 4 DATABASES

Method	Caltech	Jaffe	Orl	Yale
acs_svm	99.24	100.0	100.0	96.97
2dpca_svm	99.75	100.0	98.25	97.58
2dlda_svm	99.49	100.0	98.5	98.18
acs_knn	98.99	100.0	100.0	99.39
2dpca_knn	97.97	99.06	98.25	98.18
2dlda_knn	98.99	100.0	97.5	96.97

Face with various backgrounds: The same as the normal test, 395 images with various backgrounds from Caltech database are used for robust test. In this case we do not crop the faces from backgrounds, meanwhile we add compensation step for the appearance of unbalanced lights. So the image preprocessing steps include converting, compensation, equalization, resizing, rescaling. Fig. 5 shows some faces (without cropping) from Caltech databases (the upper photos) and the preprocessed ones (the nether photos). The robust test results are shown on Table II.

Figure 5. Some faces (without cropping) from Caltech database. Upper: raw faces; Nether: preprocessed faces

TABLE II.
FACE RECOGNITION ACCURACY (%) ON CALTECH DATABASE (WITHOUT CROPPING)

Method	acs	2dpca	2dlda
svm	93.16	97.72	96.96
knn	84.30	88.35	91.14

Faces with Gaussian noise: We add Gaussian noise $N = (0, \sigma^2)$ to Jaffe faces, where $\sigma = \{25.5, 51, 102, 204\}$ is a grayscale deviation. Fig. 6 shows some faces (with Gaussian noise) from Jaffe databases (the upper photos) and the preprocessed ones (the nether photos). We made two robust tests: use the faces without noise for training and the noisy faces for test (the test results shows on Table Ⅲ); both the training and test faces are noisy (the test results shows on Table Ⅳ).

Figure 6. Some faces (with Gaussian noise $N = (0, \sigma^2)$, $\sigma = \{0, 25.5, 51, 102, 204\}$) from Jaffe database. Upper: raw faces; Nether: preprocessed faces

TABLE III.
FACE RECOGNITION (ONLY TEST FACES ARE NOISY) ACCURACY (%) ON JAFFE DATABASE (WITH GAUSSIAN NOISE $N = (0, \Sigma^2)$, $\Sigma = \{0,25.5,51,102,204\}$)

σ / Method	0	25.5	51	102	204
acs_svm	100.0	99.06	98.59	97.18	96.71
2dpca_svm	100.0	100.0	99.53	98.59	99.06
2dlda_svm	100.0	99.53	99.53	99.06	99.53
acs_knn	100.0	100.0	100.0	100.0	93.90
2dpca_knn	99.06	98.59	99.53	98.59	96.24
2dlda_knn	99.53	99.53	99.53	98.12	97.65

TABLE IV.
FACE RECOGNITION (BOTH TRAINING AND TEST FACES ARE NOISY) ACCURACY (%) ON JAFFE DATABASE (WITH GAUSSIAN NOISE $N = (0, \Sigma^2)$, $\Sigma = \{0,25.5,51,102,204\}$)

σ / Method	0	25.5	51	102	204
acs_svm	100.0	100.0	99.06	90.14	61.50
2dpca_svm	100.0	100.0	100.0	93.43	76.53
2dlda_svm	100.0	99.53	99.06	95.77	78.40
acs_knn	100.0	100.0	99.53	98.59	88.73
2dpca_knn	99.06	98.59	99.53	99.06	96.24
2dlda_knn	99.53	99.06	99.06	97.18	88.26

Small training set: In database Orl, there are 40 subjects, each of which has 10 images. In place of leave-one-out strategy, we reduce the training images to make robust test. We randomly choose k images from each subject, who has 10 images, and the rest $10 - k$ images are for test. The test results are presented on Table Ⅴ.

Faces with partial occlusion: We partially occlude Yale faces by square blocks at random position of images with random gray value. The area of square blocks is k%, k = 0, 5, 10, 15, 25 of the area of images. Fig. 7 depicts some faces (with square blocks) from Yale database (the upper photos) and the preprocessed ones (the nether photos). The robust test results are shown on Table Ⅵ.

TABLE V.
FACE RECOGNITION ACCURACY (%) ON ORL DATABASE (K IMAGES OF EACH SUBJECT FOR TRAINING, THE REST 10−K FOR TEST)

k / Method	9	8	6	4	2	1
acs_svm	99.65	99.26	98.39	95.03	86.81	80.82
2dpca_svm	98.08	98.13	96.96	94.40	83.86	68.35
2dlda_svm	98.30	97.71	95.15	86.92	38.25	4.81
acs_knn	100.0	99.84	99.16	97.06	90.05	81.03
2dpca_knn	98.32	97.38	95.89	92.50	82.75	69.71
2dlda_knn	96.42	93.28	88.53	78.23	42.03	28.93

Figure 7. Some faces (with square block whose area is *k*%, *k* = 0, 5, 10, 15, 25 of the image) from Yale database. Upper: raw faces; Nether: preprocessed faces.

TABLE VI.
FACE RECOGNITION ON YALE DATABASE (WITH SQUARE BLOCK WHOSE AREA IS K%, K = 0, 5, 10, 15, 25 OF THE IMAGE)

k / Method	0	5	10	15	25
acs_svm	96.97	95.15	95.76	93.33	85.45
2dpca_svm	97.58	96.97	95.76	95.76	85.45
2dlda_svm	98.18	98.18	97.58	97.58	90.91
acs_knn	99.39	98.79	97.58	96.97	92.12
2dpca_knn	98.18	97.58	97.58	96.36	92.12
2dlda_knn	96.97	97.58	95.76	95.15	95.15

IV. CONCLUSIONS

This paper presents a sequence similarity, called all common subsequences (ACS), for use with support vector machine (SVM) and k-nearest neighbors (kNN) to the face recognition problem. We first decompose face images as row and column sequences. Then use ACS, which compares two sequences by counting the number of occurrence of common subsequences, to measure the similarity of each pair of corresponding sequences in two images and the average of similarity of all pairs of sequences is proposed to be the similarity of two images. Experiments on 4 public face databases: Caltech, Jaffe, Orl and Yale databases, demonstrate that ACS can achieve higher recognition accuracy than some classic face recognition methods, e.g. 2DPCA and 2DLDA.

REFERENCES

[1] Wang, H., 2007. All common subsequences. Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07), Hyderabad, India, 635–640.

[2] Yang, J., Zhang, D., &Frangi, A.F., 2004 . Two-dimensional PCA: A New Approach to Appearance-based Face Representation and Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1(26), 131–137. http://dx.doi.org/10.1109/TPAMI.2004.1261097

[3] Elzinga, C., Rahmann S., & Wang H., 2008. Algorithms for subsequence combinatorics, Theoretical Computer Science, 3(409), 394–404. http://dx.doi.org/10.1016/j.tcs.2008.08.035

[4] Guo, Z., Wang, H., Lin Z., & Guo X., 2010. A study of all common subsequences in kernel machine, Proceedings of the international conference on machine learning and cybernetics (ICMLC) 2010, Qingdao, China, 1763–1766.

[5] Nie, X.F., Tan, Z.F., & Guo, J., 2008. Face illumination compensation based on wavelet transform, Optics and Precision Engineering, 16(1),150-155

[6] Canu, S., Grandvalet Y., Guigue V., & Rakotomamonjy A., 2005. SVM and Kernel Methods Matlab Toolbox, Perception Systems et Information, INSA de Rouen, Rouen, France.

[7] Fei-Fei, L., Fergus, R., & Perona, P., 2004. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories, IEEE CVPR 2004, Workshop on Generative-Model Based Vision.

[8] Lyons, M.J., Akamatsu, S., Kamachi, M., & Gyoba, J., 1998. Coding Facial Expres- sions with Gabor Wavelets, In Proc. Int'l Conf. Automatic Face and Gesture Recognition, 200-205. http://dx.doi.org/10.1109/AFGR.1998.670949

[9] Samaria, F., & Harter, A., 1994. Parameterization of a stochastic model for human face identification, In Proc. IEEE Workshop on Applications of Computer Vision, 138-142.

[10] Belhumeur, P.N., 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, IEEE Trans. on PAMI, 19(7),711-720. http://dx.doi.org/10.1109/34.598228

[11] Li, M., & Yuan, B., 2005. 2D-LDA: A statistical linear discriminant analysis for image matrix, Pattern Recognition Letters, 5(26), 527–532. http://dx.doi.org/10.1016/j.patrec.2004.09.007

AUTHORS

C. H. Wang is with College of Shipbuilding Engineering, Harbin, 150001 PRC (e-mail: chunhui_wang@ hrbeu.edu.cn).

A. K. Hu is with College of Shipbuilding Engineering, Harbin, 150001 PRC (e-mail: annkanghu@ live.cn).

F. L. Han is with College of Shipbuilding Engineering, Harbin, 150001 PRC (e-mail: Fenglei_Han@ hrbeu.edu.cn).

This article is an extended and modified version of a paper presented at the 2013 Chinese Intelligent Automation Conference (CIAC2013), held in Yangzhou, Jiangsu Province, China, in August 2013. Submitted 19 May 2013. Published as resubmitted by the authors 23 July 2013.