

Research Article

Generalized Discriminant Orthogonal Nonnegative Tensor Factorization for Facial Expression Recognition

Zhang XiuJun^{1,2} and Liu Chang^{1,2}

¹ College of Information Science and Technology, Chengdu University, Chengdu 610106, China

² Key Laboratory of Pattern Recognition and Intelligent Information Processing in Sichuan, Chengdu 610106, China

Correspondence should be addressed to Liu Chang; chang.liu.scu@gmail.com

Received 4 August 2013; Accepted 6 January 2014; Published 26 March 2014

Academic Editors: S. Bourennane and J. Marot

Copyright © 2014 Z. XiuJun and L. Chang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to overcome the limitation of traditional nonnegative factorization algorithms, the paper presents a generalized discriminant orthogonal non-negative tensor factorization algorithm. At first, the algorithm takes the orthogonal constraint into account to ensure the nonnegativity of the low-dimensional features. Furthermore, the discriminant constraint is imposed on low-dimensional weights to strengthen the discriminant capability of the low-dimensional features. The experiments on facial expression recognition have demonstrated that the algorithm is superior to other non-negative factorization algorithms.

1. Introduction

Over the past few years, the nonnegative matrix factorization algorithm (NMF) [1] and its variants have proven to be useful for several problems, especially in facial image characterization and representation problems [2–8]. The idea of nonnegative factorization is partly motivated by the biological fact that the firing rates in visual perception neurons are nonnegative.

However, NMF and its variants have some drawbacks. First of all, NMF requires that all object images should be vectorized in order to find the non-negative decomposition. This vectorization leads to information loss, since the local structure of the image is lost. Moreover, NMF is not unique [9, 10]. In order to remedy these drawbacks, non-negative tensor factorization (NTF) has been proposed [11–13]. NTF represents a facial expression database as a three-order tensor. The tensor representation avoids the vectorization operation and preserves the structure of the data. Under some mild conditions, NTF is unique. Existing NMF and NTF algorithms project data into low-dimensional space with the inverse or pseudoinverse of the basis images, so both of them cannot guarantee the nonnegativity of low-dimensional features, which restricts the application of non-negative

factorization in real world. Furthermore, NTF do not take into account class information in data samples. Actually, it is believed that those features with discriminant constraints are of great importance for pattern recognition. Reference [14] develops a discriminant non-negative tensor factorization algorithm (DNNTF), which adds fisher discriminant constraint into the objective function. But like other discriminant non-negative matrix factorizations [6, 15–18], DNNTF employed discriminant analysis on the representation coefficients and not on the actual features used in the recognition procedure. The actual features used for recognition are derived from the projection of data samples to the bases matrix and only implicitly depend on the representation coefficients.

Based on the above analysis, the paper proposes a generalized discriminant orthogonal non-negative tensor factorization algorithm (GDONTF), which makes full use of the class information and imposes the orthogonal constraint to the objective function. The algorithm not only guarantees the non-negativity of low-dimensional features, but also generalizes discriminant constraints to low-dimension features. The experiments on facial expression recognition indicate that GDONTF achieves better performance than other non-negative factorization algorithms.

2. Generalized Discriminant Orthogonal Non-Negative Tensor Factorization

Consider an N order tensor $X \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_N}$, every data sample X_i is an $n-1$ order tensor; that is, $X_i \in \mathbb{R}^{d_1 \times d_2 \times \dots \times d_{n-1}}$, in which d_1, d_2, \dots, d_{N-1} , is the dimensionality and d_N is the number of data set. The data set is divided into C classes. Data samples belonging to class c denote $V(c)$; the number of data samples in $V(c)$ is N_c . In order to guarantee the non-negativity of low-dimensional features and take use of the class information, we propose generalized discriminant orthogonal non-negative tensor factorization algorithm; the objective function of which is defined as follow:

$$O = \left\| X - \sum_{r=1}^R U_{:,r}^{(1)} \circ U_{:,r}^{(2)} \dots \circ U_{:,r}^{(N)} \right\|^2 + \alpha \text{tr} [S_w] - \beta \text{tr} [S_b] \quad (1)$$

$$\text{s.t. } U^{(n)} U^{(n)T} = I \quad n = 1, 2, \dots, N-1.$$

In which, $\alpha \geq 0, \beta \geq 0, U_{ij}^{(n)} \geq 0, n = 1, 2, \dots, N, I$ is the identity matrix and S_w and S_b are the within- and between-class scatter matrices of the low-dimensional features, respectively. Because $U^{(n)} U^{(n)T} = I$, low-dimensional features can be computed as follows:

$$h_i = \left(U^{(N-1)} \circ U^{(N-2)} \circ \dots \circ U^{(2)} \circ U^{(1)} \right)^T \bar{X}_i = W^T \bar{X}_i, \quad (2)$$

where the basis matrix $W = U^{(N-1)} \circ U^{(N-2)} \circ \dots \circ U^{(2)} \circ U^{(1)}$. Let h_i be the low-dimensional features of the sample X_i ; then the feature matrix $H \in \mathbb{R}^{d \times M}$ consists of all low-dimensional features, d is the low dimensionality of samples, and M is the number of all samples. Actually, the separability of the weight coefficient has nothing to do with the recognition accuracy, while the class separability of the low-dimensional features has a great influence on the recognition accuracy. Consequently, the within- and between-class scatter matrices are defined as follows:

$$\begin{aligned} S_w &= \sum_{c=1}^C \sum_{u_i \in V(c)} (h_i - m_c)(h_i - m_c)^T \\ &= \sum_{c=1}^C \sum_{u_i \in V(c)} (W^T \bar{X}_i - m_c)(W^T \bar{X}_i - m_c)^T, \quad (3) \\ S_b &= \sum_{c=1}^C N_c (m_c - m)(m_c - m)^T, \end{aligned}$$

where m_c is the mean of the low-dimensional features in the class c and m is the mean of all low-dimensional features. The objective function in (1) can be written as the following optimization problem:

$$\begin{aligned} \min_{U^{(n)}} & \left\| X - \sum_{r=1}^R U_{:,r}^{(1)} \circ U_{:,r}^{(2)} \dots \circ U_{:,r}^{(N)} \right\|^2 + \alpha \text{tr} [S_w] - \beta \text{tr} [S_b] \\ \text{s.t. } & U_{ij}^{(n)} \geq 0 \\ & U^{(n)T} U^{(n)} = I, \quad n = 1, 2, \dots, N-1. \end{aligned} \quad (4)$$

Since the basis matrix W consists of the projection matrices $U^{(n)}, n = 1, 2, \dots, N-1$, we solve the projection matrices $U^{(n)}, n = 1, 2, \dots, N-1$, and the weight matrix $U^{(N)}$, respectively, to deal with the optimization problem (4). First of all, we formulate the Lagrange multipliers out of the constrained optimization problem in (4):

$$\begin{aligned} f(U^{(n)}, \lambda) &= \|X_{(n)} - U^{(n)} G^T\|^2 + \alpha \text{tr} [S_w] - \beta \text{tr} [S_b] \\ &\quad + \text{Tr} \left[\lambda \left(U^{(n)T} U^{(n)} - I \right) \right], \end{aligned} \quad (5)$$

where $G = U^{(N)} \circ \dots \circ U^{(n+1)} \circ U^{(1)} \circ U^{(2)} \circ \dots \circ U^{(n-1)}$.

Take the derivative of $f(U^{(n)}, \lambda)$ with respect to $U^{(n)}$ and $\lambda, n = 1, 2, \dots, N-1$; we have

$$\begin{aligned} \frac{\partial f}{\partial U^{(n)}} &= -2X_{(n)}G + 2U^{(n)}G^TG + 2U^{(n)}\lambda + \alpha \frac{\partial \text{Tr}(S_w)}{\partial U^{(n)}} \\ &\quad - \beta \frac{\partial \text{Tr}(S_b)}{\partial U^{(n)}} \\ &= -2X_{(n)}G + 2U^{(n)}G^TG + 2U^{(n)}\lambda + \alpha \nabla \text{Tr}(S_w) \\ &\quad - \beta \nabla \text{Tr}(S_b), \\ \frac{\partial f}{\partial \lambda} &= U^{(n)T} U^{(n)} - I. \end{aligned} \quad (6)$$

Set (6) and (7) to zeros; we get

$$-2X_{(n)}G + 2U^{(n)}G^TG + 2U^{(n)}\lambda + \alpha \nabla \text{Tr}(S_w) - \beta \nabla \text{Tr}(S_b) = 0, \quad (8)$$

$$U^{(n)T} U^{(n)} = I. \quad (9)$$

Left multiply both side of (8) by $U^{(n)T}$; we immediately have

$$\begin{aligned} -2U^{(n)T} X_{(n)}G + 2U^{(n)T} U^{(n)}G^TG + 2U^{(n)T} U^{(n)}\lambda \\ + \alpha U^{(n)T} \nabla \text{Tr}(S_w) - \beta U^{(n)T} \nabla \text{Tr}(S_b) = 0. \end{aligned} \quad (10)$$

Therefore, the update rule for $U^{(n)}$ is

$$U^{(n)t+1} = U^{(n)t} * \frac{2X_{(n)}G + \beta \nabla \text{Tr}(S_b)}{2U^{(n)}G^TG + 2U^{(n)}\lambda + \alpha \nabla \text{Tr}(S_w)}. \quad (11)$$

The gradient $[\nabla \text{Tr}(S_b)]_{i,j} = \partial \text{Tr}[S_b] / \partial u_{ij}^{(n)}$ is given by

$$\begin{aligned} \frac{\partial \text{Tr}[S_b]}{\partial u_{ij}^{(n)}} &= \frac{\partial \sum_p \sum_{c=1}^C \sum_{h_p \in V(c)} (h_p - m_c)(h_p - m_c)^T}{\partial u_{ij}^{(n)}} \\ &= \frac{\partial \sum_p \sum_{c=1}^C \sum_{h_p \in V(c)} (W^T \bar{X}_p - m_c)(W^T \bar{X}_p - m_c)^T}{\partial u_{ij}^{(n)}}. \end{aligned} \quad (12)$$

Because $W = (U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(2)} \odot U^{(1)})$, we can get

$$\begin{aligned} & \frac{\partial Tr [S_w]}{\partial u_{ij}^{(n)}} \\ &= \left(\partial \sum_p \sum_{c=1}^C \sum_{h_p \in V(c)} \left((U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(2)} \odot U^{(1)})^T \right. \right. \\ & \quad \left. \left. \times \overline{X_p} - m_c \right)^2 \right) \times (\partial u_{ij}^{(n)})^{-1} \\ &= 2 \sum_p \sum_{c=1}^C \sum_{h_p \in V(c)} (W^T \overline{X_p} - m_c) \\ & \quad \times \left(\partial \left((U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(1)})^T \right. \right. \\ & \quad \left. \left. \times \left(\overline{X_p} - \sum_{\substack{q=1 \\ h_q \in V(c)}}^{N_c} \overline{X_q} \right) \right) \right) \times (\partial u_{ij}^{(n)})^{-1}. \end{aligned} \tag{13}$$

Let $U^{pn} = U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(n+1)}$ and $U^{an} = U^{(n-1)} \odot U^{(n-2)} \odot \dots \odot U^{(1)}$; we have

$$\begin{aligned} & \frac{\partial Tr [S_w]}{\partial u_{ij}^{(n)}} \\ &= 2 \sum_p \sum_{c=1}^C \sum_{h_p \in V(c)} (W^T \overline{X_p} - m_c) \\ & \quad \times \left(\partial \left((U^{pn} \odot U^{(n)} \odot U^{an})^T \right. \right. \\ & \quad \left. \left. \times \left(\overline{X_p} - \sum_{\substack{q=1 \\ h_q \in V(c)}}^{N_c} \overline{X_q} \right) \right) \right) \times (\partial u_{ij}^{(n)})^{-1}; \end{aligned} \tag{14}$$

Since

$$\begin{aligned} & U^{pn} \odot U^{(n)} \odot U^{an} \\ &= [U_1^{pn} \otimes U_1^{(n)} \otimes U_1^{an}, \dots, U_j^{pn} \otimes U_j^{(n)} \otimes U_j^{an}, \dots] \\ &= [U_{:,1}^{pn} \otimes U_{:,1}^{(n)} \otimes U_{:,1}^{an}, \dots, U_{:,j}^{pn} \otimes U_{:,j}^{(n)} \otimes U_{:,j}^{an}, \dots] \tag{15} \\ &= [u_{11}^{pn} u_{11}^{(n)} u_{11}^{an}, \dots, u_{1j}^{pn} u_{1j}^{(n)} u_{1j}^{an}, \dots, \\ & \quad u_{1j}^{pn} u_{ij}^{(n)} u_{1j}^{an}, \dots, u_{ij}^{pn} u_{ij}^{(n)} u_{ij}^{an}, \dots]. \end{aligned}$$

We have

$$\begin{aligned} \frac{\partial Tr [S_w]}{\partial u_{ij}^{(n)}} &= 2 \sum_p \sum_{c=1}^C \sum_{h_p \in V(c)} (W^T \overline{X_p} - m_c) \\ & \quad \times [0, \dots, 0, u_{1j}^{pn} u_{1j}^{an}, \dots, u_{ij}^{pn} u_{ij}^{an}, \dots, 0, \dots, 0] \overline{X_p}. \end{aligned} \tag{16}$$

Similarly, we have

$$\begin{aligned} & \frac{\partial Tr [S_b]}{\partial u_{ij}^{(n)}} \\ &= \frac{\partial \sum_p \sum_{c=1}^C N_c (m_p^c - m_p) (m_p^c - m_p)^T}{\partial u_{ij}^{(n)}} \\ &= \frac{\partial \sum_p \sum_{c=1}^C N_c \left(\sum_{h_p \in V(c)}^{N_c} W^T \overline{X_p} - \sum_{q=1}^M W^T \overline{X_q} \right)^2}{\partial u_{ij}^{(n)}} \\ &= 2 \sum_p \sum_{c=1}^C N_c (m_p^c - m_p) \\ & \quad \times \frac{\partial \left(\sum_{h_p \in V(c)}^{N_c} W^T \overline{X_p} - \sum_{q=1}^M W^T \overline{X_q} \right)}{\partial u_{ij}^{(n)}} \\ &= 2 \sum_p \sum_{c=1}^C N_c (m_p^c - m_p) \\ & \quad \times \left(\partial (U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(1)})^T \right. \\ & \quad \left. \times \left(\sum_{h_p \in V(c)}^{N_c} \overline{X_p} - \sum_{q=1}^M \overline{X_q} \right) \right) \times (\partial u_{ij}^{(n)})^{-1} \\ &= 2 \sum_p \sum_{c=1}^C N_c (m_p^c - m_p) \\ & \quad \times [0, \dots, 0, u_{1j}^{pn} u_{1j}^{an}, \dots, u_{ij}^{pn} u_{ij}^{an}, \dots, 0, \dots, 0] \\ & \quad \times \left(\sum_{h_p \in V(c)}^{N_c} \overline{X_p} - \sum_{q=1}^M \overline{X_q} \right). \end{aligned} \tag{17}$$

To solve the weight matrix $U^{(N)}$, the objective function is

$$f(U^{(N)}) = \left\| X_{(N)} - U^{(N)} (U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(1)})^T \right\|^2. \tag{18}$$

The gradient function is

$$g(U^{(N)}) = -2X_{(N)}W + 2U^{(N)}W^TW, \tag{19}$$

where $W = U^{(N-1)} \odot U^{(N-2)} \odot \dots \odot U^{(1)}$.

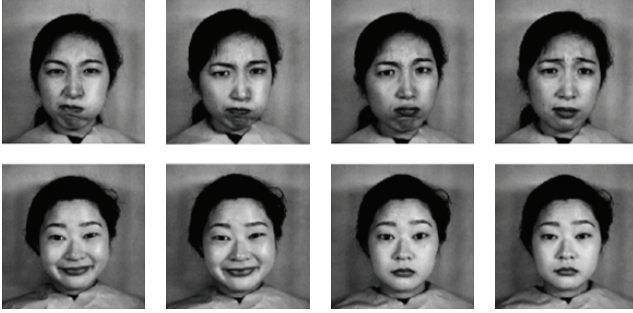


FIGURE 1: Some images in the Jaff facial expression database.

TABLE 1: Comparison of the best recognition rates for all tested algorithms.

Algorithms	Recognition rate	Algorithms	Recognition rate
NMF	79.19%	NMFOS	89.06%
DNMF	92.06%	FisherNMF	92.06%
DNTF	95.24%	GDONTF	97.07%

Consequently, the update rules of $U^{(N)}$ are

$$U_{ij}^{(N)^{t+1}} = U_{ij}^{(N)^t} \frac{(X_{(N)}W)_{ij}}{(U^{(N)}W^TW)_{ij}}. \quad (20)$$

3. Experiments

We have conducted facial expression recognition in order to compare the GDONTF with other algorithms such as NMFOS [19], DNMF [6], FisherNMF [16], and DNTF [14]. Because these algorithms calculate low-dimension features in iteration form, the iteration number is 100. For NMFOS and GDONTF, $\lambda = 1$. $\gamma = 0.5$ in DNMF and $\alpha = 1$ in FisherNMF. All low-dimension features are classified by SVM with linear kernel.

The database used for the facial expression recognition experiments is Jaff facial expression database [20]. The database contains 213 images of ten Japanese women. Each person has two to four images for each of the seven expressions: neutral, happy, sad, surprise, anger, disgust, and fear. Each image is resized into 32×32 . A few examples are shown in Figure 1. We randomly select 20 images from each expression for training; the rest is used for testing. The recognition rates with various dimensionalities of different algorithms are shown in Figure 2. Table 1 shows the best recognition rates of the above algorithms. Because NMF is unsupervised learning algorithm, it has the lowest recognition rates. DNMF and FisherNMF have better recognition rates with supervised learning. It is interesting that NMFOS is superior to DNMF and FisherNMF when the feature dimensionality is from 16 to 160 and is better than DNTF when the feature dimensionality is from 16 to 40, which also illustrates the validity of the orthogonal constraint. It is obvious that GDONTF outperforms other algorithms and the best recognition rate is up to 97.07%.

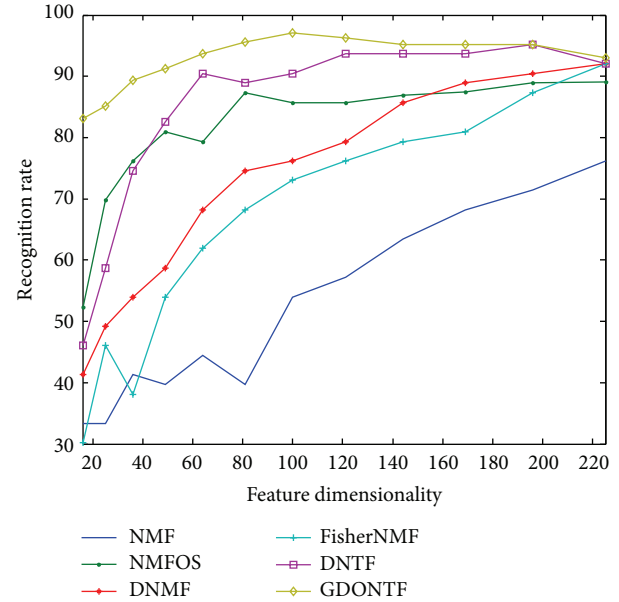


FIGURE 2: Facial expression recognition rate versus dimensionality in Jaff database.

4. Conclusion

In this paper, a generalized discriminant orthogonal non-negative tensor factorization algorithm is proposed considering the orthogonal constraint and the discriminant constraint. For the algorithm, the non-negativity of the low-dimensional features is preserved due to the orthogonal constraint for either training samples or testing samples. In order to enhance the recognition accuracy, the discriminant is conducted on low-dimensional features instead of the weight coefficient of the basis images. The experiments also validate the performance of the algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [2] I. Buciu and I. Pitas, "Application of non-negative and local non negative matrix factorization to facial expression recognition," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR '04)*, pp. 288–291, August 2004.
- [3] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.
- [4] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

- [5] I. Kotsia, S. Zafeiriou, and I. Pitas, "Novel discriminant non-negative matrix factorization algorithm with applications to facial image characterization problems," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3, pp. 588–595, 2007.
- [6] B. Ioan and N. Ioan, "Non-negative matrix factorization methods for face recognition under extreme lighting variations," in *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS '09)*, Iasi, Romania, July 2009.
- [7] J. Lu and Y.-P. Tan, "Doubly weighted nonnegative matrix factorization for imbalanced face recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '09)*, pp. 877–880, IEEE Computer Society, Taipei, Taiwan, April 2009.
- [8] H. Tong and C.-Y. Lin, "Non-negative residual matrix factorization: problem definition, fast solutions, and applications," *Statistical Analysis and Data Mining*, vol. 5, no. 1, pp. 3–15, 2012.
- [9] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts," *Advances in Neural Information Processing Systems*, vol. 16, pp. 1141–1148, 2004.
- [10] J. Liu, J. Liu, P. Wonka, and J. Ye, "Sparse non-negative tensor factorization using columnwise coordinate descent," *Pattern Recognition*, vol. 45, no. 1, pp. 649–656, 2012.
- [11] T. Van De Cruys, "A non-negative tensor factorization model for selectional preference induction," *Natural Language Engineering*, vol. 16, no. 4, pp. 417–437, 2010.
- [12] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 217–235, 2009.
- [13] C. Liu, K. He, J.-L. Zhou, and Y.-L. Zhu, "Facial Expression recognition based on Orthogonal Nonnegative CP Factorization," *Advanced Materials Research*, vol. 143-144, pp. 111–115, 2011.
- [14] S. Zafeiriou, "Discriminant nonnegative tensor factorization algorithms," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 217–235, 2009.
- [15] I. Buciu and I. Pitas, "NMF, LNMF, and DNMF modeling of neural receptive fields involved in human facial expression perception," *Journal of Visual Communication and Image Representation*, vol. 17, no. 5, pp. 958–969, 2006.
- [16] Y. Wang, Y. Jia, H. U. Changbo, and M. Turk, "Non-negative matrix factorization framework for face recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 4, pp. 495–511, 2005.
- [17] X. Li and K. Fukui, "Fisher non-negative matrix factorization with pairwise weighting," in *Proceedings of the IAPR Conference on Machine Vision Applications*, MVA Press, Tokyo, Japan, 2007.
- [18] A. Vilamala, P. J. G. Lisboa, S. Ortega-Martorell, and A. Vellido, "Discriminant convex non-negative matrix factorization for the classification of human brain tumours," *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1734–1747, 2013.
- [19] Z. Li, X. Wu, and H. Peng, "Nonnegative matrix factorization on orthogonal subspace," *Pattern Recognition Letters*, vol. 31, no. 9, pp. 905–911, 2010.
- [20] M. J. Lyons, J. Budynek, and S. Akamatsu, "Automatic classification of single facial images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 12, pp. 1357–1362, 1999.