

ESTUDO COMPARATIVO DE ALGORITMOS PARA RECONHECIMENTO FACIAL

Cristiane Kinuta,
cristiane_kinuta@yahoo.com.br
Dennis Molina,
dennismolina2002@yahoo.com.br
Eric Giovanni Dorneles,
ericgiovani@ig.com.br
Fabio Simeão Grecchi,
fabiogrecchi@ig.com.br
Gilson Torres Dias
gilson@maempec.com.br
Jailton Santana
jailton_santana@uol.com.br
Oswaldo Ortiz Fernandes Junior
oswaldo@imes.edu.br

Universidade IMES - São Caetano do Sul – SP, Brasil

Abstract

With technological advances in biometrics, face recognition is one of the most studied issues in pattern recognition. There are many problems to solve yet like dark images, physical changes in persons, discriminate twins and so on. Our objective is to analyse the techniques and algorithms to see their performance trying to improve them.

Resumo

Com avanço tecnológico na área de biometria, o reconhecimento de faces tem sido um dos mais difundidos, contudo apresenta também muitos problemas a serem resolvidos, dentre eles podemos citar: má qualidade de imagem - devido à variação de luminosidade, interferência do meio-ambiente, alterações físicas das pessoas; dificuldade na diferenciação entre gêmeos; e o uso de equipamentos com definição de imagem inadequada. Temos como objetivo analisar as técnicas existentes, comparando suas performances e tentando identificar suas melhores aplicabilidades, para futuras melhorias na implementação dos algoritmos.

Palavras Chaves: *reconhecimento de faces, reconhecimento de padrões, eigenface, fisherface.*

1 INTRODUÇÃO

Esta área da biometria vem sendo cada vez mais utilizada por suas vantagens em diversas aplicações como, por exemplo: em aeroportos, para identificação de terroristas; automação bancária; reconhecimento de usuários no caixa eletrônico; autenticação para acesso a locais restritos em geral, etc. Entretanto existe uma série de estudos e desenvolvimentos a serem realizados a fim de se obter uma melhor performance destes algoritmos com relação à vulnerabilidade ainda existente no que concerne ao desempenho mediante às variações de condições ambientais e geometria relativa à captura da imagem das faces.

2 MÉTODOS DE RECONHECIMENTO

Para reconhecer uma face específica em um banco de imagens contendo diversas faces, divide-se este processo em três fases de maneira simplificada: detecção, normalização e reconhecimento de faces. Nestas fases podem-se utilizar diversas técnicas, como com o uso de redes neurais artificiais, análise dos componentes principais (PCA) e template matching, para citar alguns.

2.1 Descrição das três fases

2.1.1 Detecção de face

Esta é a primeira etapa do processo de reconhecimento de face, que se resume em localizar uma face em uma imagem que normalmente possui muitos elementos complexos que dificultam o posicionamento de uma face na imagem, para isto podemos utilizar diversas técnicas, como citamos acima. Na figura 1, é mostrada a detecção de uma face em uma imagem.

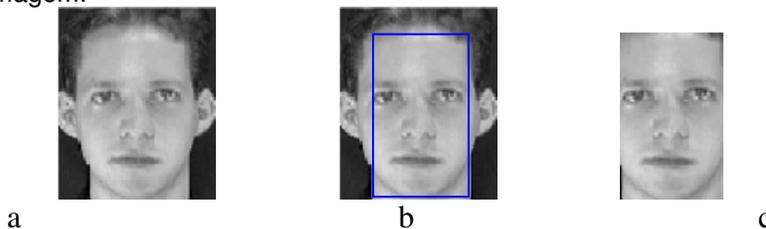


Figura 1.

Exemplo de uma face antes da detecção (a), seleção (b), extração.

2.1.2 Normalização

Esta fase tem a função básica de padronizar a face para que se possa seguir o mesmo tratamento dado às faces existentes no banco de dados, alterando, por exemplo, as características contidas na imagem obtida, tais como alteração do padrão de cor da imagem, seu tamanho e outras características para se assimilarem com as restantes no banco.

2.1.3 Reconhecimento

Após a fase de detecção e normalização contemplamos a fase de reconhecimento de face propriamente dito. Nesta fase avaliamos a imagem a ser reconhecida, comparando com as que estão em um conjunto armazenado no banco, através dos cálculos necessários para efetuarmos essa comparação, acessando-as de maneira mais eficiente possível, ou seja, diminuindo a porcentagem de erro. Para métodos que reduzem o espaço de características, como o PCA, torna-se necessário a utilização de um classificador, que tem como função, determinar valores numéricos singulares para a comparação, verificando os valores que mais se aproximam.

Para uma melhor eficiência na comparação da imagem pesquisada com o banco, utilizam-se diversos algoritmos de faces. Após o levantamento inicial, estaremos tratando dois algoritmos: Eigenface, Fisherface e KDDA. Estes são baseados na média de pixels por face e baseiam-se em aparência.

3 MÉTODO EIGENFACE

O método Eigenface baseia-se em linearmente projetar o espaço de imagens em um espaço de características com dimensões reduzidas obtido fazendo uso da análise de componentes principais (PCA), também conhecido como método Karhunen-Loeve. Entretanto, produz direções de projeção que maximiza a dispersão dos pontos no gráfico em todas as classes, isto é, em todas as imagens faciais mantém as variações indesejadas causadas pela iluminação e expressão facial [BELHUMEUR1997]. Este método baseia-se em autovetores e autovalores de uma matriz simétrica, que é a matriz de covariância.

Os passos principais para a modelagem utilizando este método, PCA, são:

1. Dada uma coleção de m imagens de treinamento identificadas, ou seja, tendo uma base de imagens com cada imagem de tamanho matricial de $n \times o$, com alguma identificação, cria-se uma matriz X_{ij} , onde $j=1,2,\dots,m$ é a quantidade de imagens de treinamento, e i é o tamanho das imagens em formato de vetor, isto é, $i = n \times o$, fazer:
 - a. Computar a imagem média

$$M = \sum_{r=1}^j X_{ir}, i = 1,2,\dots,(n \times o)$$

- b. Centralizar os vetores das imagens subtraindo cada um dos vetores pela média dos vetores encontrados.

$$\bar{X} = X - M$$

- c. Calcular a matriz de covariância

$$\Lambda = M * M^T$$

- d. Computar os k autovetores, v_k , da matriz de covariância correspondente aos k maiores autovalores, λ_k . Como a matriz de covariância é real e simétrica, todos os autovalores e autovetores serão também reais e simétricos [SPERANDIO2003]. Além disso, se essa matriz é de ordem i , então existirá i autovetores associados à i autovalores [BURDEN2003]. Os autovetores são, de certa forma, imagens, que são agrupadas em uma matriz W com k colunas.

$$W_{ik} = \{v_1, v_2, \dots, v_k\}$$

$$\lambda_k = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$$

- e. Projetar cada uma das imagens de treinamento no autoespaço k -dimensional criando um vetor de tamanho reduzido para cada uma das imagens, facilitando a comparação entre os vetores. A projeção é realizada multiplicando cada um dos vetores imagens pelo autoespaço.

$$\hat{X} = W \cdot \bar{X}$$

Os maiores autovalores da matriz de covariância tende não ser fixo. Após ter realizado esses primeiros cálculos na banco de faces, realiza-se o reconhecimento:

1. Dada uma imagem de teste Y , projetá-la no autoespaço, após tê-la centralizada também com aquele mesmo vetor de médias, assim como as de treinamento.

$$\bar{Y} = Y - M$$

$$\hat{Y} = W \cdot \bar{Y}$$

2. Classificá-la com as imagens de treinamento projetadas, fazendo uso de um classificador definido ou, às vezes, pode-se combinar dois ou mais classificadores.

4 MÉTODO FISHERFACES

O discriminante linear de Fisher (FLD), também conhecido com análise de discriminantes linear (LDA), foi desenvolvido por R. A. Fisher na década de 1930, porém, apenas recentemente tem sido utilizado para o reconhecimento de objetos. É um método específico à classe, pois, ele trabalha com o uso de "rótulos", isto é, uma vez identificado os rostos dizendo qual face pertence a qual pessoa, os mesmos são agrupados por pessoa, e cada agrupamento desses é conhecido como classe. O método tenta modelar a dispersão dos pontos visando maior confiabilidade para a classificação. O LDA busca otimizar a melhor linha em uma superfície que separa satisfatoriamente as classes [BELHUMEUR1997].

Inicia-se o algoritmo obtendo as matrizes de dispersão entre classes, interclasse, e dentro das classes, intraclasse. A projeção é feita maximizando a dispersão interclasse e minimizando a intraclasse, formulado pela razão entre as determinantes de ambas as matrizes, com isso diferindo do PCA, que maximiza o espalhamento, dispersão, dos padrões no espaço de características, independente da classe em que esses pertencem [CAMPOS2001] apud [JAIN2000]. As duas medidas citadas, matematicamente são definidas como:

1. matriz de dispersão intraclasses, *within class*:

$$S_w = \sum_{j=1}^c \sum_{i=1}^{|T_j|} (x_i^j - \mu_j) \cdot (x_i^j - \mu_j)^T,$$

em que x_i^j é o i -ésimo exemplo da classe j , μ_j é a média da classe j , c é o número de classes, e $|T_j|$ o número de exemplos na classe j ;

2. matriz de dispersão interclasses, *between class*:

$$S_b = \sum_{j=1}^c (\mu_j - \mu) \cdot (\mu_j - \mu)^T,$$

,em que μ representa a média de todas as classes.

A maximização da medida inter-classes e a minimização da intra-classes são obtidas ao maximizar a taxa $\frac{\det(S_b)}{\det(S_x)}$. O espaço de projeção é então encontrado resolvendo a equação $S_b W = \lambda S_w W$, onde W é a matriz

com autovetores generalizados associados com Λ , que é a matriz diagonal com autovalores. Essas matrizes estão limitadas à ordem $c-1$, em que c é o número de classes, limitação devido à comparação ser realizada entre duas classes diferentes.

Para identificar uma imagem de teste funciona da mesma forma que o Eigenface. A imagem de teste é projetada e comparada com cada uma das faces de treinamento também projetadas, identificando-a com a de treinamento que mais se aproxima. A comparação, de novo, é feita utilizando um classificador específico ou a combinação de dois ou mais.

5 MÉTODO KDDA

O *kernel direct discriminant analysis* (KDDA) é uma técnica proposta por Juwei Lu, et. al. [LU2003], com o intuito de melhorar a atuação de outra como Support Vector Machines (SVM), Kernel PCA (KPCA) [SCHÖLKOPF1999] e Generalized Discriminant Analysis (GDA) em regressão de padrões e tarefas de classificação, superando algumas limitações.

Este método faz uso de funções núcleo (kernel) com o intuito modificar o espaço dimensional, aumentando-o e obtendo uma maneira de dispor os dados de maneira linearmente separáveis. Seja $\varphi : z \in \mathbb{R}^n \rightarrow \varphi(z) \in F$ um mapeamento não linear do espaço de entrada em um espaço F com número de dimensões elevado, em que as classes de objetos supõem ser linearmente separáveis. Assim sendo, $\forall \varphi(z_i), \varphi(z_j) \in F$, assume-se que $\exists k(\bullet)$, tal que $k(z_i, z_j) = \varphi(z_i) \bullet \varphi(z_j)$.

5.1 Análise de autovalores e autovetores de S_b no espaço característica

Começamos resolvendo o problema do autovalor de S_b , que pode ser reescrito como:

$$S_b = \sum_{i=1}^C \left(\sqrt{\frac{C_i}{L}} (\bar{\phi}_i - \bar{\phi}) \right) \left(\sqrt{\frac{C_i}{L}} (\bar{\phi}_i - \bar{\phi}) \right)^T = \sum_{i=1}^C \bar{\phi}_i \bar{\phi}_i^T = \Phi_b \Phi_b^T$$

, onde C é o número de classes de imagens, L é a quantidade total de imagens, $\bar{\phi}_i = \frac{\sum_{j=1}^{C_i} \phi(z_{ij})}{C_i}$ que é a

média da classe Z_i , $\bar{\phi} = \frac{\sum_{i=1}^C \sum_{j=1}^{C_i} \phi(z_{ij})}{L}$ que é a média de todas as imagens, C_i é o número de elementos em

$$Z_i, \bar{\phi}_i = \sqrt{\frac{C_i}{L}} (\bar{\phi}_i - \bar{\phi}) \text{ e } \Phi_b = \begin{bmatrix} \bar{\phi}_1 \\ \Lambda \\ \bar{\phi}_C \end{bmatrix}.$$

Usando a função núcleo, o cálculo de $\Phi_b^T \Phi_b$ pode ser realizado através de:

$$\Phi_b^T \Phi_b = \frac{1}{L} B \cdot \left(A_{LC}^T \cdot K \cdot A_{LC} - \frac{1}{L} (A_{LC}^T \cdot K \cdot \mathbf{1}_{LC}) - \frac{1}{L} (\mathbf{1}_{LC}^T \cdot K \cdot A_{LC}) + \frac{1}{L^2} (\mathbf{1}_{LC}^T \cdot K \cdot \mathbf{1}_{LC}) \right) \cdot B$$

, onde K é uma matriz kernel $L \times L$, $B = \text{diag}[\sqrt{C_1} \dots \sqrt{C_c}]$, $\mathbf{1}_{LC}$ é uma matriz $L \times C$ com todos os termos iguais a 1, $A_{LC} = \text{diag}[a_{C_1} \dots a_{C_c}]$ é uma matriz diagonal de blocos $L \times C$, e a_{C_i} é um vetor $C_i \times 1$ com todos os termos iguais a: $1/C_i$.

Seja λ_i e e_i ($i = 1 \dots C$) como sendo o i -ésimo autovalor e seu correspondente autovetor de $\Phi_b^T \Phi_b$, e que estão em ordem decrescente por seus autovalores. Então, obtêm-se os m maiores autovetores de acordo com seus autovalores: $V = [v_1 \dots v_m] = \Phi_b E_m$. Com isto, percebe-se que $V^T S_b V = \Lambda_b$, sendo $\Lambda_b = \text{diag}[\lambda_1^2 \dots \lambda_m^2]$ uma matriz diagonal $m \times m$.

5.2 Análise de autovalores e autovetores de S_w no espaço característica

Com a matriz núcleo K , uma forma da expressão $\Phi_b^T S_w \Phi_b$ poderá ser obtida da seguinte maneira:

$$\Phi_b^T S_w \Phi_b = \frac{1}{L} (J1 - J2)$$

, onde $J1$ é definido como:

$$J1 = \frac{1}{L} B \left(A_{LC}^T \cdot G \cdot A_{LC} - \frac{1}{L} (A_{Nc}^T \cdot G \cdot 1_{LC}) - \frac{1}{L} (1_{LC} \cdot G \cdot A_{LC}) + \frac{1}{L^2} (1_{LC}^T \cdot G \cdot 1_{LC}) \right) B$$

, e $J2$ é definido como sendo:

$$J2 = \frac{1}{L} B \left(A_{LC}^T \cdot H \cdot A_{LC} - \frac{1}{L} (A_{LC}^T \cdot H \cdot 1_{LC}) - \frac{1}{L} (1_{LC} \cdot H \cdot A_{LC}) + \frac{1}{L^2} (1_{LC}^T \cdot H \cdot 1_{LC}) \right) B$$

, em que $G = KK$, $H = KWK$, e $W = \text{diag}[w_1 \dots w_c]$ é uma matriz diagonal de blocos $L \times L$, e w_i é uma matriz $C_i \times C_i$ com todos os termos iguais a $\frac{1}{C_i}$.

Prossegue-se, diagonalizando a matriz $m \times m$, tratável, $U^T S_w U$. Considere p_i o i -ésimo autovetor de $U^T S_w U$, onde $i = 1 \dots m$, em ordem crescente do seu correspondente autovalor λ^i . Descartamos os autovetores com os maiores autovalores, os M ($\leq m$) selecionados autovetores são denotados como $P = [p_1 \dots p_M]$. Ao definir uma matriz $Q = UP$, podemos obter $Q^T S_w Q = \Lambda_w$, sendo $\Lambda_w = \text{diag}[\lambda_1^i \dots \lambda_M^i]$, uma matriz diagonal $M \times M$.

Com os cálculos apresentados, um conjunto de vetores características pode ser derivado através de $\Gamma = Q \Lambda_w^{-1/2}$. As características formam um subespaço com pequenas dimensões em F , onde maximiza a taxa:

$$\Psi = \arg \max_{\Psi} \frac{|\Psi^T S_b \Psi|}{|\Psi^T S_w \Psi|}$$

5.3 Redução da dimensão e extração de características

Para qualquer padrão de entrada z , sua projeção no conjunto de vetores características, Γ , pode ser calculado por

$$y = \Gamma^T \varphi(z) = \left(E_m \cdot \Lambda_b^{-1/2} \cdot P \cdot \Lambda_w^{-1/2} \right)^T \left(\Phi_b^T \varphi(z) \right)$$

Assim, y é uma representação de pequenas dimensões de z com poder discriminante melhorado, o que atende tarefas de classificação.

6 CLASSIFICADORES

Como visto em ambos os métodos acima, ao chegar em um determinado ponto do processo de reconhecimento de face, torna-se necessário um outro método para realizar a comparação entre dois vetores. Os métodos que andam em voga, são aqueles que medem a distância entre as imagens em um espaço N -dimensional.

A distância euclidiana é norma euclidiana de um vetor X qualquer. Esta norma também é conhecida por norma L_2 . O cálculo dela é executado efetuando a raiz quadrada da soma das diferenças entre dois vetores, isto é, dado um vetor A e B , a norma L_2 é definida como:

$$L_2(A, B) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}$$

Existem outras normas utilizadas para o mesmo fim, como por exemplo a norma L_1 e a distância Mahalanobis.

7 TESTES E RESULTADOS

Para as aplicações dos testes, utilizamos três tipos de bancos de imagens para reconhecimento de faces e para cada banco, foi criado um diretório /RECOG, onde ficam as imagens que desejamos pesquisar e /TRAIN, com o banco de faces propriamente dito.

AT&T – Este banco contém 690 imagens, sendo 10 de cada indivíduo com características de expressão faciais diferentes.

UMIST – O Umist é um banco que possui 671 imagens com diferentes perspectivas de inclinações, totalizando 20 imagens de cada pessoa. [GRAHAM98]
 Yale – As 270 imagens deste banco, incluem 10 imagens por pessoa com variações de iluminação nas faces. [GEORGHIADES01]

7.1 Parametrização de Ambiente.

Os algoritmos testados fazem uso de alguns parâmetros, variáveis, importantes para os testes, e os valores escolhidos variam de 0 a 100, são elas:

TRASHFACE: Esta variável é importante em assumir um valor mínimo para uma imagem a ser classificada com uma face. Assumimos o valor 10, porém em nossos testes, este valor não altera o resultado graças ao fato de que todas as imagens dos bancos possuem faces.

THRESHCLASSE: Os valores aqui assumidos se encaixam mais especificamente em nossos testes, pois cada imagem do banco ao ser processada receberá seu valor correspondente. O threshclass indica a margem de erro para a comparação da imagem pesquisada. Variamos este parâmetro em três valores 3,5 e 6.

NEIGFACE: Escolhe o número de componentes principais (PCA), fixamos em 41.

7.2 Métricas de Testes

O desempenho do reconhecimento de face feito pelo algoritmo, foi medido em porcentagens de acertos e erros em relação à métrica adotada, que é o reconhecimento verdadeiro que corresponde ao acerto de cada imagem encontrada no banco em relação à imagem pesquisada.

7.3 Resultados dos testes

Analisando os testes de reconhecimento verdadeiro, o Eigenface obteve melhor porcentagem de acertos no banco AT&T, utilizando threshclass = 3, já o Fisherface teve melhores resultados com os bancos UMIST e YALE, o KDDA teve um bom desempenho com o banco YALE, porém não superou o Fisherface. Nos testes com o parâmetro threshclass = 5 e 6 o Fisherface se mostrou melhor em ambos os testes.

Para alguns algoritmos o threshclass atingiu valores altos, mas não alcançou reconhecimento total das imagens. O Fisherface obteve reconhecimento máximo com baixos valores limiares de classe, enquanto os outros, necessitaram um ajuste maior.

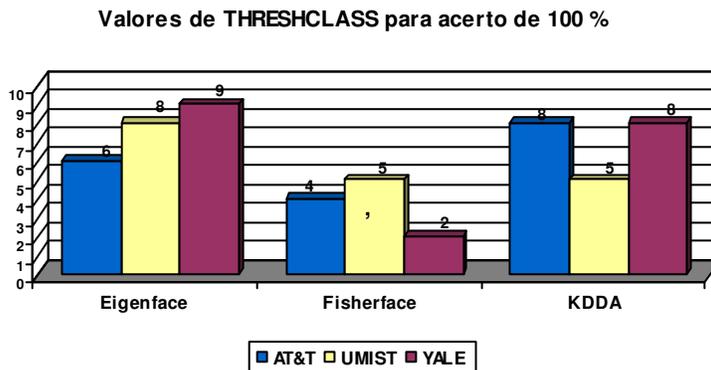


Gráfico 1: Comparativo de threshclass para o reconhecimento máximo de faces.

O problema apresentado foi o de que, ao aumentar esse parâmetro existe a grande chance de reconhecer alguém como uma outra pessoa pois aumentará a range de imagens a ser pesquisadas.

8 CONCLUSÃO E TRABALHOS FUTUROS

Como visto, o objetivo deste artigo foi levantar algumas técnicas, algoritmos, de reconhecimento de faces. Além disso, testamos alguns deles com a finalidade de comparar a eficiência de cada um em termos de reconhecimento verdadeiro. A idéia inicial dos testes incluía também o algoritmo de redes neuronais, entretanto as métricas utilizadas para a avaliação dos algoritmos impossibilitaram a comparação com os outros métodos testados.

Os algoritmos aqui utilizados, foram somente para o reconhecimento de faces. Sendo assim, os testes não isolaram as faces das pessoas para a tarefa de reconhecimento.

Cada algoritmo teve suas vantagens e desvantagens em relação aos testes, sugerimos, com o uso destes bancos, e outros, a implementação de algoritmos mesclando as técnicas aqui apresentadas, a fim de realmente isolar o rosto do resto das imagens, uma vez que os algoritmos avaliados não fazem essa tarefa. Na avaliação do reconhecimento, tudo na imagem era considerado, reconhecendo assim não só a pessoa mas, o fundo da imagem no qual a pessoa está. Além disso, a melhoria da robustez da base de imagens, o aumento da quantidade de imagens e enriquecimento da variabilidade de iluminação, inclinação e expressões faciais é fundamental para outros trabalhos. Ao isolar o rosto das pessoas torna-se possível a combinação dos três bancos.

9 REFERÊNCIAS BIBLIOGRÁFICAS

[BELHUMEUR1997] BELHUMEUR, P. N., HESPANHA, J. P., KRIEGMAN, D. J. "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.

[BROOKS2004] BROOKS, A., "Face Recognition: Eigenface and Fisherface Performance Across Pose." Disponível em: <<http://pubweb.northwestern.edu/~acb206/ece432/FaceRecReport.html>>. Acessado em 20/06/2005.

[BURDEN2003] BURDEN, R. L., FAIRES, J. D., "Análise numérica." Editora Thomson. São Paulo: 2003

[CAMPOS2001] CAMPOS, T. E., "Técnicas de seleção de características com aplicações em reconhecimento de faces." *Dissertação de mestrado apresentada ao Instituto de Matemática e Estatística da Universidade de São Paulo*. Orientador: Dr. Roberto Marcondes Cesar Junior. São Paulo: 2001.

[CHELLAPPA1995] CHELLAPPA, R., WILSON, C. L., SIROHEY, S., "Human and machine recognition of faces: A survey." *Proc. IEEE*, vol. 83, pp. 705-740, 1995.

[JAIN2000] JAIN, A. K., DUIN, R. P. W., MAO, J. "Statistical pattern recognition: a review." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4-37.

[SPERANDIO2003] SPERANDIO, D., MENDES, J. T., SILVA, L. H. M., "Cálculo Numérico: características matemáticas e computacionais dos métodos numéricos". Editora Pearson. São Paulo: 2003.

[YAMBOR2000] YAMBOR, W. S., "Analysis of PCA-based and Fisher discriminant-based image recognition algorithms". *Computer Science Department Colorado State University. Technical Report CS-00-103*. Disponível em <http://www.cs.colostate.edu>. Acessado em 08/05/2005.

[LU2003] LU, Juwei; PLATANOTIS, Kostas N.; VENETSANOPOULOS, Anastasios N. "Face recognition using kernel direct discriminant analysis algorithms." In: *IEEE Transactions on Neural Networks*, v. 14. no 1. Toronto: janeiro, 2003. p. 117-26.

[SCHÖLKOPF1999] SCHÖLKOPF Bernhard; SMOLA, Alexander; MÜLLER Klaus-Robert. "Nonlinear component analysis as a kernel eigenvalue problem." *Neural Computation*, vol. 10, pp. 1299-1319, 1999.