

Reconhecimento Facial

Trabalho de
Implementação

Professor: Fernando dos Santos

Viviane Maria Wehrmeister



Ponto de Partida

- Linguagem Java
- JavaCV x OpenCV
- Reconhecimento Facial
- Treinamento
- Reconhecimento Facial Eigenfaces
- Avaliação do Algoritmo
- Aplicação

Reconhecimento Facial

- Identifica as faces.
- Se a imagem é do Goku ou Kuririn.
- Características.



Reconhecimento Facial

- Dataset Yale Faces
- Treinamento
- Teste

Dataset Yale faces

- 15 sujeitos



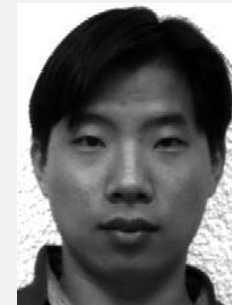
Sujeito 1



Sujeito 2



Sujeito 3



Sujeito 4



Sujeito 5



Sujeito 6



Sujeito 7



Sujeito 8



Sujeito 9



Sujeito 10



Sujeito 11



Sujeito 12



Sujeito 13



Sujeito 14



Sujeito 15

Dataset Yale faces

- 11 imagens de cada sujeito



Luz central



Óculos



Feliz



Luz esquerda



Sem óculos



Normal



Luz direita



Triste



Olhos Fechados



Surpreso



Piscada

Treinamento

- 8 imagens de cada sujeito, ou seja, 120 imagens no total para treinamento.
- 73% das imagens para treinamento.

Teste

- 3 imagens de cada sujeito, ou seja, 45 imagens no total para teste.
- 27% das imagens para testes.

■ Escolha das imagens para teste

- Sujeitos=[Sujeito1, ... ,Sujeito 15].
- Condição=[a)Luz Central, b)Óculos, c)Feliz, d)Luz Esquerda, e)Sem óculos, f)Normal, g)Luz a direita, h)Triste, i)Olhos Fechados, j)Surpreso, k)Piscada]

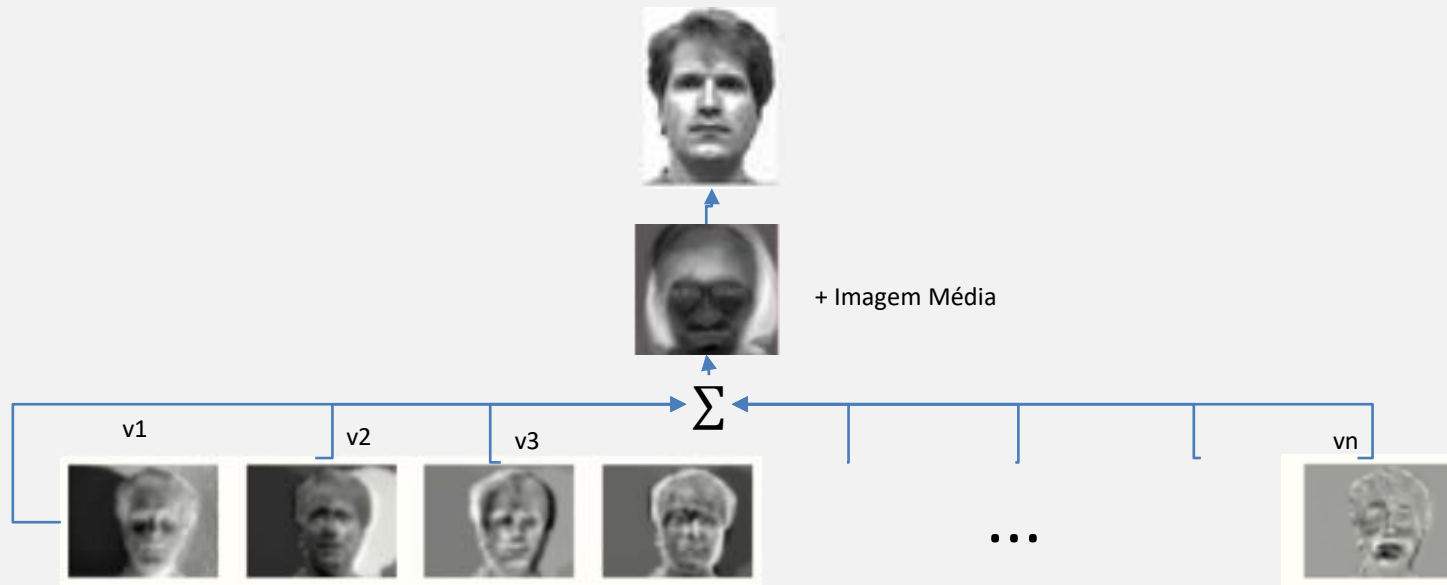
- | | | |
|-----------------------|------------------------|------------------------|
| • Sujeito 1: a, b, c. | • Sujeito 6: e, f, g. | • Sujeito 6: i, j, k. |
| • Sujeito 2: d, e, f. | • Sujeito 7: h, i, j. | • Sujeito 7: a, e, i. |
| • Sujeito 3: g, h, i. | • Sujeito 8: k, a, b. | • Sujeito 8: b, f, j. |
| • Sujeito 4: j, k, a. | • Sujeito 9: c, d, e. | • Sujeito 9: c, g, k. |
| • Sujeito 5: b, c, d. | • Sujeito 10: f, g, h. | • Sujeito 10: d, h, a. |

■ PCA Principal Component Analysis

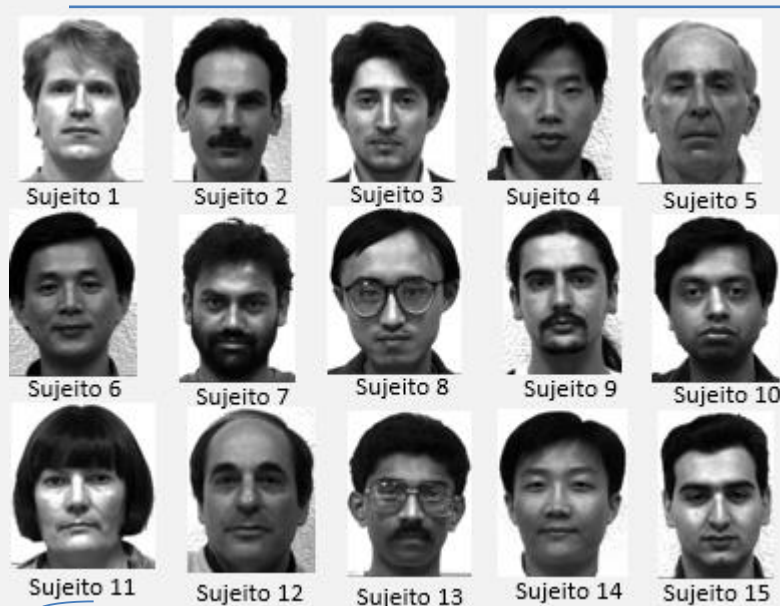
- A abordagem Eigenfaces é um método PCA, no qual um pequeno conjunto de características das imagens é usado para descrever a variação entre as imagens de rosto.
- O objetivo é encontrar os autovetores (eigenfaces) da matriz de covariância da distribuição, abrangidos pelo treinamento de um conjunto de imagens faciais.
- Posteriormente, cada imagem facial é representada por uma combinação linear desses autovetores.

Eigenfaces

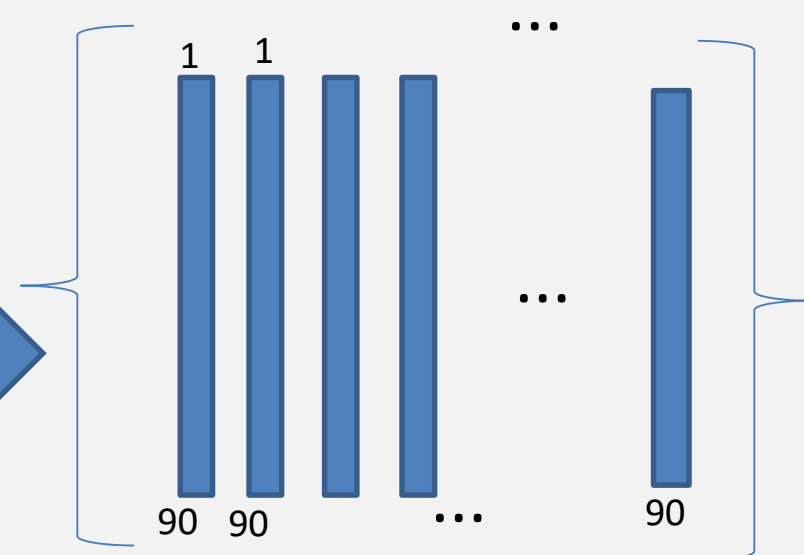
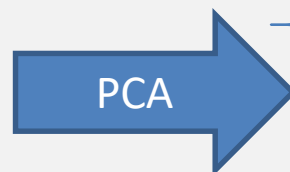
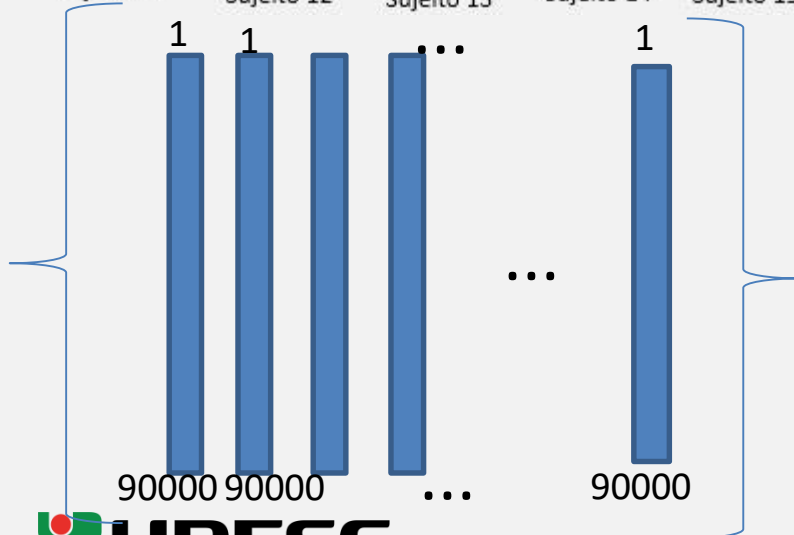
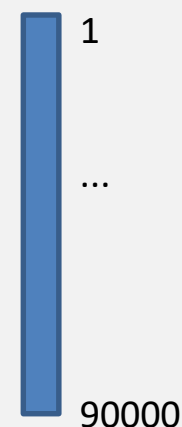
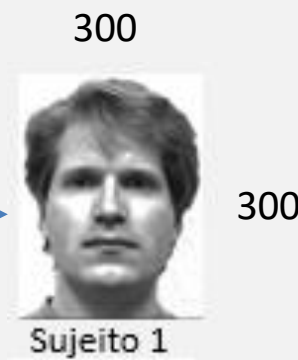
- EigenFaces é o nome para autovetores que são componentes da própria face
- Luminosidade.



PCA Componentes



Dimensões sejam $300 \times 300 = 90.000$ pixel



Eingenfaces – Treinamento

- Representar cada imagem do dataset-treino como um vetor

<https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses>

$$I_i = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}_{N \times N} \xrightarrow{\text{concatenation}} \begin{bmatrix} a_{11} \\ \vdots \\ a_{1N} \\ \vdots \\ a_{2N} \\ \vdots \\ a_{NN} \end{bmatrix}_{N^2 \times 1} = \Gamma_i$$

ImagemA=	1	3	ImagemB=	3	3
	0	2		1	7

Imagem A=	1	Imagem B=	3
	3		3
	0		1
	2		7

Eingenfaces – Treinamento

- Calcular a face média

<https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses>

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

Face media=	2
	3
	0,5
	4,5

- Subtrair cada face Γ_i do vetor médio

<https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses>

$$\Phi_i = \Gamma_i - \Psi$$

Imagem A=	-1	Imagem B=	1
	0		0
	-0,5		0,5
	-2,5		2,5

Eingenfaces – Treinamento

- Calcular a matriz de covariância

$$C = AA^T \quad A = [\Phi_1, \Phi_2 \dots \Phi_M]$$

<https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses>

A=	<table><tr><td>-1</td><td>1</td></tr><tr><td>0</td><td>0</td></tr><tr><td>-0,5</td><td>0,5</td></tr><tr><td>-2,5</td><td>2,5</td></tr></table>	-1	1	0	0	-0,5	0,5	-2,5	2,5	AT=	<table><tr><td>-1</td><td>0</td><td>-0,5</td><td>-2,5</td></tr><tr><td>1</td><td>0</td><td>0,5</td><td>2,5</td></tr></table>	-1	0	-0,5	-2,5	1	0	0,5	2,5
-1	1																		
0	0																		
-0,5	0,5																		
-2,5	2,5																		
-1	0	-0,5	-2,5																
1	0	0,5	2,5																
C=	<table><tr><td>2</td><td>0</td><td>5</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>2,5</td></tr><tr><td>5</td><td>0</td><td>12,5</td></tr></table>	2	0	5	0	0	0	1	0	2,5	5	0	12,5						
2	0	5																	
0	0	0																	
1	0	2,5																	
5	0	12,5																	

Eigenfaces – Treinamento

- Escolha os k autovetores mais importantes.

$$\text{Autovalores} = \det(A - \text{Lambda} * I)$$

$$A - \text{Lambda} * I = \begin{vmatrix} 2 - \text{lambda} & 0 & 5 \\ 0 & 0 & 0 \\ 1 & 0 & 2,5 \\ 5 & 0 & 12,5 - \text{lambda} \end{vmatrix}$$

$$\lambda_1 = 15$$

$$\lambda_2 = 0$$

$$\lambda_3 = 0$$

$$\lambda_4 = 0$$

$$AX = \text{Lambda}X$$

$$\begin{vmatrix} 2 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0,5 \\ 5 & 0 & 2,5 \end{vmatrix} * \begin{vmatrix} a \\ b \\ c \\ d \end{vmatrix} = \begin{vmatrix} \text{Lambda} * a \\ \text{Lambda} * b \\ \text{Lambda} * c \\ \text{Lambda} * d \end{vmatrix}$$

$$v_1 = \left(\frac{2}{5}, 0, \frac{1}{5}, 1 \right)$$

$$v_2 = \left(-\frac{5}{2}, 0, 0, 1 \right)$$

$$v_3 = \left(-\frac{1}{2}, 0, 1, 0 \right)$$

$$v_4 = (0, 1, 0, 0)$$

Eingenfaces – Treinamento

- Quando os eigenfaces encontrados são transformados em $n \times n$ geram uma imagem com rosto de fantasma



Eigenfaces – Treinamento

- Encontramos as k eigenfaces (autovetores) que aproximam as M imagens de faces usadas no treinamento.

$$v_1 = \left(\frac{2}{5}, 0, \frac{1}{5}, 1 \right)$$

$$v_3 = \left(-\frac{1}{2}, 0, 1, 0 \right)$$

$$v_2 = \left(-\frac{5}{2}, 0, 0, 1 \right)$$

$$v_4 = (0, 1, 0, 0)$$

$$\text{proj}(\Phi_i, E_j) = \frac{\Phi_i \cdot E_j}{|E_j|} = \Phi_i \cdot E_j, 1 \leq i \leq M, 1 \leq j \leq 9$$

<https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses>

Imagem A=	-1	Imagem B=	1
	0		0
	-0,5		0,5
	-2,5		2,5

$$\begin{pmatrix} \frac{2}{5} & 0 & \frac{1}{5} & 1 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ 0 \\ -0.5 \\ -2.5 \end{pmatrix} = (-3) \quad \begin{pmatrix} \frac{2}{5} & 0 & \frac{1}{5} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 0.5 \\ 2.5 \end{pmatrix} = (3)$$

Eingefaces - Treinamento

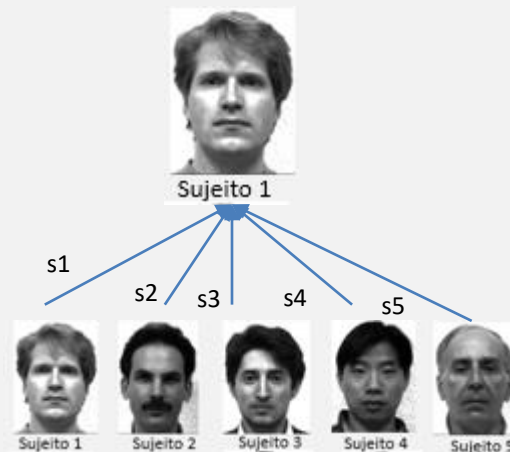
Ver no Netbeans

■ Eigenfaces - Teste

- Projeta a imagem de teste no espaço de faces
- Extrai componentes eigenfaces da imagem
- Calcula distancia entre a nova face e as face de treinamento.
- Busca nas imagem de treino pela mais semelhante.

Eingenfaces - Teste - Threshold

- Limite de confiança/ distância (vizinho mais próximo)
- Faces com valor de threshold a maior que o especificado não são consideradas (retorna -1)
- Margem de erro que indica se a face pertence a alguma classe da base de treinamento
- Como definir o valor?



Eingefaces - Teste

Ver no Netbeans

Eingenfaces - Teste

		REAL														
P R E D I T I V O		S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
	S1	3	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	S2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
	S3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
	S4	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
	S5	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0
	S6	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
	S7	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
	S8	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0
	S9	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0
	S10	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
	S11	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0
	S12	0	0	0	0	1	0	0	0	1	0	0	2	0	0	0
	S13	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0
	S14	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0
	S15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Matriz de confusão do teste prático, Dataset Yale faces.

Eingenfaces - Teste

Acurária= 0,84

Precision		Recall	
Sujeito 1	0,60	Sujeito 1	1,00
Sujeito 2	1,00	Sujeito 2	0,67
Sujeito 3	1,00	Sujeito 3	0,67
Sujeito 4	1,00	Sujeito 4	1,00
Sujeito 5	1,00	Sujeito 5	0,67
Sujeito 6	1,00	Sujeito 6	1,00
Sujeito 7	1,00	Sujeito 7	1,00
Sujeito 8	0,75	Sujeito 8	1,00
Sujeito 9	0,67	Sujeito 9	0,67
Sujeito 10	1,00	Sujeito 10	0,67
Sujeito 11	1,00	Sujeito 11	1,00
Sujeito 12	0,50	Sujeito 12	0,67
Sujeito 13	0,75	Sujeito 13	0,67
Sujeito 14	1,00	Sujeito 14	0,67
Sujeito 15	1,00	Sujeito 15	1,00

Erro	
Luz Central	20,00%
Óculos	0,00%
Feliz	0,00%
Luz a esquerda	100,00%
Sem óculos	0,00%
Normal	0,00%
Luz a direita	75,00%
Triste	0,00%
Olhos Fechados	0,00%
Surpreso	0,00%
Piscada	0,00%

Eingenfaces - Aplicação

Ver no Netbeans

Bibliografia

- Bugra, PCA, EigenFace and All That, Disponível em: <https://bugra.github.io/posts/2013/7/27/PCA-EigenFace-And-All-That/> Acesso em: 11/09/2020.
- Chen, J.M., Eigenfaces for Dummies. Disponível em: <http://jmcspot.com/Eigenface/> Acesso em : 11/09/2020.
- FingerTec, Face Recognition, Disponível em: <https://www.fingertec.com/whatsnew/face-recognition-white-paper-technology-2>. Acesso em: 11/09/2020.
- GeeksforGeeks, Face Recognition Using Eigenfaces(PCA Algorithm), Disponível em: <https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/> Acesso em : 11/09/2020.
- Huang, K., Principal Component Analysis in the Eigenface Technique for Facial Recognition, Disponível em: <https://digitalrepository.trincoll.edu/cgi/viewcontent.cgi?article=1221&context=theses> Acesso em: 11/09/2020.
- OpenCV,Face Recognizer, Disponível em: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html?highlight=eigenface Acesso em: 11/09/2020.
- Santos, E. F, et al. Aplicativo de foto chamada. Disponível em: <https://www.ic.unicamp.br/~rocha/teaching/2011s1/mc906/trabalhos/tp/tp-gr-01.pdf>. Acesso em: 11/09/2020.



Grata pela atenção!

**UDESC – Universidade do Estado de
Santa Catarina**

**CEAVI – Centro de Educação
Superior do Alto Vale do Itajaí**

**Graduandos do curso de Engenharia
de Software**

Viviane.mw@edu.udesc.br