

Reconhecimento facial

Jones Granatyr



**Visão computacional simula a
capacidade do olho humano**

Detecção x Reconhecimento



Aplicações

▶ Detecção

- Contar quantas pessoas estão no ambiente
- Alarmes
- Controle de tráfego em rodovias
- Obter o tamanho de um objeto
- Detecção de sorrisos em câmeras / ajuste de foco

▶ Reconhecimento

- Desbloqueio do celular
- Sistemas de segurança
- Validação em cursos on-line
- Identificar bandidos

Laranja camisa
Azul calção
Azul sapato

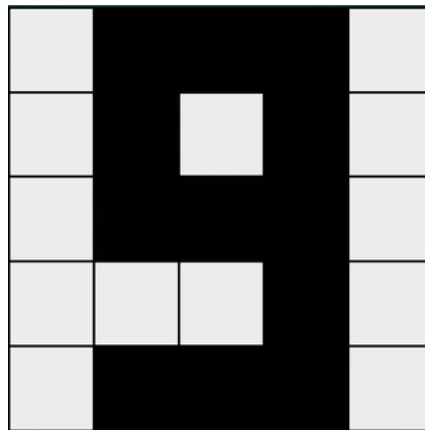


8.97,3.45,2.35,0.0,00.00,0.00,Bart
6.75,0.94,0.52,0.00,0.00,0.00,Bart
9.69,4.10,1.56,0.00,0.00,0.00,Bart
0.00,0.00,0.00,4.68,0.66,0.01,Homer
0.00,0.00,0.00,0.12,2.50,0.03,Homer
0.00,0.00,0.00,5.80,0.50,1.28,Homer

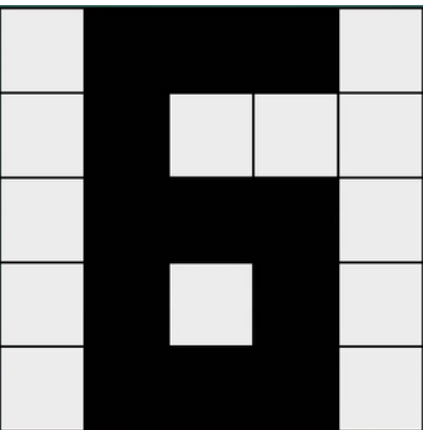
Marrom boca
Azul calca
Cinza sapato



1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0
1	2	3	4	5	6	7	8	9	0



0,1,1,1,0,0,1,0,1,0,0,1,1,1,0,0,0,0,1,0,0,1,1,1,0,9

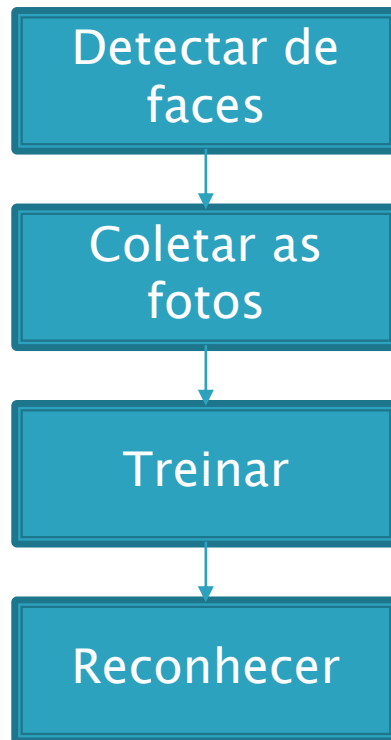


0,1,1,1,0,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,0,1,1,1,0,6

Algoritmos

- ▶ Eigenfaces
- ▶ Fisherfaces
- ▶ LBPH (Local Binary Patterns Histograms)


Etapas para o reconhecimento



Captura das fotos de treinamento

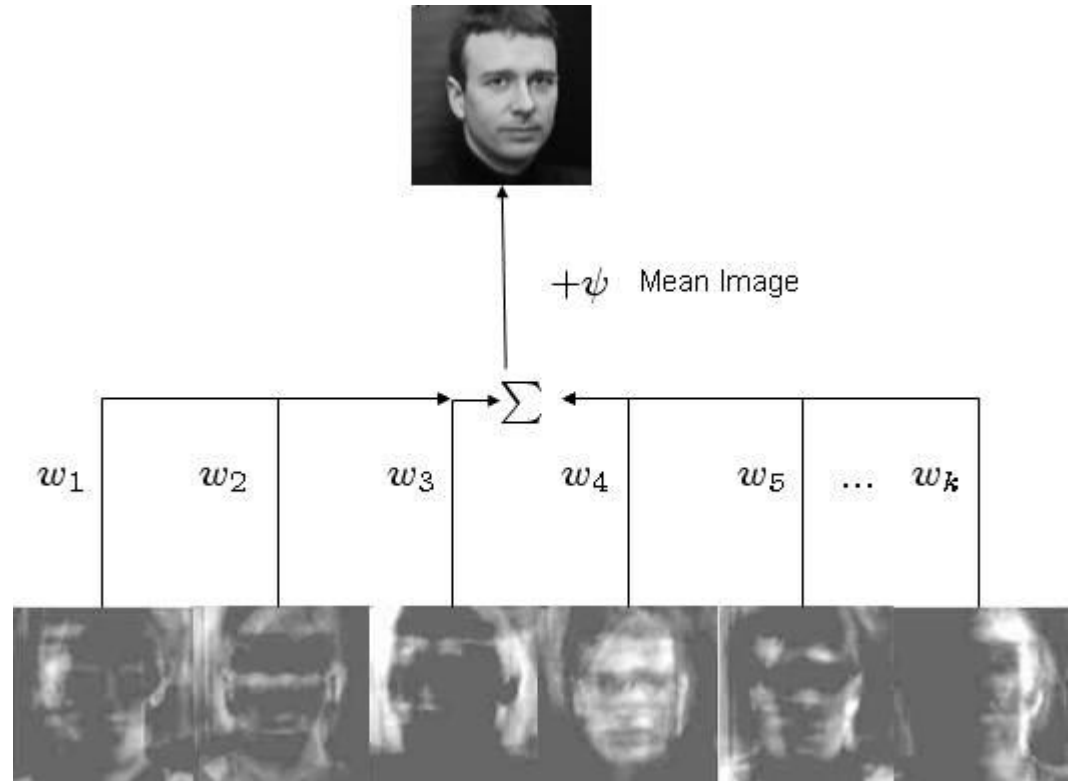
- ▶ Capturar as imagens da pessoa pela webcam e atribuir um identificador (id)
- ▶ Formato
 - pessoa.{id}.{numerofoto}.jpg
 - pessoa.1.1.jpg
 - pessoa.1.2.jpg
 - pessoa.2.1.jpg
 - pessoa.2.2.jpg
- ▶ Tamanho padrão das imagens (220 x 220)

Recomendações captura das fotos

- ▶ As imagens para o treinamento são fundamentais para um reconhecimento eficiente
 - ▶ Fazer um ensaio antes de tirar as fotos
 - ▶ Ambiente bem iluminado
 - ▶ Variações na expressão (feliz, triste, com e sem óculos)
 - ▶ Ângulo (olhando levemente para cima, baixo, esquerda, direita)
 - ▶ Luz incidindo no rosto
- 

Eigenfaces

Eigenfaces



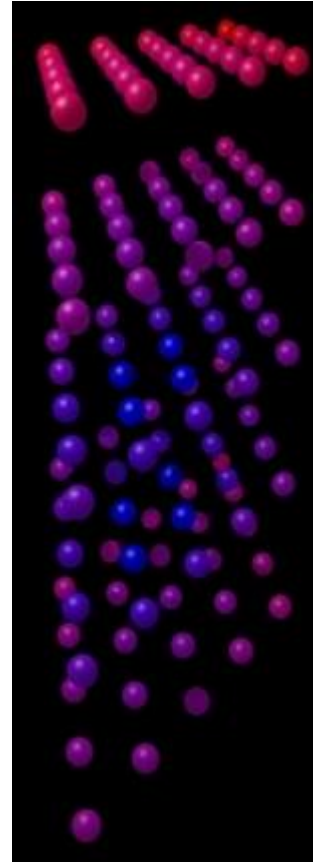
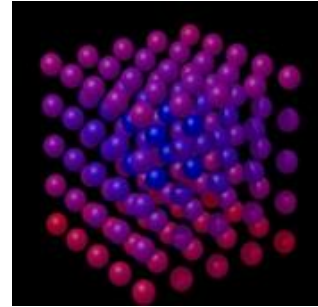
Eigenvector – “vetor próprio”

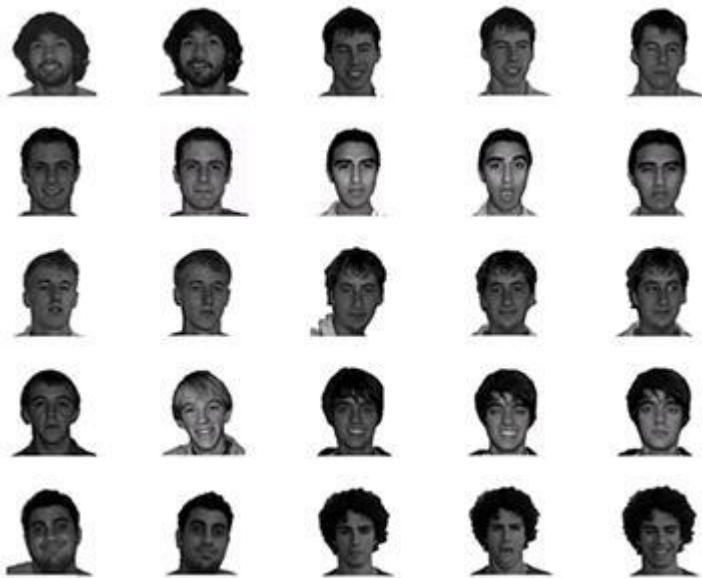
$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ -3 \end{bmatrix}$$

Eigen vector

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & -3 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 0 \end{bmatrix} = 3 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

Eigen value

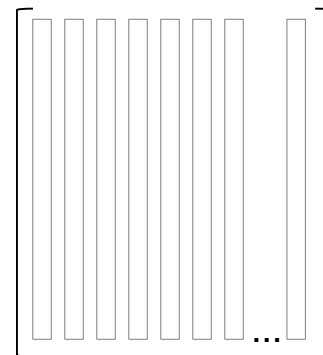




220
220
48.400 pixels



Cada face é uma coluna

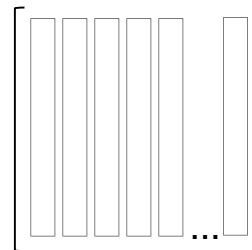


PCA (Principal Component Analysis)

Quanto as pessoas variam da média
Maior variação no cabelo
Mostra os desvios (variações)



Eigen vector/Eigen faces



Remover informação inútil
Reduzir dimensionalidade

Eigenfaces – treinamento

- ▶ Imagem $256 \times 256 = 65.536$ pixels que é um espaço muito grande
- ▶ Um conjunto de imagens pode mapear um grupo de pontos neste grande espaço
- ▶ As faces são decompostas numa série de componentes principais
- ▶ Como as faces são parecidas (olhos, boca, nariz), os pontos principais não serão distribuídos aleatoriamente (focar em áreas de mudanças)
- ▶ Cada face pode ser representada como a combinação linear de diversas eigenfaces
- ▶ A variação ajuda a diferenciar uma face da outra
- ▶ Considera a iluminação como uma característica importante
- ▶ Encontrar vetores que melhor representam a distribuição das faces
- ▶ Eigenvector projeta as imagens de faces no espaço de faces

Eigenfaces – classificação

- ▶ Projeta a imagem de teste no espaço de faces
- ▶ Extrai componentes eigenfaces da imagem
- ▶ Calcula a distância entre a nova face e as faces de treinamento (knn)
- ▶ Busca nas imagens de treinamento pela mais semelhante

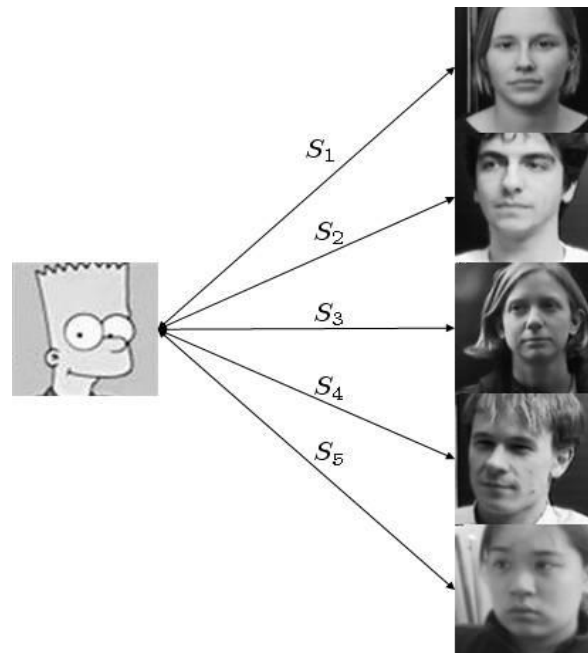
Parâmetros Eigenfaces – num_components



- ▶ Número de componentes principais (eigenfaces)
- ▶ Não há uma regra de quantos componentes devem ser usados (a documentação indica que 50 é quase sempre suficiente)
- ▶ Exemplo
 - 100 faces geraram 50 componentes
 - A face é recodificada como 50 valores de componentes (assinatura do rosto)


Parâmetros Eigenfaces – Threshold

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



Fisherfaces

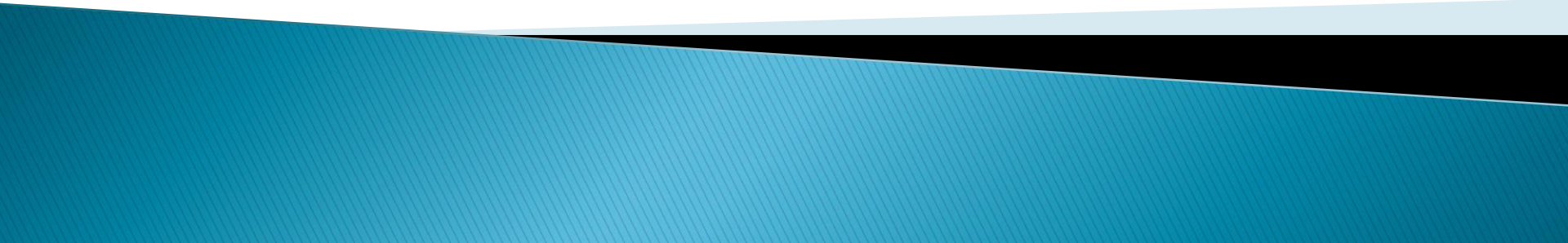
Fisherfaces

- ▶ Eigenfaces olha para todas as imagens de todas as pessoas de uma vez e tenta encontrar componentes de todas elas combinadas
 - ▶ Não foca nas características que distinguem um indivíduo do outro (faces da pessoa como um todo)
 - ▶ PCA foca na variação
 - ▶ A iluminação é um componente importante e pode descartar outras características
- 

Fisherfaces

- ▶ Utiliza LDA (Linear Discriminant Analysis), que também reduz as dimensões
- ▶ LDA não está interessado na maior variação, mas em maximizar a separação entre as classes
- ▶ Extração das características separadamente (a iluminação de uma face não afetará as outras faces)
- ▶ Fisherfaces podem ser calculados usando PCA ou regularização
- ▶ Parâmetros OpenCV
 - num_components
 - threshold

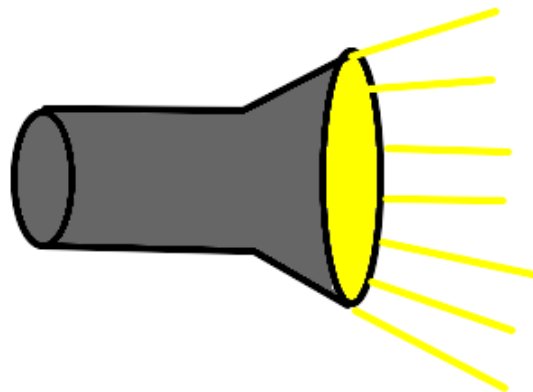
LBPH (Local Binary Patterns Histograms)



LBPH

12	15	18
5	8	3
8	1	2

Se ≥ 8 então 1
Se < 8 então 0

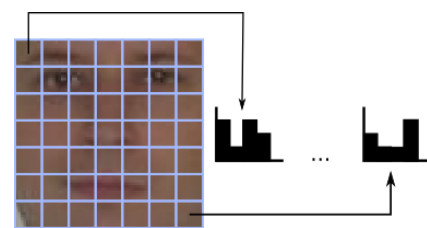
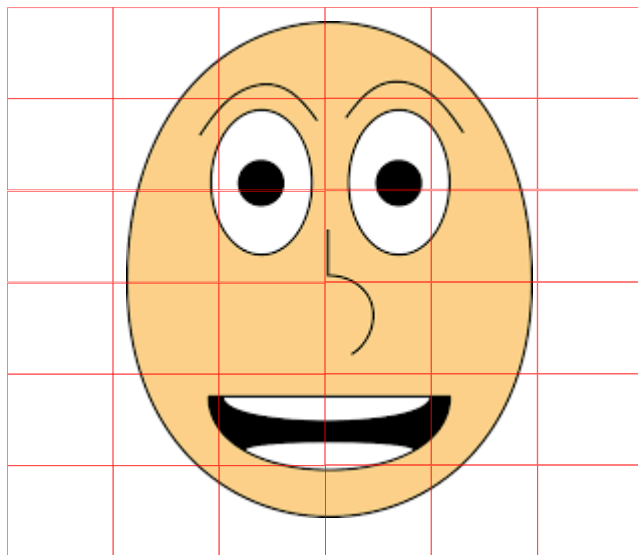
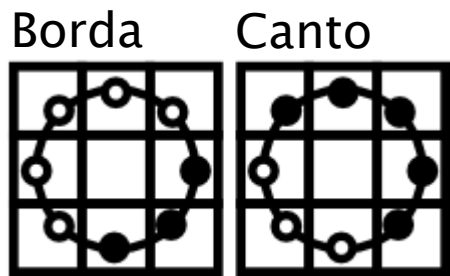


42	55	48
35	38	33
38	30	32

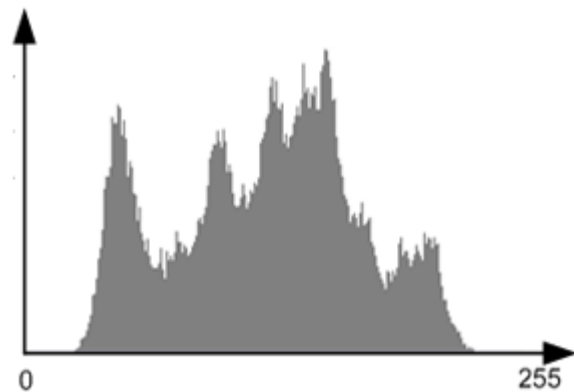
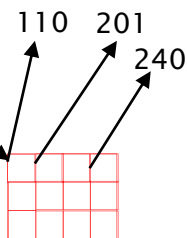
1	1	1
0	8	0
1	0	0

Binário = 11100010


Decimal = 226



Fonte: https://bytefish.de/blog/local_binary_patterns/

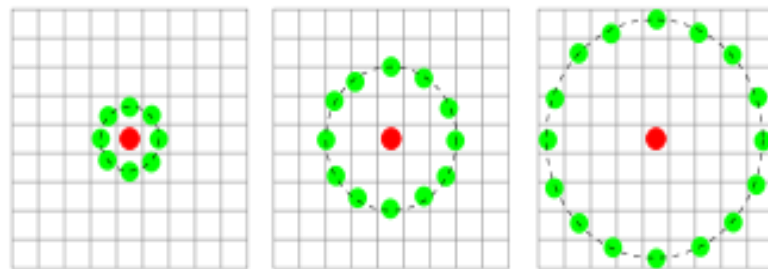


LBPH – Treinamento e classificação

- ▶ O número decimal é usado para treinar o sistema, gerando um histograma dos valores (um histograma para cada face)
 - ▶ Encontrar a estrutura local da imagem por meio dos vizinhos
 - ▶ Gera o histograma para a nova imagem e compara com os histogramas da base de dados
- 

Parâmetros – radius

- ▶ Raio maior aumenta a abrangência mas pode perder bordas finas (pontos mais distantes)
- ▶ Quanto maior o raio mais padrões podem ser codificados, mas aumenta o esforço computacional



Fonte: https://en.wikipedia.org/wiki/Local_binary_patterns

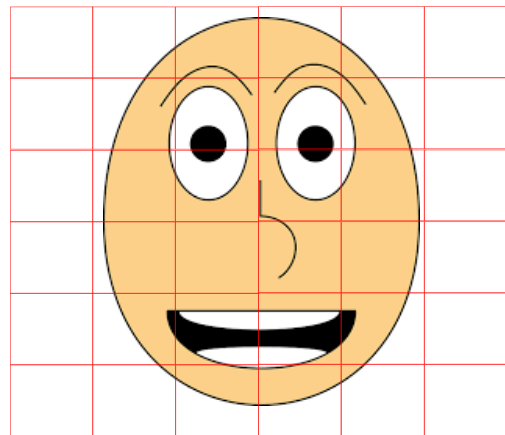
Parâmetros – neighbors

- ▶ Número de pontos da amostra para construir um padrão local
- ▶ Quanto maior o número de vizinhos maior é o esforço computacional

12	15	18
5	8	3
8	1	2

Outros parâmetros

- ▶ **grid_x**
 - Número de células na horizontal
 - Quanto mais células maior é a dimensionalidade do vetor de características (histogramas)
- ▶ **grid_y**
 - Número de células na vertical
 - Se a grade aumentar serão usados menos pixels em cada histograma (mais espaços)
- ▶ **threshold**
 - Limite de confiança



Conclusão