Face Recognition
Reconhecimento Facial

Mauricio Silva do Rego

RM: 333710

Resumo

Este trabalho tem por finalidade aprofundar o conhecimento nos classificadores de faces presentes na biblioteca OpenCV: Eingenfaces, Fisherfaces e Local Binary Patterns Histograms.

Eigenfaces

Um dos algoritmos mais antigos de reconhecimento facial, sofisticando o algoritmo PCA na qual reduz a matriz de covariância reduzindo o processamento para cálculo do auto-vetores e auto-valores, que tem por finalidade extrair eigenvectors (auto-vetores), com objetivo de gerar a face média e a partir das eigenfaces e gerar inúmeras faces similares ao database usado. A ideia é usar características principais como: olhos, nariz, orelhas e boca.

Embora atualmente não tão eficiente, é frequentemente usado para demostrar o desempenho mínimo esperado de um sistema com esse propósito.

Funcionamento:

É extraído os autofaces da imagem (PCA):

1. "Dado M imagens de rosto com o tamanho de h × w , cada imagem é transformada em um vetor de tamanho D (= h w) e colocado no conjunto"

   $$\{\Gamma_1, \Gamma_2, \cdots, \Gamma_M\}.$$

   "As imagens do rosto devem ser adequadamente dimensionadas e alinhadas, e os fundos (e possivelmente áreas que não são do rosto, como cabelo e pescoço) devem ser constantes ou removidos."

2. "Cada face difere da média pelo vetor $\Phi_{Eu} = \Gamma_{Eu} \Psi$, onde a face média é definida por $\Psi = \frac{1}{M} \sum_{i=1}^{M} \Gamma_{Eu}$ "

3. "A matriz de covariância $\mathbf{C} \in \mathbb{R}^{D \times D}$ é definido como"

   $$\mathbf{C} = \frac{1}{M} \sum_{i=1}^{M} \Phi_{Eu} \Phi_{Eu}^{\top} = \mathbf{A}\,\mathbf{A}^{\top},$$

Onde $\mathbf{A} = \{\Phi_1, \Phi_2, \cdots, \Phi_M\} \in \mathbb{R}^{D \times M}$.

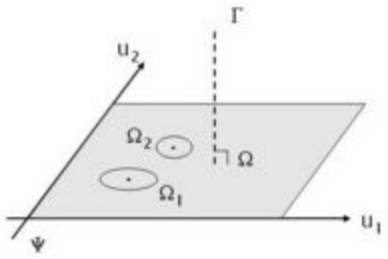
4. “Determinando os vetores próprios de C é uma tarefa intratável para tamanhos de imagem típicos quando D ≫ M . No entanto, para calcular com eficiência os vetores próprios de C , pode-se primeiro calcular os autovetores dos muito menores M× M matriz UMATUm . As matrizes de vetor próprio e valor próprio de UMATUMA são definidos como”

   $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_r\}$

   e $\Lambda = di\ um\ g\{\lambda_1, \lambda_2, \cdots \lambda_r\}, \lambda_2 \geq \cdots X_r >$, Onde r é o posto de Um . Observe que os vetores próprios correspondentes aos valores próprios de zero foram descartados.

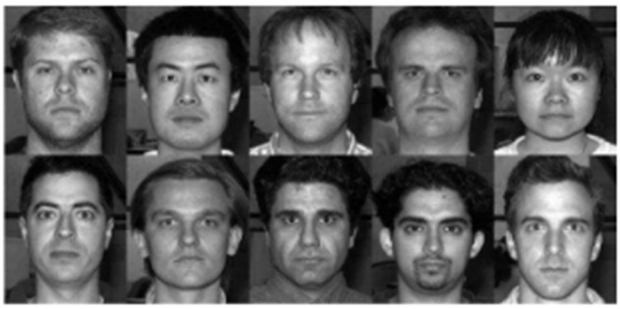
5. As matrizes de autovalor e autovetor de C estamos Λ e U = A V Λ- 1 / 2 , Onde U ={uEu} é a coleção de autofaces.

Os autofaces abrangem um msub-dimensional do espaço da imagem original escolhendo o subconjunto de vetores próprios vocêˆ= { u1 1, ⋯ , um} associado com o mmaiores autovalores. Isso resulta no chamado espaço de face , cuja origem é a face média e cujos eixos são os autofaces. Para realizar a detecção ou reconhecimento de rosto, é possível calcular a distância dentro ou a partir do espaço do rosto.



Visualização de um espaço de face 2D, com os eixos representando dois Eigenfaces.

Imagens originais (linha 1) e suas projeções no espaço de face (linha 2).



Pontos positivos:

- Processo de treinamento automático e fácil de codificar
- Após calculadas as autofaces de um database o reconhecimento facial pode ser alcançado em tempo real
- Lidar com grandes database

Pontos negativos:

- Sensível a variação de iluminação, escala, pose, expressão facial e oclusao.
- Para um funcionamento aceitável a imagem deverá estar vista frontal, iluminação semente, expressão definida (sem variação)

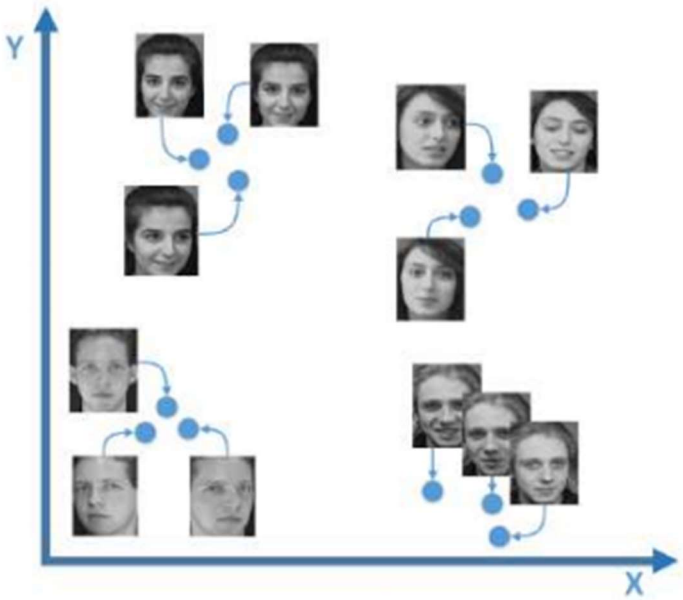
Treinamento: calcula as Eigenfaces, projetar as faces conhecidas da base de dados de imagem, no espaço de faces;

Reconhecimento: A identificação de uma face é realizada pela sua projeção no subespaço criado pelas Eigenfaces e em seguida, pela comparação da posição obtida com a posição de faces conhecidas;

Firsherfaces

Inicialmente criado com objetivo de reconhecimento de fala, posteriormente aplicado em reconhecimento de facial, visando melhoria na acurácia do Eigenfaces. É baseado no algoritmo LDA e foi desenvolvido com a ideia de ser um método não sensível a grande variação de iluminação e expressão facial.

Extraí as características principais de uma pessoa, caracterizando as variações de aparência presente nas imagens, como pose, iluminação, expressão facial. Então nota-se que quanto mais diferente uma pessoa for da outra, mais distante deverão estar projetadas



Etapas:

- Calculo da imagem média
- Cálculo da face média geral
- Transformação das imagens em vetores
- Construção da Matriz de dispersão intra-classes
- Construção da Matriz de dispersão inter-classes
- Cálculo das fisherfaces
- Cálculo dos vetores de características

- Cálculo da similaridade

Após a etapa de detecção das faces e extração das características, as imagens são projetadas neste espaço, gerando um vetor de características que representa a face média de uma pessoa.

Pontos positivos:

- Expressão facial
- Iluminação
- Pose

Pontos negativos:

- Alto custo computacional
- Alta dimensionalidade

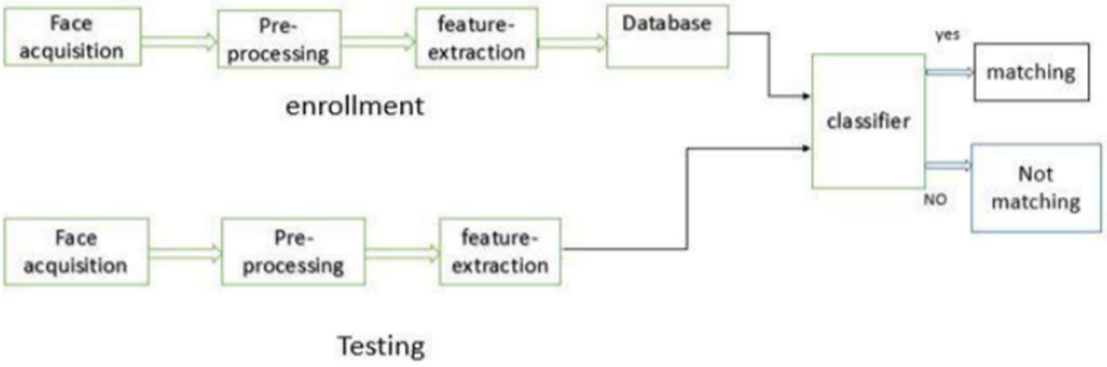
Local Binary Patterms Histograms

Inicialmente desenvolvido com intuito de análise de texturas, o LBP são descritores locais de texturas baseados na suposição de que a informação de uma textura é dividida em dois aspectos complementares padrão e intensidade. Rotula uma vizinhança 3x3 em torno de cada pixel, comparando com o pixel central atribuindo 1 caso ele maior que o pixel central e 0 caso contrário. Cada pixel é calculado com 8 bits, e ao final é criado um histograma dos códigos LBP calculados e este representa a textura presente na imagem.

Etapas:

- Criação de dataset

- Extração daface
- Extração das características
- Classificação

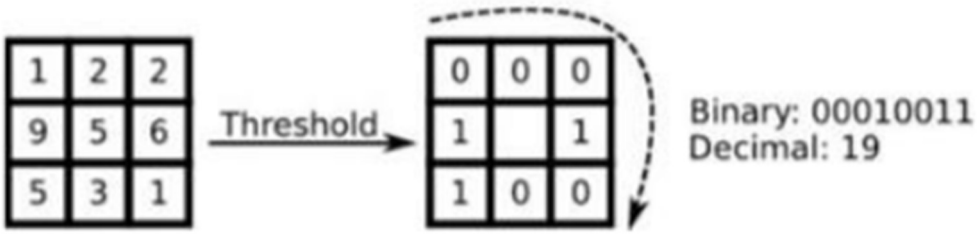


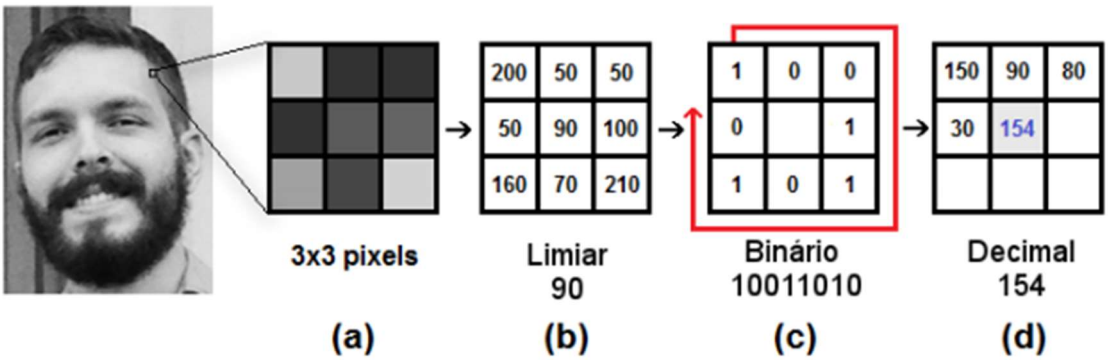
Divide a imagem em regiões de mesma altura e largura, aplica-se o operador local em todas as regiões, é definido na janela por 3x3.

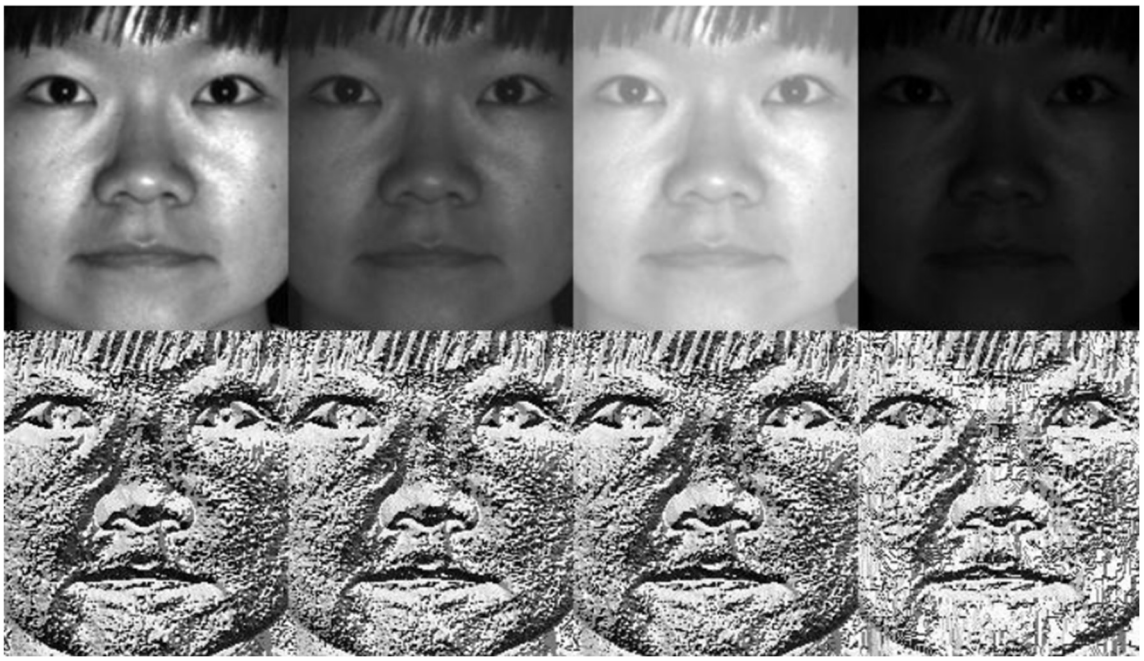
$$\mathrm{LBP}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(i_p - i_c)$$

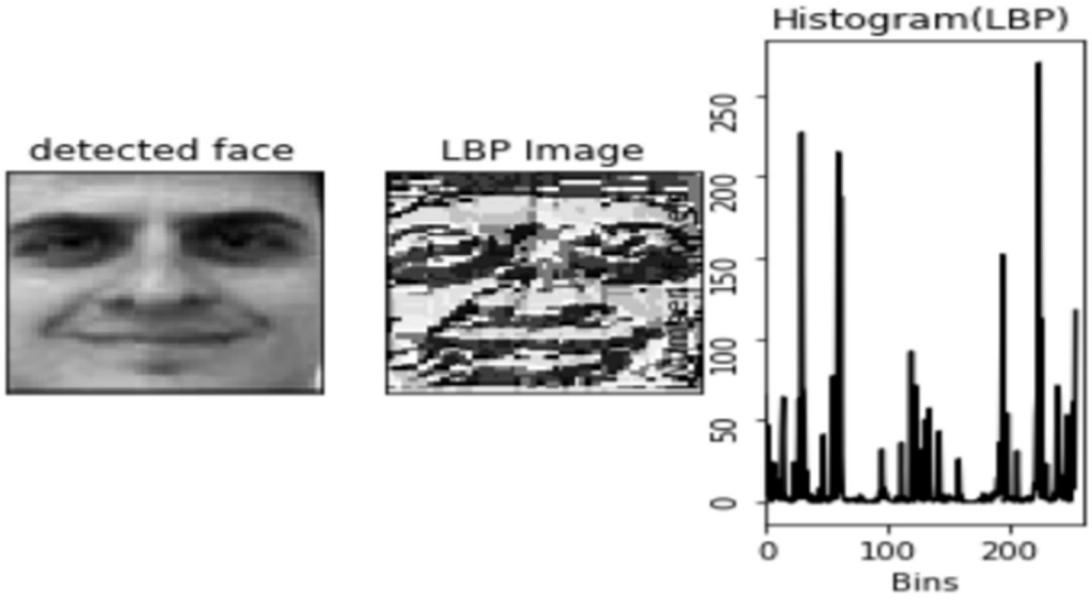
Compara o valor mediano com os 8 vizinhos mais próximos.

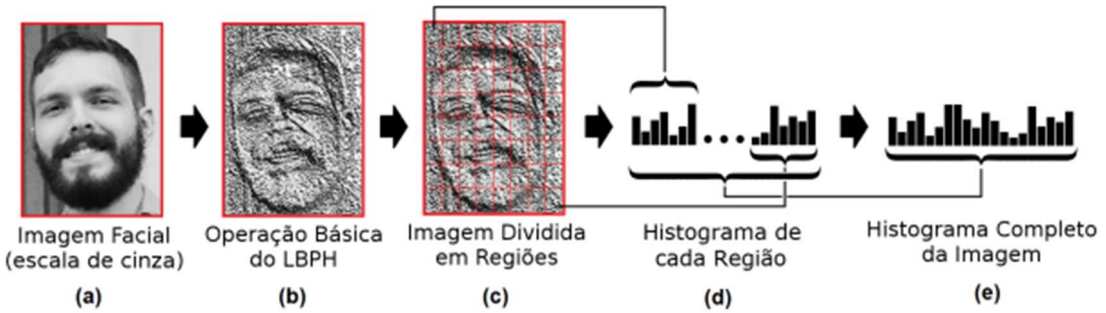
$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$




200 50 50
50 90 100
160 70 210
1 0 0
0 1
1 0 1
150 90 80
30 154
3x3 pixels
Limiar
90
Binário
10011010
Decimal
154
(a)
(b)
(c)
(d)







Pontos positivos:

- Descritores de texto com maior desempenho
- Robusto com transformações monotônicas da escala de cinza
- Excelente resultado independente do ambiente e condições de luz
- Reconhecimento de faces frontais e laterais
- Fácil de implementar

Pontos negativos:

- Cada imagem é analisada separadamente

Conclusão

Triste por não ter conseguido rodar o código fonte, e mesmo pesquisado em diversos fórum “module cv2 cv2 has no attribute” nenhum conseguiu ajudar resolver meu erro de compilação. De qualquer forma, com a literatura foi possível ter um vasto aprendizado sobre essas as técnicas acima descritas, cada uma com sua aplicação. Ficará como desafio pessoal, descobrir (nem que tenha que desinstalar o Jupter) o motivo pelo qual não foi possível executar o código fonte.

Nota: em várias buscas, percebi a vasta dificuldade dos Dvs para resolver esse erro de compilação.

Referências

https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html
https://github.com/kelvins/Reconhecimento-Facial/wiki/FaceRecognition
https://en.wikipedia.org/wiki/Eigenface
http://www.scholarpedia.org/article/Eigenfaces
http://disp.ee.ntu.edu.tw/~pujols/Eigenfaces%20and%20Fisherfaces.pdf
http://www.scholarpedia.org/article/Fisherfaces
https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b
https://iq.opengenus.org/lbph-algorithm-for-face-recognition/
https://iq.opengenus.org/lbph-algorithm-for-face-recognition/
https://www.univates.br/bdu/bitstream/10737/2328/1/2018LuizHenriquedeOliveiraGalimberti.pdf
https://www.teses.usp.br/teses/disponiveis/100/100131/tde-07012018-222531/publico/Corrigida_Kelvin_Salton.pdf