

Análise e comparativo de métodos de reconhecimento facial a partir de técnicas da visão computacional

Ewerton Carlos de Araujo Assis

Resumo—O presente trabalho tem por finalidade estabelecer um comparativo entre três técnicas para realizar reconhecimento facial em sistemas computacionais, através da Inteligência Computacional e Visão Computacional: Eigenfaces, Fisherfaces e Local Binary Patterns Histograms. Cada técnica é apresentada com suas motivações, funcionamento e pontos positivos e negativos. Um exemplo de cada técnica está disponível através de um Jupyter Notebook.

Palavras-Chave—Reconhecimento de face; visão computacional; inteligência computacional; aprendizado de máquina

Abstract—The present work aims to establish a comparative analysis of three techniques to work with face recognition in computer systems, through computational intelligence and computer vision methods: Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms. Each technique is presented with its motivation, how it works, pros, and cons. An example with each technique is available through a Jupyter Notebook.

Keywords—Face recognition; computer vision; computational intelligence; machine learning

I. INTRODUÇÃO E MOTIVAÇÃO

Diferentes abordagens para trabalhar com reconhecimento facial estão disponíveis a partir de métodos desenvolvidos em visão computacional. O presente trabalho tem por finalidade realizar um comparativo entre os três principais métodos utilizados e disponibilizados na biblioteca OpenCV: *Eigenfaces*, *Fisherfaces* e *Local Binary Patterns Histograms*.

As análises são feitas tendo por finalidade evidenciar a motivação, o funcionamento e pontos positivos e negativos de cada abordagem. Este material acompanha um *Jupyter Notebook* com exemplo prático de cada abordagem.

II. TÉCNICAS

A. Eigenfaces

O *Eigenfaces* — uma das primeiras técnicas de reconhecimento facial — tem por motivação principal realizar o reconhecimento facial a partir de uma abordagem que analisa detalhes da face como um todo, ao invés de focar em regiões da face ou características específicas. Esta abordagem, portanto, realiza a “extração de informações relevantes da face”, a partir de “variações estatísticas entre imagens faciais”. Esta extração é feita de forma a ser eficiente, em termos de consumo de espaço e representação das características (*features*) relevantes [1].

Ewerton Carlos de Araujo Assis, RM 330737, FIAP, MBA em Inteligência Artificial e Aprendizado de Máquina. São Paulo-SP. E-mail pessoal: earaujoassis@gmail.com.

Esta técnica está fortemente baseada no conceito de Análise das Componentes Principais (PCA — *Principal Component Analysis*), que tem por finalidade extrair justamente as características principais de um conjunto de dados, reduzir sua dimensionalidade e estabelecer relações entre os elementos deste conjunto de dados [2].

O algoritmo principal inicia o trabalho de extração das *features* a partir da construção das *eigenfaces* (ou “auto-faces”). As *eigenfaces* são construídas da seguinte forma [1], a partir de uma base de dados com imagens de faces:

- i) A partir de uma base de dados com M imagens faciais, com tamanho padronizado em $h \times w$, cada imagem é transformada em um vetor de tamanho D e colocada em um conjunto de vetores: $\{\Gamma_1, \Gamma_2, \dots, \Gamma_M\}$;
- ii) cada face é comparada com as demais faces já disponíveis na base de dados, e um vetor de comparativo (ou diferenciação) $\Phi_i = \Gamma_i - \Psi$ é construído: Φ_i é o vetor de comparativo da face i ; Γ_i é o vetor da face i e Ψ é a face “padrão” (ou a face média), com $\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$;
- iii) a matriz de covariância $C \in \mathbb{R}^{D \times D}$, que contribui para a definição dos autovalores das *eigenfaces*, é definida conforme a seguir: $C = \frac{1}{M} \sum_{i=1}^M \Phi_i \Phi_i^T = AA^T$, com $A = \{\Phi_1, \Phi_2, \dots, \Phi_M\} \in \mathbb{R}^{D \times M}$;
- iv) a partir da matriz de covariância C e de AA^T , podemos obter os autovalores e os autovetores de C , conforme a seguir: $V = \{v_1, v_2, \dots, v_r\}$, $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_r\}$, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$, com r sendo o *rank* de A ;
- v) por fim, os autovalores e os autovetores de C são Λ e $U = AV\Lambda^{-1/2}$, no qual $U = \{u_i\}$ é o conjunto de *eigenfaces* da imagem i .

Com as *eigenfaces* construídas para as imagens de faces de uma base de dados, o processo de reconhecimento facial se dá a partir da busca pela *eigenface* que melhor se aproxima daquilo que pretendemos encontrar: ou seja, pretendemos realizar uma aproximação (*match*) entre a imagem de entrada e as imagens que estão disponíveis na base de dados. O pseudo código para o processo de reconhecimento facial encontra-se a seguir [3]:

- i) dada uma imagem facial de entrada/busca E , construímos os autovalores, autovetores e o conjunto de *eigenfaces* da imagem, conforme explicitado no processo anterior, obtendo o conjunto $U' = \{u'\}$ para a imagem de entrada;
- ii) obtemos a distância Euclidiana (d) para os conjuntos de *eigenfaces* que temos na base de dados: $d = \|U' - U_i\|$, sendo U' as *eigenfaces* da imagem de entrada e U_i é as

eigenfaces da imagem i no banco de dados;

- iii) se $d < \epsilon_1$, então a imagem i é um candidato para reconhecimento; se $\epsilon_1 < d < \epsilon_2$, então a face de entrada/busca não está disponível na base de dados e pode ser adicionada; ou, se $d > \epsilon_2$, então a imagem de entrada/busca não se trata de uma imagem facial. ϵ_1 e ϵ_2 seriam *thresholds* de distância que devem ter as *eigenfaces* da base de dados, o que norteará o funcionamento do algoritmo.

A técnica baseada em *eigenfaces* é relativamente fácil de ser compreendida e de ser codificada; existe uma redução na complexidade para representar e trabalhar com as *features* de dados de imagens faciais, o que torna possível ter base de dados grandes e reconhecimento (“busca e reconhecimento”) praticamente em tempo real. No entanto, a técnica em si pode apresentar dificuldades para trabalhar com imagens que apresentam divergência em luminosidade, escala e translação; e mudanças de expressão são dificilmente capturadas [4].

B. Fisherfaces

O *Fisherfaces*, assim como o *Eigenfaces*, tem também por motivação principal realizar o reconhecimento de padrões (não apenas faciais) partir de uma abordagem que analisa detalhes e características da imagem (*features*), ao invés de focar em regiões ou características específicas. A técnica faz uso de análise discriminante linear (LDA — *Linear Discriminant Analysis*) para analisar a similaridade ou dissimilaridade entre as características extraídas da base de dados e construir as classes de dados, aproximando os objetos (dados) de classes similares e distanciando objetos de classes distintas [2].

A técnica surgiu, no entanto, como uma forma de obter-se melhores resultados, quando comparados com *Eigenfaces* [5]. Problemas relacionados a luminosidade e variação da expressão facial são minimizados nesta técnica. A partir do que é provido pelo LDA, um único indivíduo seria capaz de gerar uma classe própria, dentro da base de dados, e imagens desse mesmo indivíduo que mudasse apenas nos aspectos luminosidade ou expressão facial estariam mais próximos (pertencentes a uma mesma classe), enquanto imagens de indivíduos diferentes estariam em classes distintas. Desta forma, o algoritmo consegue segmentar e classificar melhor os objetos na base de dados [2].

É calculada uma face média para cada classe de objetos da base de dados; uma face média para toda a base de dados; a transformação das imagens em vetores (de forma similar ao *Eigenfaces*); construção das matrizes de dispersão intra- e inter-classes; determinação das *fisherfaces* (a partir dos autovalores e autovetores). A busca na base de dados também é feita partir do cálculo da similaridade entre o objeto de entrada/busca com os demais objetos da base de dados — o que pode ser também a distância Euclidiana.

A técnica apresenta melhorias em relação ao *Eigenfaces*, no entanto, ainda assim apresenta algumas desvantagens gerais: como a técnica requer maiores cálculos para estabelecer as relações entre os objetos de uma base de dados, requer mais custo computacional, mais informações são extraídas (consumindo mais recurso de armazenamento), o que compromete um eventual uso em tempo-real da técnica [6].

C. Local Binary Patterns Histograms

O *Local Binary Patterns Histograms* (LBPH) é uma das técnicas de reconhecimento facial mais antigas, inicialmente apresentada em 1994 [7] [8]. A técnica baseia-se no *Local Binary Patterns* (LBP), que é um operador simples e poderoso para limitar e analisar a relação de um pixel com sua vizinhança, contribuindo na extração de *features* de uma imagem facial. Posteriormente, a partir da extração das *features*, o histograma de cada segmento analisado (da imagem) é comparado para se classificar o reconhecimento facial.

Diferentemente do *Fisherfaces* e do *Eigenfaces*, que são técnicas de aprendizado não-supervisionado, o LBPH é uma técnica de aprendizado supervisionado: um treinamento prévio do algoritmo é necessário. Os seguintes processos são realizados quando se utiliza o LBPH como técnica de reconhecimento facial [7]:

- i) os principais parâmetros são definidos para analisar a imagem: raio de análise de um pixel; os vizinhos de um pixel (definição de grupos); grade X ou o número de pixels na horizontal; e grade Y ou o número de pixels na vertical;
- ii) treinamento da base de dados, a partir dos parâmetros definidos anteriormente. Uma janela deslizante (com base no raio e na definição dos grupos na imagem) irá analisar a imagem, criando as *grids* e discretizando cada pixel da imagem a partir de um *threshold*. Posteriormente, para cada *grid* é criado um histograma, que serão posteriormente concatenados num único vetor, representando aquela imagem (ou face); e
- iii) por fim, para realizar a busca/reconhecimento facial, o vetor de histogramas da imagem de entrada/busca é comparado com os vetores de histogramas da base de dados, a partir da similaridade entre estes vetores (por exemplo, usando distância Euclidiana). Caso a distância respeite um limiar de similaridade, podemos inferir se houve ou não o reconhecimento (ou *match*) facial.

O algoritmo em si não é complexo, comparado ao *Fisherfaces*, e apresenta uma implementação relativamente simples. No entanto, ele apresenta dificuldades ao trabalhar com imagens que tenham bastante diferença em luminosidade ou que apresentem variação de expressão facial — e que tenham bastante ruídos nas imagens —; e como é necessário trabalhar com os histogramas, o algoritmo necessita de bastante dados armazenados para realizar o reconhecimento e atualização da base de dados, o que dificulta trabalhar com um coleção grande de objetos [9]. Algumas extensões são propostas a fim de solucionar estes problemas [8] [9].

III. CONCLUSÕES

Três técnicas de reconhecimento facial foram comparadas a partir de suas motivações individuais, seus mecanismos de funcionamento e pontos positivos e negativos. As técnicas apresentadas foram estendidas e aprimoradas, ainda que suas versões canônicas sejam utilizadas.

Dentre as técnicas utilizadas, a que apresenta ainda hoje bastante qualidade e robustez em seus resultados é a *Fisherfaces*. O Jupyter Notebook que acompanha este texto apresenta os

principais resultados que podem ser obtidos a partir de cada técnica.

REFERÊNCIAS

- [1] *Eigenfaces* - *Scholarpedia*. Disponível em: <http://www.scholarpedia.org/article/Eigenfaces>. Último acesso em 27 de Agosto de 2019.
- [2] T. D. Bissi, *Reconhecimento Facial com os algoritmos Eigenfaces e Fisherfaces*, 2018.
- [3] M. A. Turk, A. P. Pentland, "Face recognition using eigenfaces", *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA, 1991, pág. 586–591. doi: 10.1109/CVPR.1991.139758
- [4] *Eigenfaces* - *Wikipedia*. Disponível em: <https://en.wikipedia.org/wiki/Eigenface>. Último acesso em 27 de Agosto de 2019.
- [5] P. N. Belhumeur, J. P. Hespanha, D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection", in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 7, pág. 711–720, Jul. 1997. doi: 10.1109/34.598228
- [6] K. S. M. Reddy, "Comparison of Various Face Recognition Algorithms". *International Journal of Advanced Research in Science, Engineering and Technology*, v. 4, n. 2, 2017, pág. 3357–3361.
- [7] K. Salton, "Reconhecimento Facial: Como funciona o LBPH". Disponível em: <https://updatedcode.wordpress.com/2017/11/26/reconhecimento-facial-como-funciona-o-lbph/>. Último acesso em 27 de Agosto de 2019.
- [8] *Local Binary Patterns* - *Scholarpedia*. Disponível em: http://www.scholarpedia.org/article/Local_Binary_Patterns. Último acesso em 27 de Agosto de 2019.
- [9] X. Fu, W. Wei, "Centralized Binary Patterns Embedded with Image Euclidean Distance for Facial Expression Recognition". In *Proceedings of the 2008 Fourth International Conference on Natural Computation*, v. 4, 2008, IEEE Computer Society, USA, pág. 115–119. doi: 10.1109/ICNC.2008.94
- [10] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.