

Faculdade de Informática e Administração Paulista - FIAP

MBA em Artificial Intelligence e Machine Learning

Turma 11A / Mar 2019

Tecnologias de Processamento de Imagens

Prof Msc Michel Pereira Fernandes

Leonardo Gonzaga Carvalho

RM330208

Projeto Final de Avaliação Substitutiva

Proposta: “Avaliação de métodos de reconhecimento facial utilizando visão Computacional”

Este projeto tem por objetivo aprofundar o conhecimento nos classificadores de faces presentes na biblioteca OpenCV (versão 3): Eigenfaces, Fisherfaces e Local Binary Patterns Histograms.

Para cada tipo de classificador, pesquise as seguintes características:

1. Motivação
2. Funcionamento
3. Pontos positivos
4. Pontos negativos
5. Exemplo prático (utilizando Jupyter Notebook, linguagem Python)

Ao final, desenvolva uma conclusão comparando os resultados finais de cada um deles e ressaltando suas principais aplicações e casos de uso recomendados baseado em seus principais benefícios e deficiências.

A entrega será por meio de repositório no GitHub, dividido pelas seguintes pastas:

1. Pasta “doc” contém a dissertação no formato PDF
2. Pasta “lab” contém um único arquivo Jupyter Notebook exemplificando os exemplos práticos e comparando os classificadores por métricas de escolha do aluno.

É obrigatório a presença de um arquivo “readme.md” detalhando o projeto, que ficará na raiz do repositório.

Anexar no Portal do Aluno somente um arquivo texto contendo o endereço do Repositório.

Não esqueça de referenciar todas as fontes bibliográficas desta pesquisa.

Sumário

Sumário	1
1. Introdução e Motivação	2
2. Algoritmos de Reconhecimento Facial	5
2.1. Eigenfaces	5
2.1.1. Algoritmo	6
2.1.2. Vantagens e Desvantagens	9
2.2. Fisherfaces	10
2.2.1. Algoritmo	10
2.2.2. Vantagens e Desvantagens	11
2.3. Local Binary Patterns Histogram	12
2.3.1. Algoritmo	12
2.3.2. Vantagens e Desvantagens	13
3. Bibliografia e Referências	14

1. Introdução e Motivação

Percepção facial é a capacidade de se identificar e interpretar as características dos rostos (especialmente humanos) e as informações que convêm. As proporções e expressões da face humana são importantes para o reconhecimento da origem de um indivíduo, seus padrões emocionais, estado de saúde e outras informações sociais.

O reconhecimento facial é um importante mecanismo neurológico que indivíduos usam diariamente em seu convívio social desde o nascimento e durante toda a vida. Embora recém nascidos sejam capazes de facilmente compreender expressões faciais, tal tarefa é muito complexa e envolve diversas áreas no cérebro.^[1]

Desde 1964, Woody Bledsoe, Helen Chan Wolf e Charles Bisson -pioneiros do reconhecimento facial- trabalhavam na criação de sistemas computacionais para o reconhecimento de faces humanas.^[1] As décadas seguintes foram marcadas por grandes avanços neste tipo de aplicação e atualmente a detecção de faces é utilizada na robótica, sistemas de segurança, interfaces homem-máquina, indexação de imagens humanas, marketing, dentre outras.

Nosso cérebro possui células nervosas especializadas para responder a características locais específicas de uma cena, como linhas, bordas, ângulos ou movimento - como foi demonstrado por David Hubel e Torsten Wiesel. Como não vemos o mundo como elementos geométricos distintos, nosso córtex visual combina as diferentes fontes de informação em padrões úteis. Seguindo este raciocínio, o reconhecimento automático de faces busca extrair recursos significativos de uma imagem, os representar de maneira útil (computacionalmente) e os classificar.^[2]

Uma das abordagens mais intuitivas para a realização do reconhecimento facial computacional é baseada na análise das características geométricas do rosto. Sendo que um dos primeiros sistemas automatizados capazes de reconhecer faces desta maneira foi descrito por Takeo Kanade em 1974 na dissertação *Picture Processing System by Computer Complex and Recognition of Human Faces*.^[3] Nela, pontos marcadores (posição dos olhos, orelhas, nariz, etc) foram utilizados para construir “vetores de feição” (distância entre os pontos, ângulo entre eles, etc) de forma que o reconhecimento fosse realizado ao se calcular a distância euclidiana entre os vetores de característica de uma amostra e sua respectiva imagem de referência. Tal método é robusto contra variações na iluminação mas possui grande desvantagem quanto ao registro preciso dos pontos marcadores, tarefa complexa mesmo para os algoritmos do estado-da-arte atuais. Ainda assim, trabalhos mais recentes sobre o reconhecimento geométrico de faces como o artigo *Face Recognition through Geometrical Features* de R. Brunelli e T. Poggio (1995)^[4] mostra que apenas características geométricas não carregam informações suficientes para o reconhecimento facial.

Em 1999 foi lançado o OpenCV (Open Source Computer Vision), uma popular biblioteca multiplataforma de visão computacional criada pela Intel. Seu foco é a eficiência de processamento computacional de imagens, com forte foco em aplicações em tempo real. O OpenCV é escrito em C/C++ (mas também possui interfaces em Python e Java), suporta processamento multi-core de maneira otimizada em Windows/Linux/MacOS/iOS/Android e é registrado sob a licença BSD - empoderando tanto projetos acadêmicos quanto produtos comerciais.^[5] A partir da versão 2.4 do OpenCV, o módulo de reconhecimento facial foi introduzido para acelerar o desenvolvimento deste tipo de aplicação. Três algoritmos foram disponibilizados para tal tarefa: *Eigenfaces*, *Fisherfaces* e *Local Binary Pattern Histograms*.

Matthew Turk e Alex Pentland evoluíram, também em 1992, as abordagens de reconhecimento facial com o trabalho *Eigenfaces for Recognition* ^[6] em que atacaram o problema a partir de perspectivas fisiológicas e da teoria da informação: a imagem de uma face é a representação em 2D de um objeto de três dimensões, tornando a classificação mais simples. Por meio da análise de componentes principais (PCA) ^[7], o algoritmo de Eigenfaces funciona projetando amostras em um espaço de características que cubra as variações mais significativas dentre as imagens de face conhecidas - desconsiderando os conceitos humanos específicos sobre as partes do rosto (como olhos e orelhas). A operação de projeção caracteriza, então, uma face individual pela soma ponderada das características dos eigenvectors e sua comparação com os pesos dos rostos conhecidos.^[6]

Embora esse tipo de transformação seja ideal do ponto de vista da reconstrução, ela não leva em conta nenhum marcador. Portanto, seria possível que fontes externas ao rosto (como luminosidade) crie variâncias tamanhas que inviabilizem a classificação e reconhecimento de uma face.

Peter Belhumeur, João Hespanha e David Kriegman atacaram o ponto fraco do algoritmo Eigenfaces em seu artigo *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection* de 1997.^[8] Nele os autores apresentam o algoritmo Fisherfaces para reconhecimento facial, que é insensível a grandes variações na direção da iluminação e na expressão facial, simultaneamente minimizando a variância intra-classes e maximizando a variância inter-classes. Eles consideram que é possível classificar os padrões de uma face se:

1. Cada pixel de uma imagem for considerado como uma coordenada em um espaço de alta dimensão e;
2. Imagens de uma face com a mesma pose sob iluminações diferentes pertencem a um subespaço linear 3D (em relação ao item 1) e;
3. A face é uma superfície Lambertiana sem sombreamento;^[9]

No entanto, imagens de faces não são totalmente Lambertianas no mundo real. Então os autores utilizaram o discriminante linear de Fisher (LDA)^[10] para projetar a imagem em um subespaço de dimensão menor, com classes mais definidas (mesmo que sob variações severas de luminosidade) e taxas de erros menores que a técnica Eigenface.

Outros métodos de reconhecimento facial surgiram desde então, como o *Face Recognition by Elastic Bunch Graph Matching*.^[11] Porém o algoritmo apresentado no artigo *Face Recognition with Local Binary Patterns*^[12] de Timo Ahonen, Abdenour Hadid e Matti Pietikainen em 2004 se destacou pela sua eficiência, simplicidade de método e rápida extração de características. Para evitar os tratamentos de alta dimensionalidade do Eigenfaces e Fisherfaces, somente regiões específicas da imagem são analisadas, considerando-se dados de textura e formato do rosto. No algoritmo LBPH, a área da face é primeiramente dividida em pequenas regiões das quais histogramas de Local Binary Patterns^[13] são extraídos e concatenados em um único histograma enriquecido com dados espaciais. O reconhecimento é realizado utilizando um classificador de vizinho-mais-próximo (KNN)^[14] no espaço de características da amostra e com o erro Chi quadrado como medida de dissemelhança.

Os resultados experimentais apresentados no artigo claramente mostram que o método LBPH desempenha ambos algoritmos Eigenfaces e Fisherfaces.^[12]

Apesar dos avanços com o reconhecimento de faces, as técnicas mais recentes ainda apresentam espaço para melhorias. Por exemplo, os algoritmos atuais são capazes de classificar eficientemente fotos frontais e de ângulos de no máximo 20°. No entanto, fotos em perfil são extremamente desafiadoras para o estado-da-arte. Ainda assim, comparado com outras técnicas de reconhecimento de indivíduos, como as biométricas, o reconhecimento facial apresenta vantagens por funcionar em larga escala (aeroportos, estádios, etc) e não necessitar inputs específicos dos usuários.

2. Algoritmos de Reconhecimento Facial

2.1. Eigenfaces

A abordagem Eigenface é considerada por muitos como a primeira tecnologia de reconhecimento facial funcional. Refere-se a uma abordagem baseada em aparências para o reconhecimento facial, procurando capturar variações em uma coleção de imagens faciais e usar essas informações para codificar e comparar imagens de faces individuais de maneira holística. Especificamente, as eigenfaces são os principais componentes de uma distribuição de faces, ou equivalentemente, dos autovetores da matriz de covariância do conjunto de imagens faciais, onde uma imagem com N pixels é considerada um ponto (ou vetor) no espaço N -dimensional. A ideia original de se usar componentes principais para representar rostos humanos foi desenvolvida por L. Sirovich e M. Kirby em 1986 na obra *Low-dimensional procedure for the characterization of human faces* ^[15], e posteriormente usada como pilar da obra de Turk e Pentland^[6] para detecção e reconhecimento de faces.

Desde o seu desenvolvimento inicial e publicação, muitas extensões para o método original e muitos novos desenvolvimentos têm surgido em sistemas automáticos de reconhecimento facial. Eigenfaces ainda é frequentemente considerado como um benchmark para demonstração do desempenho mínimo esperado para um sistema de reconhecimento facial.^[16]

A motivação do algoritmo Eigenfaces é dupla:

- Extrair as informações faciais relevantes, que podem ou não estar diretamente relacionadas à intuição humana de características de um rosto (como olhos, nariz e lábios). Uma maneira de se fazer isso é capturar a variação estatística entre as imagens faciais.
- Representar imagens faciais de forma eficiente. A fim de reduzir a computação do espaço de complexidade, cada imagem da face pode ser representada usando um pequeno número de parâmetros.

Os eigenfaces podem ser considerados como um conjunto de características que classificam a variação global dentre as imagens faciais utilizadas no treinamento, cujos vetores de características têm as correlações desnecessárias removidas com o método dos mínimos quadrados. Cada amostra pode então ser aproximada usando um subconjunto das eigenfaces (de acordo com os maiores valores), para que cada característica possua a maior variância no conjunto de treinamento.^[6]

2.1.1. Algoritmo

Tomando como dados de treinamento um subconjunto de seis faces do dataset CMU PIE,^[17] apresentadas na Fig. 1, antes do algoritmo^[16] gerar eigenfaces as imagens faciais necessitam ser normalizadas em relação ao alinhamento de olhos e bocas:

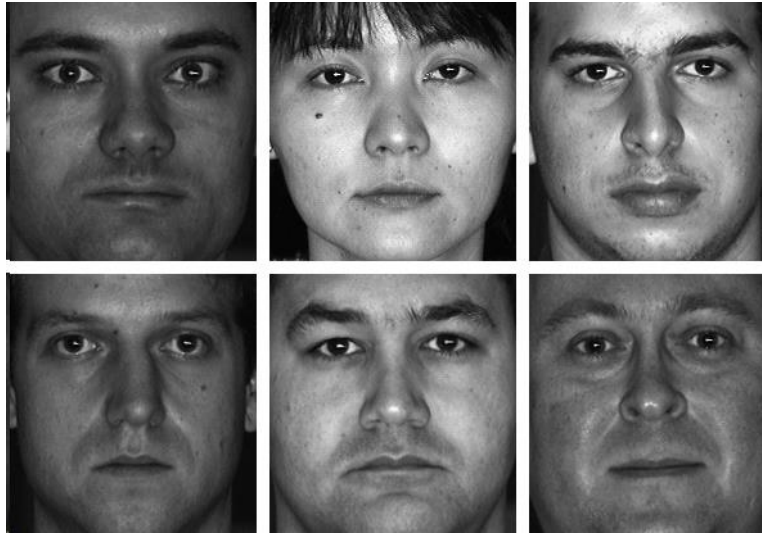


Fig 1. Exemplos de faces da base CMU PIE^[18]

Em seguida, as imagens são re-amostradas para a mesma resolução e as eigenfaces então extraídas dos dados de cada face por meio da análise de componentes principais (PCA).



Fig 2. Eigenfaces calculadas a partir das faces da Fig 1^[19]

As faces geradas na Fig 2 representam:

- a. As características médias dentre todas as faces da Fig 1
- b. A eigenface mais próxima de a

- c. A segunda eigenface mais próxima de a
- d. A antepenúltima eigenface mais próxima de a
- e. A penúltima eigenface mais próxima de a
- f. A última eigenface mais próxima de a, ou seja a mais distante

As eigenfaces abrangem um subespaço de m dimensões (em relação ao espaço de características das imagens originais) cujos vetores de características são associados aos m maiores valores. Isso resulta no chamado *espaço de faces*, cuja origem é a face média e cujos eixos são os eigenfaces, como representado na Fig 3 (considerando-se apenas um espaço 2D de duas eigenfaces):

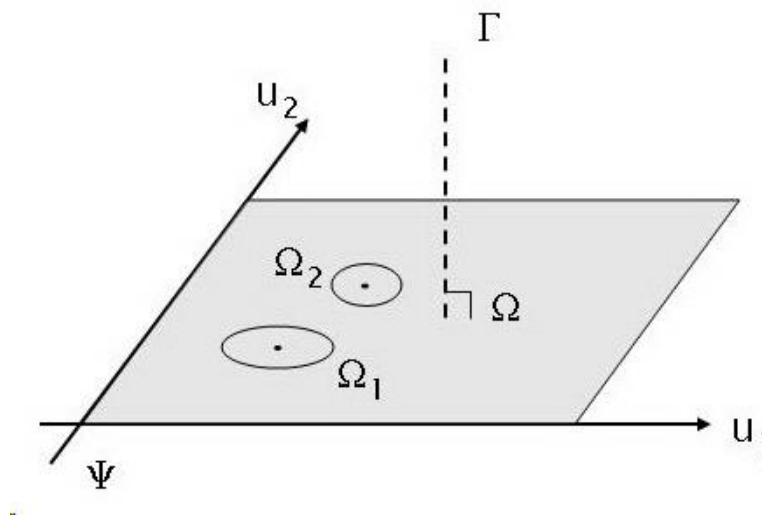


Fig 3. Espaço de faces^[20]

Assim, a detecção ou reconhecimento de faces podem ser feitas a partir do cálculo da distância euclidiana dentro ou a partir do *espaço de faces*:

- **Reconhecimento de faces:** Uma imagem N pode ser considerada pertencente a uma classe de faces F se a distância euclidiana D entre ela e o *espaço de faces* é menor que um limite L definível. A imagem necessita possuir intersecções com as eigenfaces calculadas previamente, e caso a distância D seja maior que o limite L a imagem N é considerada como desconhecida. A Fig 4 apresenta um exemplo em que a imagem superior representa uma amostra para ser classificada e a inferior a respectiva eigenface projetada. Como a distância entre a eigenface calculada e o *espaço de faces* previamente definido é menor que o threshold estabelecido, a amostra é considerada como membra da classe.

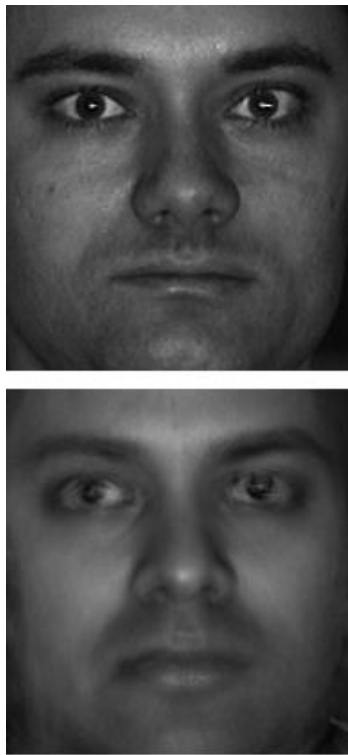


Fig 4. Amostra (cima) e respectiva eigenface calculada (baixo)^[21]

- **Detecção de faces:** Uma imagem M pode ser considerada como uma face se estiver próxima (mas não contida) do *espaço de faces*. Proximidade neste caso é definida se a distância euclidiana D é menor ou igual a um limite L definível. A Fig 5 apresenta um exemplo em que a imagem superior representa uma amostra para ser classificada e a inferior a respectiva eigenface projetada. Como a distância entre a eigenface calculada e o espaço de faces previamente definido é maior que o threshold estabelecido, a amostra é considerada como não-face.



Fig 5. Amostra (cima) e respectiva eigenface calculada (baixo)^[21]

2.1.2. Vantagens e Desvantagens

O algoritmo de Eigenfaces foi um passo importante em direção ao reconhecimento baseado em aparências para o campo de visão computacional e ainda é tido como referência de desempenho mínimo de sistemas de reconhecimento facial. Uma grande vantagem do método é a classificação do conjunto de características como um todo, utilizando toda a informação da representação facial independentemente das formas geométricas da face (olhos, nariz, orelhas e boca).^[22]

No entanto, o método é sensível à variação de iluminação, escala, pose, expressão facial e oclusão. Para seu bom funcionamento, o rosto deve ser apresentado em vista frontal, na escala apropriada, em uma condição de iluminação similar, em uma expressão definida (tipicamente neutra) e não obstruída. Para lidar com a variedade de desafios enfrentados pelos sistemas de reconhecimento facial real, ao longo dos anos muitas modificações e extensões foram propostas desde o trabalho original.

2.2. Fisherfaces

Um dos desafios fundamentais no ramo visão computacional, e aprendizado de máquina em geral, é definir uma representação de dados apropriada para a problema em questão. No reconhecimento facial, uma abordagem tradicional para se representar os dados de entrada é encontrar um subespaço de características da imagem que maior variação possível - como o *espaço de faces* do algoritmo Eigenfaces discutido no item 2.1.1.

Quando o objetivo é a classificação de imagens, a solução de mínimos quadrados do Eigenfaces pode não produzir os resultados mais desejáveis. Nesses casos, deseja-se encontrar um subespaço que mapeie os vetores amostrais de características em um único ponto de representação ao mesmo tempo em que as classes estejam o mais distante possível uma das outras. As técnicas derivadas para atingir tal objetivo são conhecidas como análise discriminante.

A técnica de análise discriminante mais utilizada foi sugerida por Ronald A. Fisher em 1936 e é chamada Análise Discriminante Linear (LDA) ou Discriminante Linear de Fisher^[10]. Quando o LDA é utilizado para encontrar a representação do subespaço de um conjunto de imagens faciais, os vetores de características resultantes que definem esse espaço são conhecidos como Fisherfaces.^[23]

2.2.1. Algoritmo

Para computar as Fisherfaces, é assumido que os dados de cada classe siga uma distribuição normal. O pseudocódigo^[24] do algoritmo de reconhecimento Fisherfaces consiste em:

1. Construir uma matriz de imagens X em que cada coluna representa uma imagem. A cada imagem é designada uma classe e seu respectivo vetor de características S_1, S_2 , etc;
2. Projetar X no subespaço de $(N - S)$ dimensões P com a matriz de rotação WP identificada pela técnica PCA, onde N é o número de amostras em X e C é o número de classes únicas;
3. Calcular as projeções (distribuições) S_w inter-classe em relação a P ;
4. Calcular as projeções (distribuições) S_B intra-classe em relação a P ;
5. Aplicar a técnica LDA a fim de maximizar a proporção do determinante entre S_w e S_B . A solução é dada pelo conjunto generalizado de vetores de características W de S_w e S_B ;
6. Obter as Fisherfaces $F = WP * W$;

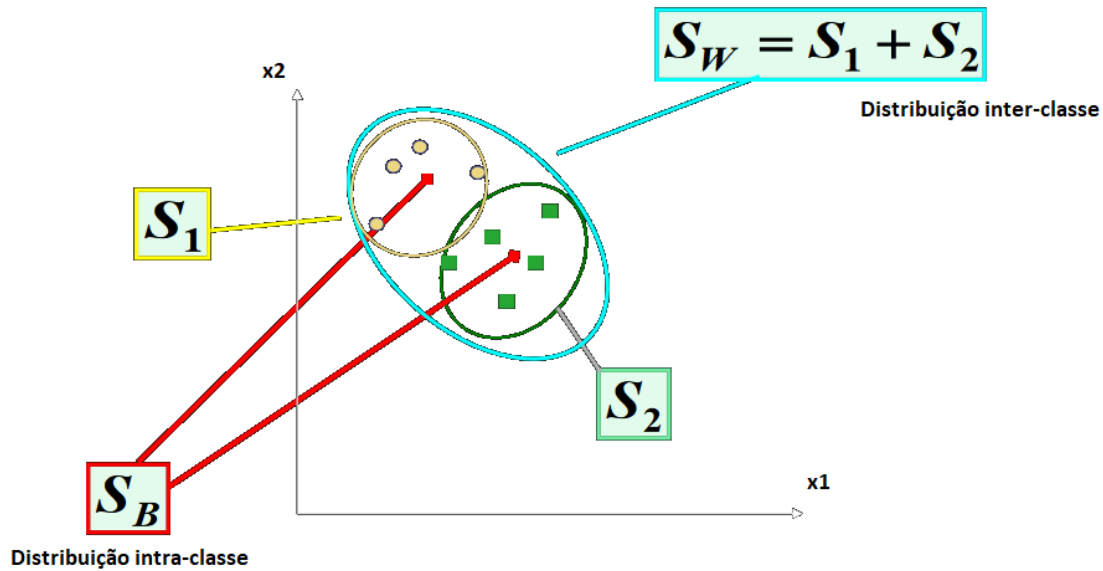


Fig 6. Distribuições intra e inter classe realizadas pelo algoritmo Fisherfaces^[25]

2.2.2. Vantagens e Desvantagens

O método Fisherface executa otimamente na extrapolação e interpolação de luminosidade e diferentes expressões faciais.^[8] Em detrimento, a categorização por Fisherface perde desempenho com datasets de treinamento pequeno ou quando um rosto é desconhecido em relação aos dados de treinamento.^[26]

2.3. Local Binary Patterns Histogram

Padrões Binários Locais (Local Binary Patterns - LBP) é um tipo de descritor visual usado para classificações no campo da visão computacional, sendo poderoso especialmente para classificação de texturas de uma imagem.^[13]

Baseada nesta tecnologia, uma abordagem alternativa às técnicas de PCA e LDA utilizadas pelos algoritmos de Eigenfaces e Fisherfaces para reconhecimento de faces foi criada: Local Binary Pattern Histograms (LBPH). O LBPH é um operador de texturas eficiente e de baixa complexidade computacional, capaz de identificar os pixels de uma imagem limitando-se a vizinhança de cada pixel e considerando o resultado como um número binário. Devido ao seu poder discriminativo responsivo em cenários desafiadores em tempo real e a sua robustez à mudanças monotônicas em escala de cinza (como variação de iluminação) ele se tornou um método muito popular em várias aplicações de visão computacional atuais.^[27]

A ideia básica do LBPH é que texturas de superfícies 2D podem ser descritas por duas medidas complementares: padrões espaciais locais e contrastes na escala de tons de cinza. O método original forma categorias para os pixels da imagem ao limiar a vizinhança 3x3 ao redor do pixel analisado, e considerando o resultado como um número binário; então, o histograma das $2^8 = 256$ diferentes categorias podem ser usadas para descrever a textura. Unido a uma simples medida de contraste local, as duas métricas provêm ótimo desempenho na segmentação não-supervisionada de texturas - os métodos baseados em análises locais são mais resilientes contra variações de pose, iluminação e rotação do que as abordagens holísticas.

Na abordagem LBPH, as ocorrências das classes LBP em uma imagem são coletadas em um histograma; a classificação é então realizada pelo cálculo de semelhanças simples de histogramas. A fim de evitar perdas de informações espaciais, o modelo codifica as informações de textura ao mesmo tempo em que se mantêm suas localizações. Em seguida, as categorias de textura LBP são usadas para criar várias descrições locais da face e combiná-las em uma descrição global.^[12]

2.3.1. Algoritmo

A método básico para o reconhecimento facial baseado em LBP, proposta por Timo Ahonen, Abdenour Hadid e Matti Pietikainen no artigo *Face Description with Local Binary Patterns: Application to Face Recognition* (2006)^[28] é composto pelos seguintes passos representados na Fig 7:

1. A imagem da uma face é dividida em regiões locais e os descritores de textura LBP são extraídos de cada região independentemente;

2. Um histograma é computado para cada uma das regiões;
3. O conjunto de histogramas é concatenado para formar um histograma com a descrição global da face e sua localização espacial, como mostrado na Fig 7
4. O histograma de características possui três níveis de informações:
 - a. As categorias LBP possuem informações quanto os padrões dos clusters de pixels;
 - b. As seções contém histogramas regionais e de detalhes de posicionamento/orientação;
 - c. O histograma concatenado de características possui informações da face como um todo.

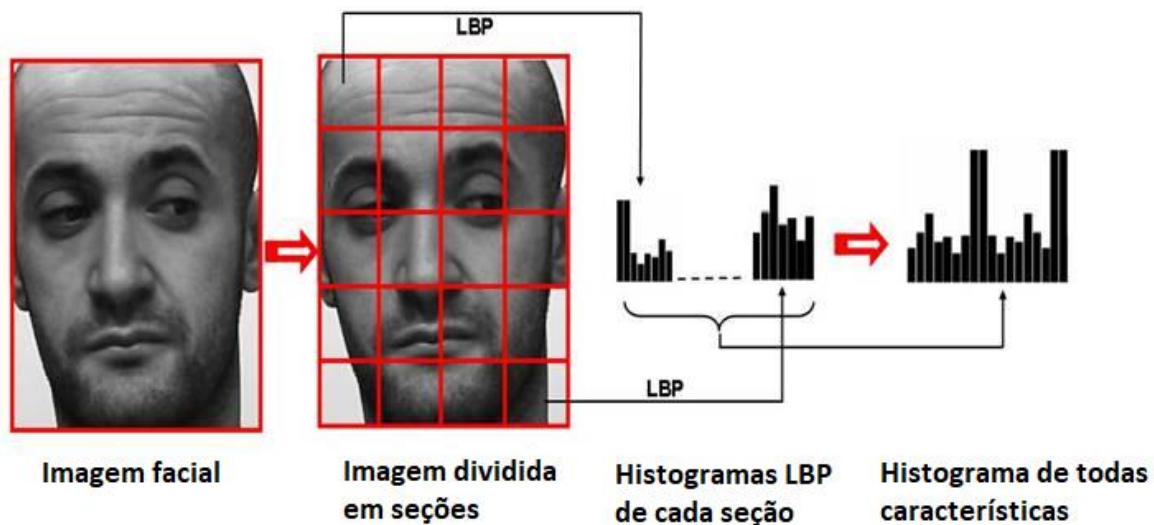


Fig 7. Fluxograma do algoritmo Local Binary Pattern Histogram^[29]

2.3.2. Vantagens e Desvantagens

Devido ao seu poder discriminativo e simplicidade computacional, o método LBPH tem sido muito bem sucedido em diversos problemas de visão computacional, como reconhecimento facial e análise de movimento (antes não considerados problemas de textura). Desde sua apresentação, um progresso significativo nestes campos foi atingido^[27] - com o algoritmo sendo amplamente utilizado em todo o mundo tanto em pesquisas quanto aplicações comerciais.

3. Bibliografia e Referências

[1] *Facial Recognition Systems*, Wikipedia (2019).

Disponível em: https://en.wikipedia.org/wiki/Facial_recognition_system

[2] *Face Recognition with OpenCV*, OpenCV (2019).

Disponível em: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_tutorial.html

[3] *Picture Processing System by Computer Complex and Recognition of Human Faces*, K. Takeo (1974).

Disponível em: https://repository.kulib.kyoto-u.ac.jp/dspace/bitstream/2433/162079/2/D_Kanade_Takeo.pdf

[4] *Face Recognition through Geometrical Features*, R. Brunelli, T. Poggio (1995).

Disponível em:

https://www.researchgate.net/profile/Roberto_Brunelli/publication/2716938_Face_Recognition_through_Geometrical_Features/links/00b7d513f3ac774d2c000000/Face-Recognition-through-Geometrical-Features.pdf

[5] *About OpenCV Library*, OpenCV (2019).

Disponível em: <https://opencv.org/>

[6] *Eigenfaces for Recognition*, T. Matthew, P. Alex (1992).

Disponível em: <https://www.mitpressjournals.org/doi/pdfplus/10.1162/jocn.1991.3.1.71>

[7] *Principal Component Analysis*, Wikipedia (2019).

Disponível em: https://en.wikipedia.org/wiki/Principal_component_analysis

[8] *Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection*, N. B. Peter, P. H. Joao, K. David (1997).

Disponível em:

https://www.researchgate.net/publication/317083802_Eigenfaces_vs_Fisherfaces_Recognition_Using_Class_Specific_Linear_Projection

[9] *Lambertian Reflectance*, Wikipedia (2019).

Disponível em: https://en.wikipedia.org/wiki/Lambertian_reflectance

[10] *Linear Discriminant Analysis*, Wikipedia (2019).

Disponível em: https://en.wikipedia.org/wiki/Linear_discriminant_analysis

[11] *Face Recognition by Elastic Bunch Graph Matching*, W. Laurenz, F. Jean-Marc (1999)

Disponível em: <http://www.face-rec.org/algorithms/ebgm/wisfelkrue99-facerecognition-jainbook.pdf>

- [12] *Face Recognition with Local Binary Patterns*, A. Timo, H. Abdenour, P. Matti (2004).
Disponível em: <http://www.face-rec.org/algorithms/ebgm/wisfelkrue99-facerecognition-jainbook.pdf>
- [13] *Local Binary Pattern*, Wikipedia (2019).
Disponível em: https://en.wikipedia.org/wiki/Local_binary_patterns
- [14] *K-nearest Neighbors Algorithm*, Wikipedia (2019).
Disponível em: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- [15] *Low-dimensional procedure for the characterization of human faces*, L. Sirovich, M. Kirby (1986),
Disponível em: <http://www.face-rec.org/interesting-papers/General/ld.pdf>
- [16] *Eigenfaces*, Scholarpedia (2019).
Disponível em: <http://www.scholarpedia.org/article/Eigenfaces>
- [17] *The CMU Multi-PIE Face Database*, Carnegie Mellon University School of Computer Science (2019)
Disponível em: <http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html>
- [18] *PIE.jpg*, Scholarpedia (2019).
Disponível em: <http://www.scholarpedia.org/article/File:PIE.jpg>
- [19] *Eigenfaces.jpg*, Scholarpedia (2019).
Disponível em: <http://www.scholarpedia.org/article/File:Eigenfaces.jpg>
- [20] *Facespace.jpg*, Scholarpedia (2019).
Disponível em: <http://www.scholarpedia.org/article/File:Facespace.jpg>
- [21] *Recons.jpg*, Scholarpedia (2019).
Disponível em: <http://www.scholarpedia.org/article/File:Recons.jpg>
- [22] *Reconhecimento Facial com os algoritmos Eigenfaces e Fisherfaces*, D. B. Thelry (2018).
Disponível em: <https://repositorio.ufu.br/bitstream/123456789/22158/3/ReconhecimentoFacialAlgoritmos.pdf>
- [23] *Fisherfaces*, Scholarpedia (2019).
Disponível em: <http://www.scholarpedia.org/article/Fisherfaces>
- [24] *Fisherfaces*, W. Philipp (2012)
Disponível em: <https://www.bytefish.de/blog/fisherfaces/>

[25] *Recognition using class specific linear projection*, S. S. Magali (2015).

Disponível em:

http://www.cs.haifa.ac.il/~rita/visual_recog_course/talks/Eigenfaces%20vs%20Fisherfaces.pdf

[26] *Face Recognition*, Indian Institute of Technology Roorkee (2017).

Disponível em: <https://www.iitr.ac.in/departments/MA/uploads/Face%20recognition.ppt>

[27] *Local Binary Patterns*, Scholarpedia (2019).

Disponível em: http://www.scholarpedia.org/article/Local_Binary_Patterns

[28] *Face Description with Local Binary Patterns: Application to Face Recognition*, A. Timo, H. Abdenour, P. Matti (2006).

Disponível em: http://www.ee.oulu.fi/mvg/files/pdf/pdf_730.pdf

[29] *LBP-face.jpg*, Scholarpedia (2019).

Disponível em: <http://www.scholarpedia.org/article/File:LBP-face.jpg>