



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE ITAPAJÉ**  
**CURSO DE GRADUAÇÃO EM SEGURANÇA DA INFORMAÇÃO**

**ANTÔNIO FÁBIO BRANDÃO DE ANDRADE**

**UTILIZAÇÃO E COMPARAÇÃO DOS MÉTODOS EIGENFACES, FISHERFACES E  
LBPH NO PROCESSO DE RECONHECIMENTO FACIAL**

**ITAPAJÉ-CEARÁ**

**2025**

ANTÔNIO FÁBIO BRANDÃO DE ANDRADE

UTILIZAÇÃO E COMPARAÇÃO DOS MÉTODOS EIGENFACES, FISHERFACES E LBPH  
NO PROCESSO DE RECONHECIMENTO FACIAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Segurança da Informação do Campus de Itapajé da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de tecnólogo em Segurança da Informação.

Orientador: Prof. Dr. Alyson Bezerra  
Nogueira Ribeiro

ITAPAJÉ-CEARÁ

2025

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Sistema de Bibliotecas  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

A565u Andrade, Antônio Fábio Brandão de.  
Utilização e Comparação dos Métodos Eigenfaces, Fisherfaces e LBPH no Processo de Reconhecimento Facial / Antônio Fábio Brandão de Andrade. – 2025.  
79 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Itapajé, Curso de Segurança da Informação, Fortaleza, 2025.  
Orientação: Prof. Dr. Alyson Bezerra Nogueira Ribeiro.

1. Reconhecimento Facial. 2. Eigenfaces. 3. Fisherfaces. 4. LBPH. 5. Visão Computacional. I. Título.  
CDD 005.8

---

ANTÔNIO FÁBIO BRANDÃO DE ANDRADE

UTILIZAÇÃO E COMPARAÇÃO DOS MÉTODOS EIGENFACES, FISHERFACES E LBPH  
NO PROCESSO DE RECONHECIMENTO FACIAL

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Segurança da Informação do Campus de Itapajé da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de tecnólogo em Segurança da Informação.

Aprovada em: 25/02/2025

BANCA EXAMINADORA

---

Prof. Dr. Alyson Bezerra Nogueira  
Ribeiro (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Juan Sebastian Toquica Arenas  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Eduardo de Melo Mapurunga  
Universidade Federal do Ceará (UFC)

Dedico este trabalho, primeiramente, à minha querida mãe, que me apoiou de forma incondicional em todos os momentos desta conquista, sempre almejando meu sucesso. Também dedico a todos os meus familiares e amigos, que ao longo desta jornada foram fundamentais para que eu mantivesse minha perseverança e força de vontade para superar os desafios da graduação.

## **AGRADECIMENTOS**

Ao Prof. Dr. Alyson Bezerra Nogueira Ribeiro, pela excelente orientação. Aos professores participantes da banca examinadora Juan Sebastian Toquica Arenas e Eduardo de Melo Mapurunga pelo tempo, pelas valiosas colaborações e sugestões.

“A imaginação é superior à ciência, porque a ciência é finita, enquanto a imaginação envolve todo o mundo.”

(Albert Einstein)

## RESUMO

A segurança digital tem sido um tema cada vez mais relevante diante do avanço tecnológico e do aumento das ameaças cibernéticas. Nesse contexto, o reconhecimento facial surge como uma solução promissora para a autenticação e controle de acesso. Este trabalho apresenta uma comparação entre os métodos de reconhecimento facial *Eigenfaces*, *Fisherfaces* e *Local Binary Patterns Histograms* (LBPH). O estudo justifica-se pela crescente demanda por métodos precisos e eficientes de identificação facial em contextos que exigem altos níveis de controle de acesso e proteção de informações. Inicialmente, é explorada a fundamentação teórica relacionada à visão computacional, detecção de faces e reconhecimento facial. No processo metodológico, os algoritmos são implementados utilizando a biblioteca *OpenCV* em *Python*. A etapa de aquisição de imagens é realizada a partir de dois *datasets*: um criado pelo autor e outro obtido na internet. A fase de pré-processamento das imagens inclui conversão para a escala de cinza, redimensionamento e aplicação de filtros. Essa etapa é fundamental, pois fornece a base necessária para as etapas seguintes de detecção e reconhecimento facial. Os algoritmos são avaliados em termos de acurácia, precisão, *recall* e *F1-score*. Os resultados indicam que o algoritmo LBPH apresenta a melhor performance geral, com uma acurácia de 98%, seguido pelo *Fisherfaces* com 92%, e pelo *Eigenfaces* com 88%. Além disso, são discutidas as principais vantagens e limitações de cada algoritmo. Este estudo contribuiu para a compreensão comparativa dos métodos de reconhecimento facial e oferece um diferencial para a escolha do algoritmo mais adequado para sistemas de controle de acesso em ambientes internos.

**Palavras-chave:** reconhecimento facial; eigenfaces; fisherfaces; LBPH; visão computacional.

## ABSTRACT

Digital security has become an increasingly relevant topic in light of technological advancements and the rise of cyber threats. In this context, facial recognition emerges as a promising solution for authentication and access control. This work presents a comparison between the Eigenfaces, Fisherfaces, and Local Binary Patterns Histograms (LBPH) methods for facial recognition. The study is justified by the growing demand for accurate and efficient facial identification methods in contexts requiring high levels of access control and information protection. Initially, the theoretical foundations related to computer vision, face detection, and facial recognition are explored. In the methodological process, the algorithms are implemented using the OpenCV library in Python. The image acquisition phase is carried out using two datasets: one created by the author and another obtained from the internet. The image preprocessing stage includes conversion to grayscale, resizing, and the application of filters. This step is crucial, as it provides the necessary foundation for the subsequent stages of face detection and recognition. The algorithms are evaluated in terms of accuracy, precision, recall, and F1-score. The results indicate that the LBPH algorithm demonstrates the best overall performance, with an accuracy of 98%, followed by Fisherfaces with 92%, and Eigenfaces with 88%. Additionally, the main advantages and limitations of each algorithm are discussed. This study contributes to a comparative understanding of facial recognition methods and provides a valuable reference for selecting the most suitable algorithm for access control systems in indoor environments.

**Keywords:** facial recognition; eigenfaces; fisherfaces; LBPH; computer vision.

## LISTA DE FIGURAS

Figura 1 – Etapas padrão no processamento de imagens em visão computacional . . . . .	17
Figura 2 – Exemplo de uma matriz bidimensional com várias linhas e colunas . . . . .	20
Figura 3 – (a) Imagem contínua projetada em uma matriz de sensores. (b) Resultado da amostragem e quantização da imagem . . . . .	22
Figura 4 – Processo de discretização de uma imagem . . . . .	23
Figura 5 – (a) Imagem representada graficamente como uma superfície. (b) Imagem representada como uma matriz de intensidade visual. (c) Imagem representada como uma matriz numérica 2-D . . . . .	24
Figura 6 – Falso contorno gerado pela redução dos níveis de cinza . . . . .	25
Figura 7 – Efeito da resolução espacial na qualidade da imagem . . . . .	26
Figura 8 – Padrões possíveis . . . . .	27
Figura 9 – Exemplo Aplicando recursos de Haar sobre uma imagem facial . . . . .	27
Figura 10 – Retângulo D de posição arbitrária . . . . .	28
Figura 11 – fluxograma de funcionamento da cascata dos classificadores . . . . .	29
Figura 12 – Face Média e Eigenfaces . . . . .	30
Figura 13 – Subtração da face média . . . . .	31
Figura 14 – Representação de um espaço de faces no plano cartesiano . . . . .	33
Figura 15 – Projeção do espaço com a LDA . . . . .	34
Figura 16 – Redução da dimensionalidade com a PCA . . . . .	34
Figura 17 – Projeção das imagens no subespaço . . . . .	35
Figura 18 – Determinando os <i>pixels</i> vizinhos à partir do <i>pixel</i> central . . . . .	36
Figura 19 – Transformação LBP . . . . .	37
Figura 20 – Histograma de Padrões Binários Locais . . . . .	37
Figura 21 – Exemplo da extração do histograma da imagem . . . . .	38
Figura 22 – Exemplo de LBP circular . . . . .	38
Figura 23 – Detecção de face subdomínio da detecção de objetos . . . . .	39
Figura 24 – Etapas padrão de detecção e reconhecimento facial . . . . .	39
Figura 25 – Desafios da detecção facial . . . . .	40
Figura 26 – Fluxograma do algoritmo de Viola Jones . . . . .	41
Figura 27 – Característica HOG . . . . .	41
Figura 28 – Detecção dos pontos de referência . . . . .	42

Figura 29 – Classificação dos métodos de RF . . . . .	44
Figura 30 – Procedimento metodológico adotado . . . . .	49
Figura 31 – Exemplos de imagens dos <i>datasets</i> . . . . .	50
Figura 32 – Etapas de pré-processamento de imagem . . . . .	50
Figura 33 – Exemplo de Imagens pós pré-processamento . . . . .	51
Figura 34 – Demarcação da face detectada . . . . .	52
Figura 35 – Diagrama das etapas do processo de treinamento dos algoritmos . . . . .	54
Figura 36 – Diagrama etapas do processo de detecção e reconhecimento facial . . . . .	55
Figura 37 – Matriz Confusão Eigenfaces . . . . .	60
Figura 38 – Matriz Confusão Fisherfaces . . . . .	62
Figura 39 – Matriz Confusão LBPH . . . . .	64

## LISTA DE TABELAS

Tabela 1 – Parâmetros para o algoritmo Eigenfaces . . . . .	53
Tabela 2 – Parâmetros para o algoritmo Fisherfaces . . . . .	53
Tabela 3 – Parâmetros para o algoritmo LBPH . . . . .	53
Tabela 4 – Métricas de desempenho <i>Eigenfaces</i> . . . . .	58
Tabela 5 – Métricas de desempenho <i>Fisherfaces</i> . . . . .	60
Tabela 6 – Métricas de desempenho <i>LBPH</i> . . . . .	62
Tabela 7 – Tempo de treinamento dos algoritmos . . . . .	65

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Objetivos</b>	<b>15</b>
<b>1.2</b>	<b>Objetivos específicos</b>	<b>16</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
<b>2.1</b>	<b>Visão computacional</b>	<b>17</b>
<b>2.2</b>	<b>Processamento digital de imagens</b>	<b>19</b>
<b>2.2.1</b>	<i>Amostragem e quantização de imagens</i>	<b>21</b>
<b>2.2.2</b>	<i>Representação de imagens digitais</i>	<b>23</b>
<b>2.2.3</b>	<i>Resolução espacial e de intensidade</i>	<b>25</b>
<b>2.3</b>	<b>Classificador Haar</b>	<b>26</b>
<b>2.4</b>	<i>Eigenfaces</i>	<b>29</b>
<b>2.5</b>	<b>Fisherfaces</b>	<b>33</b>
<b>2.6</b>	<b>Local Binary Patterns Histograms(LBPH)</b>	<b>35</b>
<b>2.7</b>	<b>Detecção de faces</b>	<b>38</b>
<b>2.7.1</b>	<i>Extração das características</i>	<b>43</b>
<b>2.8</b>	<b>Reconhecimento facial</b>	<b>44</b>
<b>2.8.1</b>	<i>Métodos baseados em imagens</i>	<b>45</b>
<b>2.8.1.1</b>	<i>Métodos de Aparência</i>	<b>45</b>
<b>2.8.1.2</b>	<i>Métodos de Pontos de Referência</i>	<b>45</b>
<b>2.8.1.3</b>	<i>Métodos híbridos</i>	<b>46</b>
<b>2.8.2</b>	<i>Métodos baseados em vídeos</i>	<b>46</b>
<b>2.8.2.1</b>	<i>Métodos baseados em conjuntos</i>	<b>47</b>
<b>2.8.2.2</b>	<i>Métodos baseados em sequências</i>	<b>47</b>
<b>3</b>	<b>METODOLOGIA</b>	<b>48</b>
<b>3.1</b>	<b>Tecnologias utilizadas</b>	<b>48</b>
<b>3.2</b>	<b>Pipeline</b>	<b>49</b>
<b>3.3</b>	<b>Aquisição das imagens</b>	<b>50</b>
<b>3.4</b>	<b>Pré-processamento</b>	<b>50</b>
<b>3.5</b>	<b>Método de detecção facial</b>	<b>51</b>
<b>3.6</b>	<b>Parametrização dos modelos</b>	<b>52</b>

3.7	Treinamento dos modelos . . . . .	53
3.8	Usabilidade dos modelos . . . . .	54
3.9	Avaliação dos modelos . . . . .	55
4	<b>RESULTADOS</b> . . . . .	58
4.1	Resultados <i>Eigenfaces</i> . . . . .	58
4.2	Resultados <i>Fisherfaces</i> . . . . .	60
4.3	Resultados LBPH . . . . .	62
4.4	Comparação entre os algoritmos <i>Eigenfaces</i> , <i>Fisherfaces</i> , LBPH . . . . .	64
5	<b>CONCLUSÃO</b> . . . . .	66
	<b>REFERÊNCIAS</b> . . . . .	67
	<b>APÊNDICES</b> . . . . .	75
	APÊNDICE A – Treinamento dos Modelos . . . . .	75
	APÊNDICE B – Testando os modelos . . . . .	79
	APÊNDICE C – Criar dataset . . . . .	80

## 1 INTRODUÇÃO

O avanço tecnológico tem possibilitado o desenvolvimento de sistemas de segurança digital cada vez mais sofisticados, nos quais a identificação facial desempenha um papel relevante. De acordo com Saragih *et al.* (2022), o reconhecimento facial computacional é uma técnica que habilita o computador a detectar e identificar rostos em imagens de forma automática. Nesse contexto, o reconhecimento facial emerge como uma solução confiável e amplamente aceita. Este estudo justifica-se pela demanda crescente por métodos precisos e eficientes de identificação facial, com ênfase em sua aplicabilidade não invasiva em contextos que exigem altos níveis de controle de acesso e proteção de informações.

Diversos estudos têm explorado a aplicação e o aprimoramento do reconhecimento facial, destacando diferentes abordagens e técnicas. Surve *et al.* (2020) desenvolveram um sistema de registro de frequência baseado no algoritmo Haar Cascade, demonstrando eficiência na captura de imagens e no processamento rápido de reconhecimento facial, permitindo a identificação automática com data e hora. Ele salva dados, treina e reconhece rostos com um clique, registrando frequência automaticamente com data e hora. De forma semelhante, Moreira (2021) realizou um estudo comparativo entre as técnicas PCA e LDA para a extração de atributos, utilizando os bancos de dados ORL e FLW. Os resultados indicam que o banco B apresentou melhor desempenho no banco ORL, com taxa média de 94,5%, enquanto no banco LFW o conjunto D obteve uma média de 86,2%.

A comparação entre diferentes métodos também foi analisada por Jr e Carvalho (2023) que avaliaram o desempenho do Haar Cascade e das Redes Neurais Convolucionais (CNN) na detecção e reconhecimento facial. Utilizando um conjunto de 1000 imagens provenientes das bases YaleFaces, LFW e Dogs vs Cats, os autores concluíram que o modelo CNN obteve acurácia de 98,9%, superando significativamente os 81,7% do Haar Cascade. Essa superioridade reforça a tendência de adoção de redes neurais profundas para melhorar a precisão do reconhecimento facial. Outra abordagem é apresentada por Ishida (2023) propôs um sistema de controle de acesso baseado em reconhecimento facial. O sistema captura as imagens das faces por meio de uma câmera acoplada a uma placa com microcontrolador, processando-as localmente e transmitindo os dados para um servidor. O servidor, por sua vez, compara as imagens recebidas com um banco de dados previamente cadastrado, identificando usuários registrados de forma remota.

Além dessas abordagens, Silva *et al.* (2023) realizou uma avaliação dos algoritmos EigenFaces e FisherFaces, destacando que o primeiro apresentou baixa confiança e alta taxa

de falsos positivos. Em contrapartida, o FisherFaces demonstrou maior robustez, com menor incidência de falsos positivos e uma taxa de acerto aceitável, destacando-se como uma opção mais eficaz para o reconhecimento facial.

Esses estudos reforçam a relevância do reconhecimento facial para a segurança digital e justificam a necessidade de aprimorar os algoritmos existentes. O propósito deste estudo é avaliar o desempenho de três dos algoritmos de reconhecimento facial de destaque na literatura: *Eigenfaces*, *Fisherfaces* e *Local Binary Patterns Histograms* (LBPH). O *Eigenfaces*, proposto por Turk e Pentland (1991), é um método baseado em análise de componentes principais (PCA) para reconhecimento facial. *Fisherfaces*, também conhecido como Discriminante Linear de Fisher (LDA), apresentado por Belhumeur *et al.* (1997), visa maximizar a separação entre as classes no espaço de características. Enquanto isso, o LBPH, introduzido por Ahonen *et al.* (2006), destaca-se por sua robustez a variações de iluminação e expressões faciais.

Dessa forma, a análise comparativa desses algoritmos permitirá compreender suas vantagens e limitações. Espera-se que os resultados obtidos contribuam para a escolha do algoritmo mais adequado para a implantação em sistemas de segurança, oferecendo uma compreensão mais profunda sobre as melhores práticas e estratégias a serem adotadas.

Este trabalho está estruturado em capítulos para facilitar sua compreensão. O Capítulo 1 apresenta uma contextualização do tema, esclarecendo os objetivos e a motivação do estudo. O Capítulo 2 traz a fundamentação teórica relacionada ao tema, com discussões sobre visão computacional, detecção de faces e os algoritmos de detecção e reconhecimento facial utilizados no estudo. O Capítulo 3 detalha a metodologia adotada e os materiais utilizados, incluindo explicações sobre as ferramentas computacionais empregadas e a implementação do trabalho proposto. O Capítulo 4 apresenta e analisa os resultados obtidos na comparação dos algoritmos. Por fim, o Capítulo 5 oferece a conclusão do estudo, sintetizando os principais achados e sugerindo possíveis aplicações futuras.

## 1.1 Objetivos

O propósito deste trabalho é empregar técnicas de visão computacional para avaliar o desempenho de algoritmos de reconhecimento facial, com o objetivo de identificar sua aplicabilidade no desenvolvimento de ferramentas de controle de acesso. De maneira específica, serão analisados três algoritmos amplamente citados na literatura: *Eigenfaces* (TURK; PENTLAND, 1991), *Fisherfaces* (BELHUMEUR *et al.*, 1997) e LBPH (AHONEN *et al.*, 2006), com o intuito

de avaliar a eficácia de cada um.

## **1.2 Objetivos específicos**

Para melhor compreensão do desenvolvimento deste trabalho, foram listados os seguintes objetivos específicos:

- a) criação e aquisição de Dataset a partir de vídeos;
- b) implementação de detecção de face;
- c) treinar e testar os algoritmos;
- d) gerar métricas de desempenho e discutir os resultados.

## 2 FUNDAMENTAÇÃO TEÓRICA

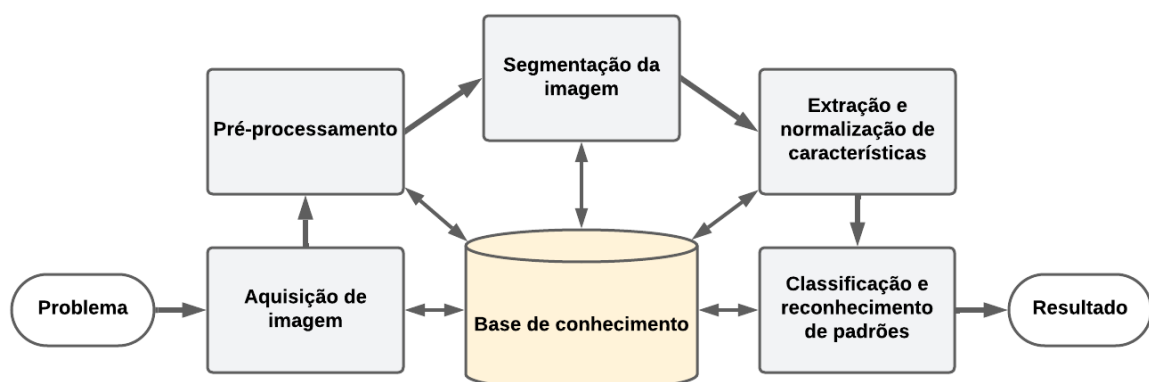
Para Fonseca (2002), todo trabalho científico começa com uma pesquisa bibliográfica, que possibilita ao pesquisador compreender o que já foi estudado sobre o tema. Algumas pesquisas científicas se concentram exclusivamente nessa abordagem, buscando referências teóricas publicadas para obter informações ou conhecimentos prévios sobre o problema em questão (FONSECA, 2002, 31). Dito isso, neste capítulo são abordados os principais conceitos técnicos e teóricos que serviram como base para o desenvolvimento do trabalho. São apresentados os algoritmos, bem como as técnicas computacionais baseadas em visão computacional para a detecção e reconhecimento facial.

### 2.1 Visão computacional

Visão computacional é um campo da inteligência artificial que lida com a capacidade de máquinas de interpretar e entender imagens e vídeos. Segundo Gonzalez e Woods (2010), visão computacional é definida como o processo de extração de informações úteis de imagens digitais por meio de algoritmos e técnicas computacionais.

Conforme Almeida (2018), a estrutura de um sistema de visão computacional é composta por etapas sequenciais que se baseiam na descrição de um problema visual. Todas essas etapas estão interligadas a uma base de conhecimento que reúne os resultados obtidos, como apresenta o diagrama de blocos na Figura 1.

Figura 1 – Etapas padrão no processamento de imagens em visão computacional



Fonte: Adaptado de (GONZALEZ; WOODS, 2018)

Nesta seção, serão descritas as etapas sequenciais do processamento de imagens.

a) aquisição de imagem: é a primeira etapa de um processamento de imagens de

um sistema de visão computacional. A aquisição pode ser tão simples quanto receber uma imagem que já está em formato digital, afirma Gonzalez e Woods (2018). O processo de captura considera diversos fatores, incluindo o tipo de sensor utilizado para aquisição (como ultrassom, ressonância magnética ou câmera digital), a quantidade de sensores no sistema, seu posicionamento em relação ao campo de observação, a resolução da captura e as condições de iluminação do ambiente analisado (no caso de sensores luminosos). Normalmente a imagem a ser analisada pode estar em algum formato padrão, tais como \*.jpg, \*.png, \*.bmp, etc, ou estar apenas armazenada na memória dinâmica do sistema, em forma de uma variável e compactada aos moldes de uma matriz de *pixels*. (JUNIOR, 2010);

- b) pré-processamento: envolve elementos que assegurem a qualidade da imagem, tais como redução de ruídos, ajuste de luz e contraste; processamento morfológico e compactação de imagem (GONZALEZ; WOODS, 2018). Segundo Filho e Neto (1999), o pré-processamento é uma etapa do processamento de imagens que permite duas principais aplicações: aprimorar informações visuais para interpretação humana e facilitar a análise automática de dados pelo computador, de informações extraídas de uma cena;
- c) segmentação de imagem: divide uma imagem em suas partes ou objetos constituintes. A etapa de segmentação separa uma imagem em regiões homogêneas, facilitando a extração de informações. Por exemplo, em imagens monocromáticas, algoritmos identificam áreas com diferentes níveis de cinza (GONZALEZ; WOODS, 2018). Para Padilha *et al.* (2003), "Quando analisamos uma imagem, geralmente não queremos outra imagem como resultado, mas sim uma descrição detalhada dela." Essa descrição refere-se, em regra, a partes da imagem e esse processo de decomposição da imagem é conhecido como segmentação;
- d) extração e normalização de características: O resultado da segmentação é usado para coletar dados de *pixel*, que servem de base para identificar padrões (GONZALEZ; WOODS, 2018). Os resultados da segmentação são convertidos em formato computacional, permitindo extração de características relevantes (FILHO; NETO, 1999);
- e) classificação e reconhecimento de padrões: responsável por classificar ou agrupar as imagens com base em seus conjuntos de características. Por exemplo, ao observar

a imagem de uma única laranja, é provável que você consiga identificá-la como pertencente à classe "laranja", com base em atributos como cor, textura da casca, formato e tamanho (BACKES; JUNIOR, 2019). Gonzalez e Woods (2010), afirmam que é o processo de identificar um objeto e atribuir-lhe um rótulo (como "veículo") com base em suas características, utilizando métodos como *deep learning* e redes neurais convolucionais.

O reconhecimento facial é um bom exemplo de como as etapas de visão computacional funcionam na prática. Ele usa essas etapas para processar imagens de rostos e transformá-las em informações que permitem identificar ou verificar pessoas. Isso mostra como os princípios da visão computacional são úteis em tecnologias que impactam diretamente o dia a dia, como sistemas de segurança e autenticação (ORVALHO, 2019).

## 2.2 Processamento digital de imagens

A representação visual de um objeto é denominada imagem, podendo ser gerada por diversos dispositivos eletrônicos, como câmeras de vídeo, *scanners* e outros equipamentos similares. Esses dispositivos capturam as características visuais do objeto e as convertem em um formato digital, permitindo sua análise, armazenamento e manipulação (FILHO; NETO, 1999).

Uma imagem pode ser vista como uma função em duas dimensões,  $f(x,y)$ , onde  $x$  e  $y$  são coordenadas no espaço, ou seja, no plano. O valor de  $f$  em cada ponto  $(x,y)$ , é o que chamamos de intensidade ou nível de cinza naquele lugar da imagem. Quando  $x$ ,  $y$  e os valores de intensidade de  $f$  são números finitos e bem definidos, essa função é chamada de imagem digital, como ilustra a Figura 2. Nessa definição se encaixam fotografias, que são formadas a partir da faixa visível de ondas eletromagnéticas, até as imagens geradas por tecnologias mais avançadas, como os raios X e o ultrassom. Essa diversidade indica que uma imagem não se limita àquilo que o olho humano pode perceber, mas pode também revelar informações sobre a realidade que estão além do nosso espectro visual (GONZALEZ; WOODS, 2010).

Figura 2 – Exemplo de uma matriz bidimensional com várias linhas e colunas

0	0	8	41	77	119	178	221	234	248
0	8	28	41	77	128	192	234	248	255
0	8	28	41	119	178	221	248	255	248
0	8	41	77	128	192	248	255	248	234
8	28	77	119	178	234	255	248	221	192
28	77	119	178	221	248	255	234	221	192
51	95	128	192	234	255	248	234	178	150
77	119	150	221	248	248	234	192	150	128

Fonte: (BETTER PHOTOGRAPHY TUTORIALS, 2011)

O Processamento Digital de Imagens (PDI) é o tratamento de imagens digitais por meio de algoritmos, transformando dados visuais em informações úteis. Ele é fundamental em diversas áreas, como robótica, sistemas de visão computacional e detecção de objetos. Uma imagem digital é formada por *pixels*, pequenos pontos coloridos que compõem a imagem. O PDI permite melhorar a qualidade das imagens, extrair informações relevantes e auxiliar na tomada de decisões em diversas aplicações (ZHOU *et al.*, 2010).

O PDI é uma área da ciência da computação que se dedica à manipulação e análise de imagens digitais. Ele abrange uma variedade de técnicas e algoritmos que permitem melhorar a qualidade das imagens, extrair informações relevantes e habilitar a compreensão automática de cenas visuais por máquinas (GONZALEZ; WOODS, 2010).

Ainda segundo o que foi dito por Silva (2014) Silva (2014) “O processamento de imagens é realizado em etapas bem definidas que compartilham informações entre si para produzir o resultado esperado, de forma que o resultado de cada etapa interfere diretamente nas etapas subsequentes”, elucidando ainda mais a importância da etapa no processo de PDI.

Contudo, conforme mencionado por Gonzalez e Woods (2010), a definição de PDI pode variar de acordo com o autor. Isso demonstra que o Processamento Digital de Imagens, sendo uma área na qual tanto a entrada quanto a saída são necessariamente imagens, faz com que o cálculo da intensidade média de uma imagem não seja considerado um processamento.

Para lidar com questões conceituais relacionadas a PDI, são empregados paradigmas específicos de pré-processamento.

Essas operações computacionais são conhecidas como processos de nível baixo, médio e alto da etapa de pré-processamento.

- a) nível baixo: Para esse processo envolve operações no pré-processamento, de redução de ruídos, realce de contraste, aguçamentos de imagens, onde o processo de entrada e saída são sempre imagens;
- b) nível médio: Nesta parte como separar a imagem em partes ou objetos, classificar e descrevê-los, onde a entrada é uma imagem e a saída um objeto extraído dessa imagem;
- c) nível alto: Por fim a última parte do processamento é onde ocorre a interpretação do objeto, conferindo sentido aos dados abstraídos da imagem. As informações extraídas são exibidas na saída do sistema, representando características da imagem de entrada. Esse estágio pode empregar técnicas como redes neurais e algoritmos genéticos para tomada de decisões, e, mesmo quando não atinge total eficácia, ainda é considerado parte do PDI (GONZALEZ; WOODS, 2010).

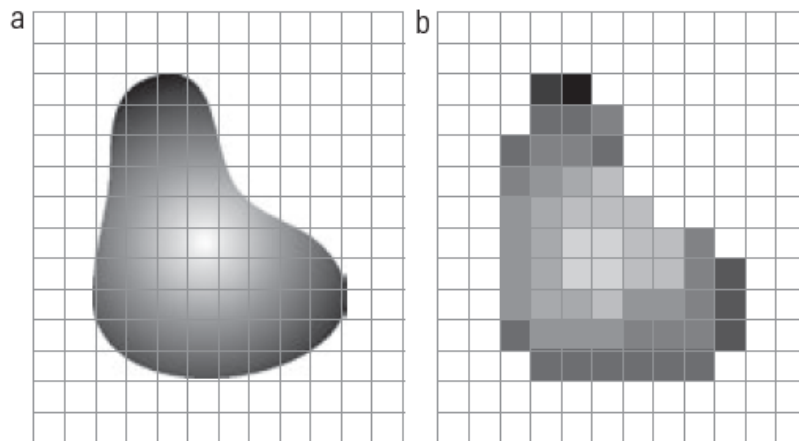
### ***2.2.1 Amostragem e quantização de imagens***

Uma imagem monocromática pode ser representada matematicamente por uma função  $f(x,y)$ , que descreve a intensidade luminosa em cada ponto das coordenadas espaciais  $(x,y)$ . O valor dessa função é proporcional ao brilho (ou nível de cinza) da imagem no ponto correspondente. Para que uma imagem possa ser processada digitalmente, é necessário convertê-la de sua forma contínua (analógica) para uma forma discreta (digital). Isso ocorre por meio de dois processos fundamentais: amostragem e quantização. Esses processos transformam a saída de dispositivos de aquisição, como câmeras, que capturam sinais contínuos, em uma representação digital (GONZALEZ; WOODS, 2010).

- a) amostragem: Refere-se à escolha de um número finito de pontos do mundo real para compor a imagem digital. Esse processo define a resolução espacial da imagem, ou seja, o nível de detalhamento em termos de coordenadas  $(x,y)$ ;
- b) quantização: Consiste em atribuir a cada ponto amostrado um valor discreto de intensidade luminosa, representando o nível de cinza. Esse processo determina a precisão dos níveis de intensidade registrados em cada *pixel* da imagem digital.

A Figura 3(a) mostra uma imagem contínua projetada sobre uma matriz de sensores. Já a Figura 3(b) apresenta a imagem após passar pelos processos de amostragem e quantização. A qualidade da imagem digital depende, principalmente, da quantidade de amostras e dos níveis de intensidade usados nesses processos.

Figura 3 – (a) Imagem contínua projetada em uma matriz de sensores. (b) Resultado da amostragem e quantização da imagem



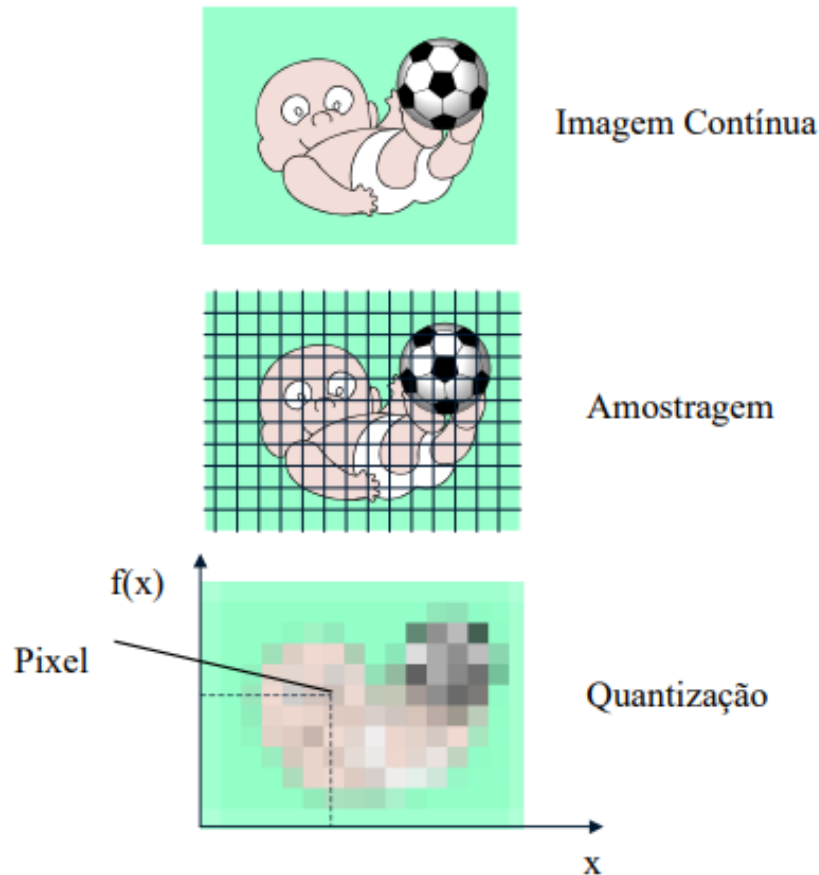
Fonte: (GONZALEZ; WOODS, 2010)

De acordo com Marques Filho e Neto (1999), os conceitos de amostragem e quantização referem-se a dois processos principais na digitalização de imagens:

- a) amostragem é o processo de discretização espacial. Ele converte uma imagem analógica contínua em uma matriz de  $M \times N$  pontos (*pixels*). Cada pixel representa a intensidade da imagem em um ponto específico. Quanto maior os valores de  $M$  e  $N$ , maior será a resolução da imagem, pois haverá mais *pixels* para representar os detalhes espaciais;
- b) quantização é o processo de discretização em amplitude. Ele atribui a cada *pixel* um valor inteiro dentro de um intervalo, geralmente de 0 a  $2^n - 1$ , onde  $n$  representa o número de bits do conversor analógico-digital (A/D). Quanto maior o valor de  $n$ , maior será o número de níveis de cinza disponíveis na imagem, permitindo maior precisão na representação dos tons.

A Figura 4 ilustra como uma imagem analógica (contínua) é transformada em uma imagem digital (discreta) através dos processos de amostragem e quantização.

Figura 4 – Processo de discretização de uma imagem



Fonte: (SCURI, 2002)

### 2.2.2 Representação de imagens digitais

A representação de imagens digitais envolve dois processos essenciais: amostragem e quantização. A amostragem converte uma imagem contínua em uma matriz bidimensional de coordenadas discretas  $f(x, y)$ , composta por  $M$  linhas e  $N$  colunas, onde  $x$  e  $y$  são números inteiros que representam as coordenadas espaciais. A quantização, por sua vez, atribui valores discretos de intensidade (níveis de cinza) a cada *pixel* dessa matriz, baseando-se em uma escala de intensidade predefinida (GONZALEZ; WOODS, 2010).

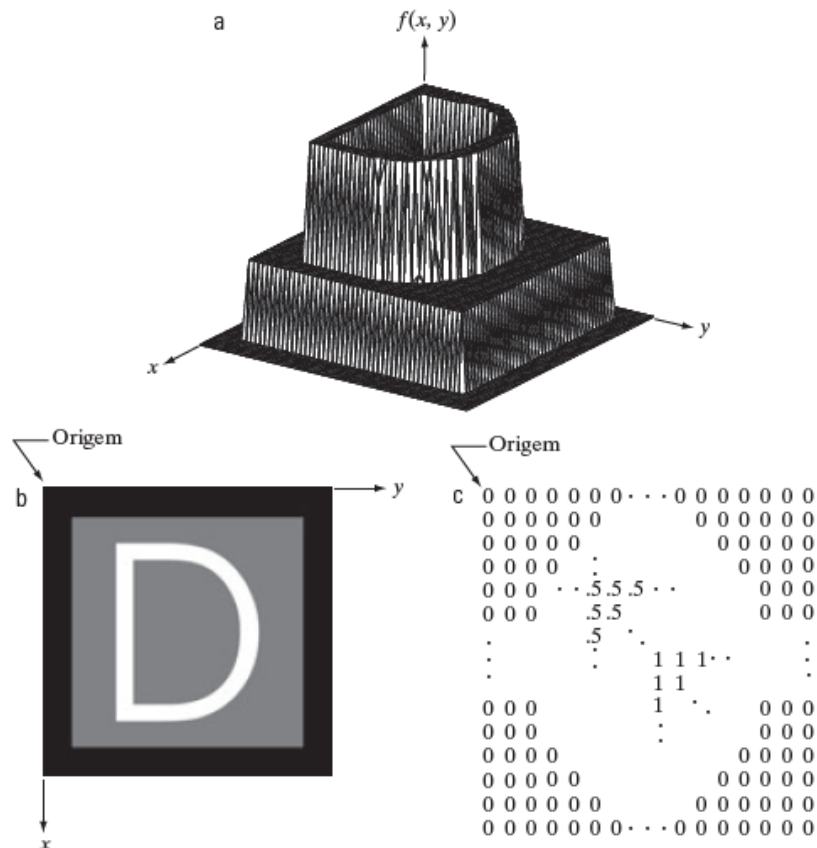
Ainda de acordo com Gonzalez e Woods (2010), existem três formas principais de representar uma imagem digital:

- representação gráfica tridimensional: Essa forma utiliza três eixos para descrever a imagem:  $x$  e  $y$ , que determinam as coordenadas espaciais, e um eixo  $z$ , que representa os valores de intensidade luminosa  $f(x, y)$ . Essa representação assemelha-se a uma superfície tridimensional, sendo útil para visualizações analíticas detalhadas de pequenas regiões;

- b) representação visual em escala de cinza: Nessa abordagem, a intensidade de cada ponto  $f(x, y)$  é representada diretamente como tons de cinza visíveis em um monitor ou impressão. Os valores de intensidade podem ser normalizados, geralmente no intervalo  $[0, 1]$ , e convertidos em tonalidades específicas como preto (0), cinza intermediário (0,5) e branco (1). Essa forma de representação é a mais comum e intuitiva, pois permite a visualização direta da imagem como percebida pelo olho humano;
- c) representação matricial: A matriz bidimensional organiza os valores de intensidade  $f(x, y)$  de forma numérica, onde cada elemento corresponde a um *pixel* da imagem. Por exemplo, uma matriz  $M \times N$  descreve a imagem como um conjunto de  $M$  linhas e  $N$  colunas. Essa forma é amplamente utilizada no desenvolvimento e implementação de algoritmos de processamento de imagens.

A Figura 5 exemplifica as três formas principais de representar uma imagem digital.

Figura 5 – (a) Imagem representada graficamente como uma superfície. (b) Imagem representada como uma matriz de intensidade visual. (c) Imagem representada como uma matriz numérica 2-D



Fonte: (GONZALEZ; WOODS, 2010)

### 2.2.3 Resolução espacial e de intensidade

A resolução espacial refere-se ao menor detalhe discernível em uma imagem e é expressa em unidades como pares de linha ou pixels por unidade de distância. Para ser significativa, a resolução deve considerar as dimensões espaciais da imagem e não apenas seu tamanho em *pixels*. Por exemplo, uma imagem de  $1.024 \times 1.024$  *pixels* tem pouco sentido sem especificar suas dimensões físicas. A resolução espacial de dispositivos, como câmeras digitais, influencia diretamente na capacidade de distinguir detalhes. Por outro lado, a resolução de intensidade está relacionada à menor variação discernível nos níveis de intensidade da imagem. Normalmente, a intensidade é quantizada em  $2^k$  níveis, onde  $k$  é o número de bits. A resolução mais comum é de 8 bits (256 níveis de intensidade), embora 16 bits sejam usados em aplicações específicas. Níveis mais baixos, como 4 ou 2 bits, podem resultar em perda de detalhes e surgimento de efeitos indesejados, como falso contorno (GONZALEZ; WOODS, 2010).

A Figura 6 exemplifica que, mantendo o tamanho da imagem constante ( $249 \times 199$ ) e reduzindo o número de níveis de cinza ( $=2^m$ ) produz contornos falsos (PETROU, 2010).

Figura 6 – Falso contorno gerado pela redução dos níveis de cinza

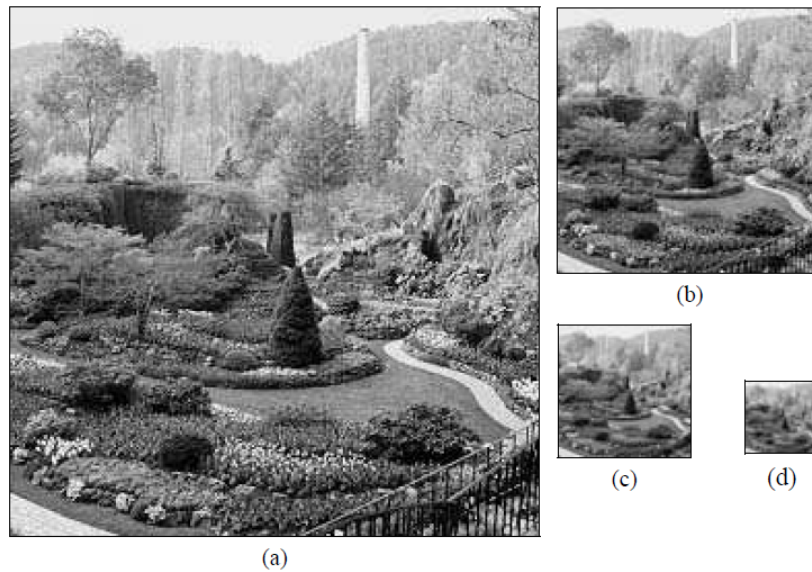


Fonte: (PETROU, 2010)

Filho e Neto (1999), destacam que a resolução espacial é essencial para a qualidade

da imagem digital, pois afeta diretamente a nitidez e a capacidade de capturar pequenos detalhes. Eles enfatizam a importância de escolher uma resolução adequada para cada aplicação específica, balanceando a necessidade de detalhes com as limitações de armazenamento e processamento. A Figura 7(a) mostra uma imagem de 256 x 256 *pixels*, com 256 níveis de cinza. Mantendo constante o número de tons de cinza, as figuras 7(b)-(c)-(d) mostram os resultados da redução espacial de  $N = 256$  para  $N = 128$ , 64 e 32, respectivamente.

Figura 7 – Efeito da resolução espacial na qualidade da imagem



Fonte: (FILHO; NETO, 1999)

### 2.3 Classificador Haar

Embora a motivação inicial tenha sido resolver o problema de detecção facial, esse classificador pode ser treinado para identificar uma ampla gama de objetos (VIOLA; JONES, 2004).

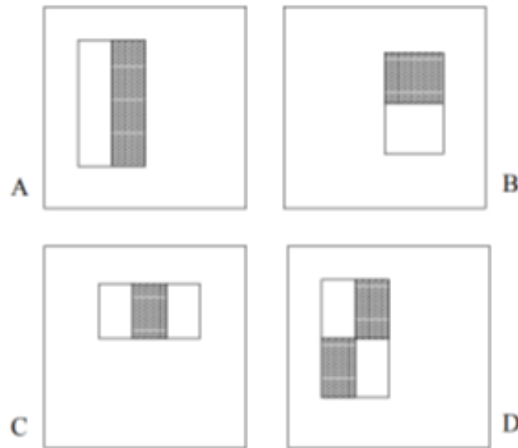
No classificador proposto por Viola e Jones (2004), são utilizados três formatos de *features* retangulares, baseados nos estudos de Papageorgiou *et al.* (2002), que introduziram o uso de "características Haar" no treinamento de classificadores, capazes de formar quatro padrões distintos:

- a) dois retângulos: O valor da característica é obtido pela diferença entre a soma dos *pixels* de duas regiões retangulares;
- b) três retângulos: O valor da característica é calculado como a soma dos *pixels* de um retângulo central subtraída da soma dos *pixels* de dois retângulos externos;

- c) quatro retângulos: O valor da característica é a diferença entre as somas dos *pixels* de cada par de retângulos dispostos diagonalmente.

Esses quatro padrões possíveis, como ilustra a Figura 8, onde as características A e B representam o formato de "dois retângulos", a característica C corresponde ao formato de "três retângulos" e a característica D ao formato de "quatro retângulos".

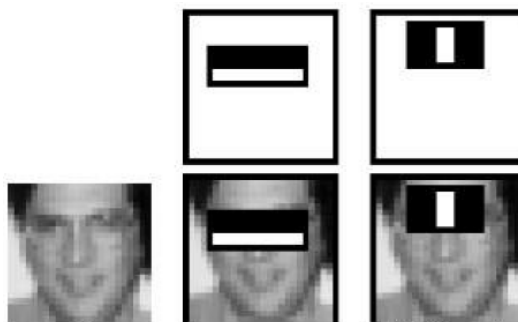
Figura 8 – Padrões possíveis



Fonte: (VIOLA; JONES, 2004)

O processo de detecção de Viola e Jones utiliza, como principal função, a *detectMultiScale*, responsável por detectar objetos, incluindo faces. O algoritmo realiza uma análise da imagem em busca de padrões que caracterizem um objeto. Quando se trata de uma face, a função utiliza como parâmetro as características específicas desse objeto, como a região dos olhos. Essas regiões precisam apresentar características que as distingam de outras, como os olhos, por exemplo, que geralmente estão em uma posição do rosto mais escura do que as bochechas. Esses tipos de características são conhecidos como recursos *Haar*, conforme demonstra a Figura 9 (OPENCV.ORG, 2024).

Figura 9 – Exemplo Aplicando recursos de Haar sobre uma imagem facial



Fonte: (OPENCV.ORG, 2024)

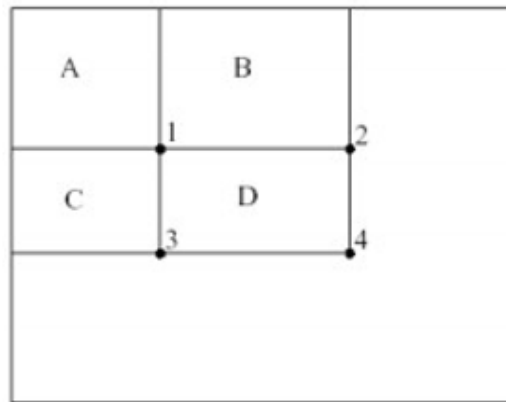
Para otimizar os cálculos dos valores dessas características, Viola e Jones (2004) propõem o uso de uma matriz conhecida como Imagem Integral, que funciona como uma representação intermediária da imagem original. Na Imagem Integral, cada ponto  $(x, y)$  contém a soma dos *pixels* localizados acima e à esquerda de  $(x, y)$ , incluindo esses próprios *pixels*. De acordo com a Equação (2.2).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

Usando a Imagem Integral, é possível calcular rapidamente a soma dos *pixels* de qualquer retângulo DDD, como apresenta a Figura 10, utilizando a Equação (2.3).

$$\sum Dpixels = ii(4) + ii(1) - (ii(2) + ii(3)) \quad (2.2)$$

Figura 10 – Retângulo D de posição arbitrária



Fonte: (VIOLA; JONES, 2004)

A classificação de uma *feature* dentro de uma imagem se dá pela classificação pelos chamados classificadores fracos. Representados pela função representada na Equação (2.4):

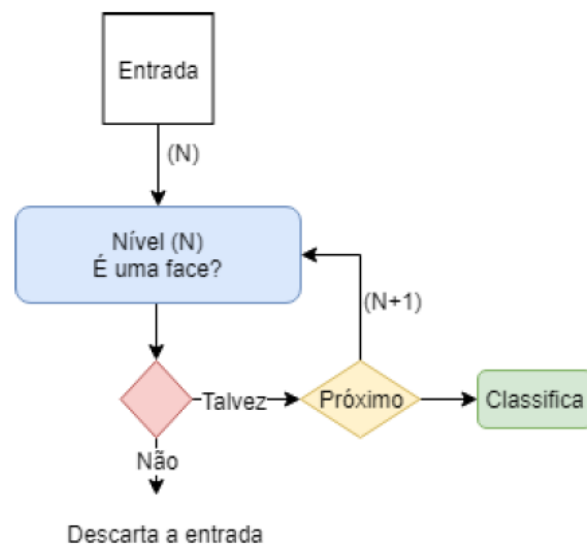
$$h(w, p, f, \theta) = \begin{cases} 1, & \text{se } pf(w) < p\theta \\ 0, & \text{se } pf(w) \geq p\theta \end{cases} \quad (2.3)$$

onde,  $w$  é a sub-janela de  $24 \times 24$  *pixels*;  $f$  é o número de *features*;  $p$  é a intensidade do pixel;  $\theta$  é limite (*threshold*) de distância para estimar a verossimilhança.

Para a classificação, o algoritmo de Viola e Jones (2004) combina classificadores fracos para formar classificadores fortes. Em seguida, constrói uma cascata de classificadores,

onde cada nível utiliza um conjunto distinto de classificadores fortes. A Figura 11 ilustra o fluxograma deste processo. Para aumentar a eficiência, o algoritmo emprega o *AdaBoost* para treinar os classificadores fracos. O *AdaBoost* combina classificadores baseados em *boosting*, seleção de características relevantes e aprendizado de classificação. Isso permite identificar características mais adequadas e melhorar a precisão da classificação.

Figura 11 – fluxograma de funcionamento da cascata dos classificadores



Fonte: (COSTA, 2019)

## 2.4 Eigenfaces

Turk e Pentland (1991) desenvolveram o algoritmo *Eigenfaces*, pioneiro em reconhecimento facial utilizando Análise de Componentes Principais (PCA). Este método estatístico multivariado compara características faciais através de projeções derivadas da redução dimensional do espaço de faces. O algoritmo gera uma matriz de transformação para imagens não correlacionadas, permitindo reconhecimento facial a partir do banco de imagens (MEDEIROS *et al.*, 2012).

O algoritmo *Eigenfaces*, empregado no reconhecimento facial, opera em duas fases principais: aprendizagem e reconhecimento. Durante a fase de aprendizagem, o *Eigenfaces* utiliza um conjunto de imagens de treinamento para construir um modelo facial. Isso é feito por meio da técnica de Análise de Componentes Principais (PCA - *Principal Component Analysis*), que transforma as informações visuais em vetores. Esses vetores são então usados para detectar padrões, identificar objetos e reconhecer faces humanas. As informações visuais processadas

pelo algoritmo *Eigenfaces* incluem os pixels das imagens faciais, representados como vetores numéricos que indicam a intensidade da luz. A técnica PCA reduz a dimensionalidade ao focar em características principais, como a forma do rosto e a posição dos olhos, ignorando detalhes menos relevantes, como variações de iluminação. Esses dados são usados para detectar padrões e identificar rostos no processo de reconhecimento facial (TURK; PENTLAND, 1991).

O método *Eigenfaces* é utilizado para extrair informações importantes de uma imagem facial, configurando-a da maneira mais eficiente possível e comparando a similaridade da face com uma base de modelos pré-treinados. Matematicamente, o *Eigenfaces* busca identificar os componentes principais das faces, utilizando os autovetores da matriz de covariância do conjunto de imagens, tratando a imagem como um vetor. A dimensão de uma imagem é determinada pelo número de *pixels* que ela contém. O PCA filtra a imagem, armazenando apenas as características relevantes para o reconhecimento e eliminando *pixels* com ruídos, como variações de brilho e distorções causadas pela compactação da imagem. Os *pixels* úteis são armazenados, formando assim o subespaço da imagem (TURK; PENTLAND, 1991).

O nome *eigenfaces* é atribuído aos autovetores (*eigenvectors*) da matriz de covariância das imagens das faces do banco de faces de treinamento por possuírem aspectos de faces. Eles são extraídos de imagens de treinamento e formam uma face média. Combinando esses *eigenfaces* através de multiplicações escalares, é possível representar diversas faces, demonstrando a eficácia da técnica em reconhecimento facial. Na Figura 12 é possível ver as *eigenfaces* extraídas das faces do banco de dados Yale, bem como a face média que será a base para a reconstrução das diversas faces (DAVE *et al.*, 2010).

Figura 12 – Face Média e Eigenfaces



Fonte: (DAVE *et al.*, 2010)

Matematicamente, o algoritmo *Eigenfaces* utiliza a Análise de Componentes Prin-

cipais (PCA) para extrair as principais características de um conjunto de imagens, gerando uma "face média" por meio da combinação linear de diversas *Eigenfaces*. Em uma imagem de dimensão  $N \times N$ , o algoritmo transforma a imagem em um vetor unidimensional de dimensão  $N^2$  e busca calcular os autovetores da matriz de covariância associada ao conjunto de imagens. À medida que aumenta o número de imagens de um mesmo indivíduo, a precisão do processo também melhora, uma vez que mais informações são incorporadas nos autovetores (*Eigenvectors*) (TURK; PENTLAND, 1991).

Para a geração da face média, no processo de treinamento, considera-se a Equação (2.5):

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (2.4)$$

Considerando  $M$  o conjunto de imagens faciais na base de treinamento supervisionada, e  $\Gamma$  como vetor da imagem analisada e  $\Gamma_i$  é a  $i$ -ésima face, tem como resultado um vetor médio  $\Psi$ , que representa a "face média" das imagens utilizadas para treinamento do algoritmo, como na Figura 13. Após o cálculo da face média na Equação (2.6), é gerado um novo vetor  $\Phi$ , subtraindo  $\Psi$  de cada imagem  $\Gamma$ , como mostra a Equação (2.6) (TURK; PENTLAND, 1991).

$$\Phi = \Gamma - \Psi \quad (2.5)$$

Figura 13 – Subtração da face média



A face da esquerda é uma imagem de treinamento individual, enquanto que a face da direita representa uma média calculada a partir de varias imagens de rosto do conjunto de dados.

Realiza-se o cálculo da matriz de covariância (C) Equação (2.7), pois a dimensionalidade de  $M^1$  é maior que a dimensionalidade de  $\phi$ .

$$C = \phi T \phi \quad (2.6)$$

Calcula-se os autovetores  $v$  e autovalores  $\lambda$  da matriz C, e em sequência cria-se a matriz de transformação utilizando a Equação (2.8).

$$\mu = v\Phi \quad (2.7)$$

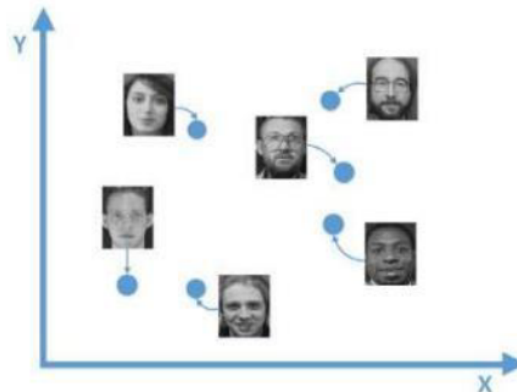
A matriz  $\mu$  conterá M-1 autovetores significativos. Entretanto, caso M-1 ainda seja muito grande, é possível realizar a eliminação de alguns desses autovetores pela ordem de importância de seus correspondentes autovalores  $\lambda$ , gerando  $M'$  autovetores selecionados. Para finalizar o cálculo de treinamento, é realizada a normalização de  $M'$  vetores da matriz  $\mu$ . Para a realização do reconhecimento, é feito o seguinte: dada uma imagem de teste Y, projeta-se a imagem junto ao autoespaço, centralizando-a com a face média gerada no treinamento, então é realizada a classificação usando classificadores definidos, ou N classificadores combinados. A face é classificada calculando a distância da imagem matriz  $\mu Y$  de teste com os valores da matriz  $\mu$  de treinamento, quando a distância é mínima ou abaixo do limiar predefinido. Caso contrário, a classe é classificada como não conhecida (TURK; PENTLAND, 1991).

Este método permite a classificação de faces calculando a distância entre a imagem analisada e sua projeção no novo espaço. Se essa distância estiver dentro de um limite (*threshold*), a imagem é considerada uma face; caso contrário, é considerada não face. O método *Eigenfaces* é interessante porque, além da classificação, possibilita a reconstrução e compactação de imagens faciais. Para calcular a distância, são escolhidas as técnicas que melhor se adaptam ao banco de imagens utilizado, sendo a mais comum a distância euclidiana aplicada ao algoritmo KNN. Na Figura 14, temos a representação de um espaço de faces no plano cartesiano (BISSI *et al.*, 2018).

---

<sup>1</sup> A dimensionalidade de M refere-se ao número de imagens faciais presentes na base de dados de treinamento. Cada imagem facial é representada como um vetor de pixels, e a dimensionalidade de M está diretamente relacionada ao número de imagens utilizadas para o treinamento do modelo (TURK; PENTLAND, 1991).

Figura 14 – Representação de um espaço de faces no plano cartesiano



Fonte: (SILVA, 2016)

## 2.5 Fisherfaces

O algoritmo *Fisherfaces*, proposto por Belhumeur *et al.* (1997), visa minimizar a variância intraclases e maximizar a variância interclases. Essa abordagem é especialmente útil para mitigar distorções causadas pela sensibilidade à iluminação em expressões faciais, superando limitações do método PCA Avançado (PRADO, 2018).

O método *Fisherfaces* utiliza a Análise de Discriminante Linear (LDA) para ser insensível a grandes variações de iluminação e expressões faciais. É eficaz na identificação e reconhecimento de rostos, mesmo em condições de iluminação diferentes das usadas no treinamento dos modelos. Ele trabalha com rótulos agrupados por pessoas, onde cada agrupamento é conhecido como classe. O método tenta modelar a dispersão dos pontos para obter maior confiabilidade na classificação. A LDA busca otimizar a redução da melhor linha de uma superfície que separa as classes de forma satisfatória (BELHUMEUR *et al.*, 1997).

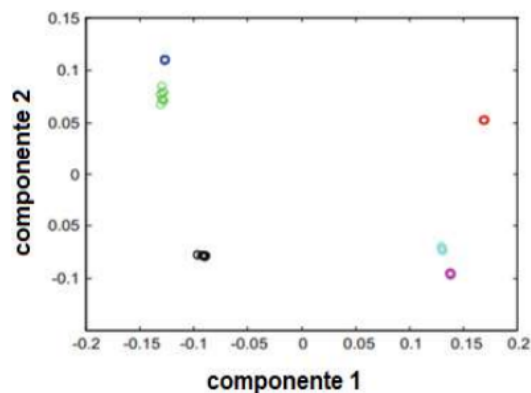
A LDA foi proposta como uma abordagem superior à Análise de Componentes Principais (PCA) para separação de classes. Enquanto a PCA se concentra na máxima variância global, a LDA busca direções que maximizem a separação entre classes e minimizem a dispersão intraclases (KITANI, 2007; PRADO, 2018). Ambas as técnicas visam reduzir a dimensionalidade dos dados (JUNIOR; MARTINS, 2015).

Inicialmente, o método *Fisherfaces* projeta o conjunto de imagens em um espaço de baixa dimensionalidade utilizando a técnica PCA. Em seguida, aplica a técnica LDA para projetar as imagens faciais em um espaço de dimensionalidade ainda menor. Dessa forma, o reconhecimento facial é realizado através de um subespaço de seus próprios vetores, utilizando uma medida de similaridade para comparar com as outras faces projetadas nesse subespaço

(FIGUEREDO, 2011).

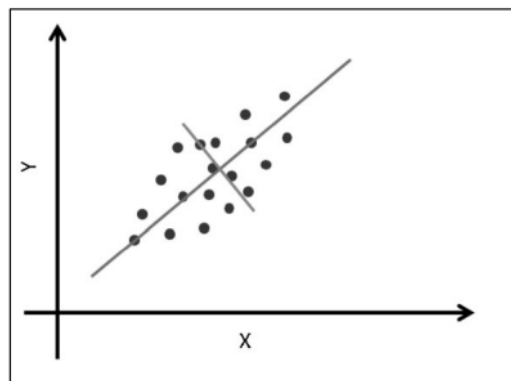
A Figura 15 apresenta a distribuição das faces no espaço projetado pela Análise Discriminante Linear (LDA), onde cores iguais indicam faces da mesma classe. Conforme esperado, a LDA promove uma separação eficaz entre classes distintas e uma agregação das faces dentro da mesma classe. Isso contrasta com a PCA, que prioriza a máxima variância global, Figura 16, enquanto a LDA otimiza a variância interclasses e minimiza a variância intraclasses, Figura 17.

Figura 15 – Projeção do espaço com a LDA



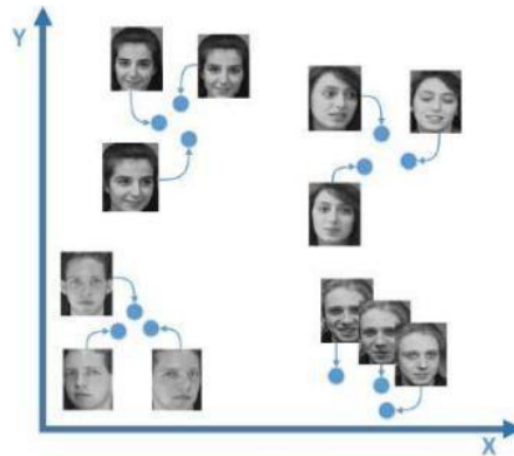
Fonte: (GIRON-SIERRA, 2016)

Figura 16 – Redução da dimensionalidade com a PCA



Fonte: (MENDES, 2019)

Figura 17 – Projeção das imagens no subespaço



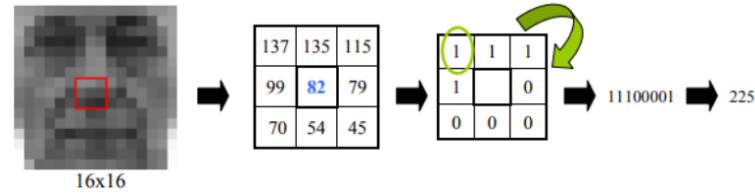
Fonte: (SILVA, 2016)

Em suma, o método *Fisherfaces* aplica a técnica PCA para reduzir as dimensões, projetando as imagens faciais originais em um espaço de menor dimensão. Em seguida, utiliza a técnica LDA para identificar a melhor característica discriminante linear dentro do subespaço gerado pelo PCA. Dessa forma, é possível classificar uma nova imagem ao comparar sua similaridade com as imagens faciais do conjunto pré-treinado, determinando assim a quem pertence a face detectada. Portanto, quanto mais distintas forem duas pessoas, mais afastadas estarão suas projeções. Da mesma forma, quanto mais semelhantes forem as imagens de uma pessoa, mais próximas estarão suas projeções (BISSI *et al.*, 2018).

## 2.6 Local Binary Patterns Histograms(LBPH)

O LBPH, de início, foi desenvolvido para a análise de texturas, mas ao longo do tempo, passou a ser utilizado para a extração de características no reconhecimento e classificação de imagens faciais. O LBP analisa os *pixels* de uma imagem utilizando uma matriz, geralmente de 3 x 3, onde o *pixel* central serve como limiar para determinar o valor dos *pixels* vizinhos, que são convertidos em valores binários (AMARAL; THOMAZ, 2013). A Figura 18 ilustra o procedimento.

Figura 18 – Determinando os *pixels* vizinhos à partir do *pixel* central



Fonte: (LÓPEZ, 2010)

Dessa forma, uma matriz com valores binários resultantes da comparação: valor binário 1 nos vizinhos que têm valor superior ou igual ao *pixel* central e 0. A limiarização do descritor LBP =  $E_1, E_2, \dots, E_8$  é obtida pela Equação (2.8) (AMARAL; THOMAZ, 2013).

$$E_i = \begin{cases} 0, & \text{se } V_i < V_0 \\ 1, & \text{se } V_i \geq V_0 \end{cases} \quad (2.8)$$

Onde:

- $E_i$ : valor binário (0 ou 1) atribuído ao *pixel* vizinho na posição  $i$ ;
- $V_i$ : intensidade do *pixel* vizinho;
- $V_0$ : intensidade do *pixel* central (limiar).

Os valores dos *pixels* na vizinhança linearizada são multiplicados pelos pesos dados como *pixels* correspondentes. Feita esta etapa, realiza-se a soma dos valores, resultando em um valor do código LBP, como representado na Equação (2.9) (AMARAL; THOMAZ, 2013).

$$N_{LBP} = \sum_{i=1}^0 E_i \cdot 2^{i-1} \quad (2.9)$$

Depois, cria-se um histograma para calcular as ocorrências dos diferentes códigos LBP de todos os *pixels* da imagem. Isso forma o vetor de características LBP, que descreve a textura da imagem. A principal diferença entre TU e LBP é que o LBP possui apenas 256 códigos resultantes. O operador LBP é resistente a transformações monótonas em escala de cinza (AMARAL; THOMAZ, 2013). Podemos observar isso facilmente ao olhar para uma imagem artificialmente modificada, como mostrado na Figura 19.

Dessa forma, podemos integrar informações espaciais no modelo de reconhecimento facial. Através dessa abordagem, a região facial é dividida em sub-regiões. Em cada sub-região, aplica-se o operador LBP (Padrões Binários Uniformes), gerando um histograma que contém os padrões de textura (AHONEN *et al.*, 2006).

A etapa final consiste em calcular um histograma a partir da matriz LBP resultante, que utiliza uma vizinhança de  $3 \times 3$ , há  $2^8 = 256$  padrões possíveis. Assim, a matriz LBP

Figura 19 – Transformação LBP

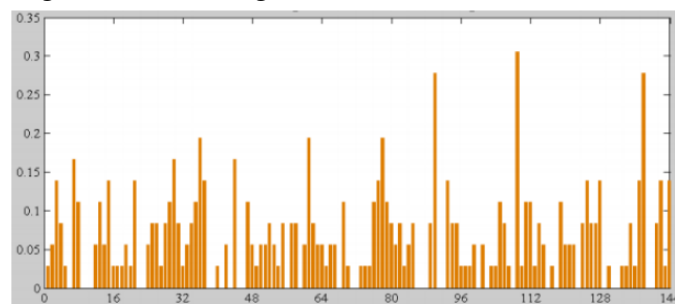


Fonte: (PRADO, 2018)

2D apresenta valores mínimos de 0 e máximos de 255, o que possibilita a construção de um histograma com 256 intervalos (bins) correspondentes aos códigos LBP. Este histograma serve como nosso vetor de características final (ROSEBROCK, 2015).

A Figura 20 ilustra um exemplo de histograma que registra a frequência de ocorrência de cada padrão LBP. Esse histograma pode ser interpretado como um vetor de características. Um histograma é uma representação gráfica de dados organizados em classes, onde a altura de cada barra indica o número de observações pertencentes a uma determinada classe. No histograma do LBPH, as classes correspondem aos possíveis valores de um *pixel* (0 a 255), e a altura de cada barra representa a frequência com que cada valor de *pixel* ocorre (LÓPEZ, 2010).

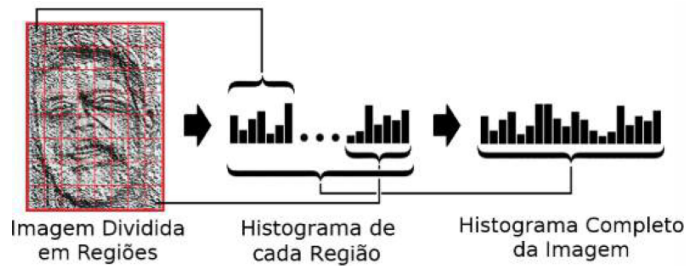
Figura 20 – Histograma de Padrões Binários Locais



Fonte: (LÓPEZ, 2010)

Para cada *pixel* de uma imagem, é realizado o cálculo dos seus autovalores de LBP. Esses autovalores compõem o espectro de características do LBP. O algoritmo LBPH utiliza o histograma desse espectro característico como vetor de características para a classificação. A comparação é feita entre o histograma gerado a partir da imagem de teste e os histogramas das imagens presentes na base de treinamento (PRADO, 2018). A Figura 21 ilustra um exemplo de extração de características por meio de histogramas.

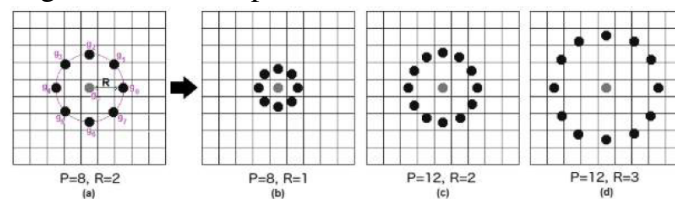
Figura 21 – Exemplo da extração do histograma da imagem



Fonte: (PRADO, 2018)

O LBPH é um processo capaz de representar  $N$  histogramas a partir de imagens de alta dimensionalidade, reduzindo essas características em vetores de características de menor dimensão. Ele permite operar com diferentes tamanhos de vizinhança, incluindo vizinhanças circulares, ajustando o raio e o número de vizinhos, o que aprimora o operador LBP (AHONEN *et al.*, 2006). A Figura 22 apresenta um exemplo do LBP com vizinhança circular.

Figura 22 – Exemplo de LBP circular



Fonte: (PRADO, 2018)

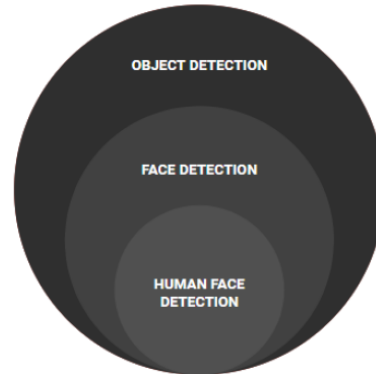
Dessa maneira, é possível distinguir as particularidades desses três algoritmos. Tanto o *Eigenfaces* quanto o *Fisherfaces* compartilham semelhanças, pois operam sobre as dimensões e vetores da imagem, focando na identificação dos principais pontos faciais durante o processo de reconhecimento. Já o LBPH utiliza matrizes e a análise da vizinhança.

## 2.7 Detecção de faces

Nos últimos vinte anos, a detecção de faces em imagens tornou-se um tópico amplamente pesquisado no campo da visão computacional (SURVE *et al.*, 2020; RANJAN *et al.*, 2019). De maneira geral, a detecção de objetos consiste em identificar instâncias de classes específicas, como pessoas, carros, edifícios ou faces, em imagens ou vídeos (FENG *et al.*, 2022). Nesse contexto, a detecção de faces é considerada um subdomínio da detecção de objetos, permitindo identificar faces entre outros elementos presentes em uma cena, como ilustra a Figura 23. Já a detecção de faces humanas foca na distinção entre faces de pessoas e de outros seres, como

animais (REINOSO; BRESSAN, 2024).

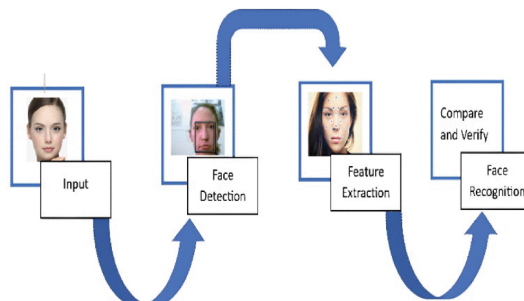
Figura 23 – Detecção de face subdomínio da detecção de objetos



Fonte: (REINOSO; BRESSAN, 2024)

Segundo Gururaj *et al.* (2024), um sistema de reconhecimento facial é composto por três etapas fundamentais: detecção facial, extração de características e reconhecimento facial. Na etapa de detecção facial, o sistema identifica e localiza rostos humanos nas imagens imediatamente após sua captura. Na etapa de extração de características, são gerados vetores de características únicos para cada rosto detectado, capturando elementos distintivos de sua composição. Por fim, na etapa de reconhecimento facial, o sistema compara as características extraídas com um banco de dados de rostos previamente registrados, identificando correspondências precisas com os modelos armazenados. Esta última etapa será abordada com mais detalhes na Seção 2.8. A Figura 24 apresenta o fluxo de trabalho de um sistema de reconhecimento facial.

Figura 24 – Etapas padrão de detecção e reconhecimento facial

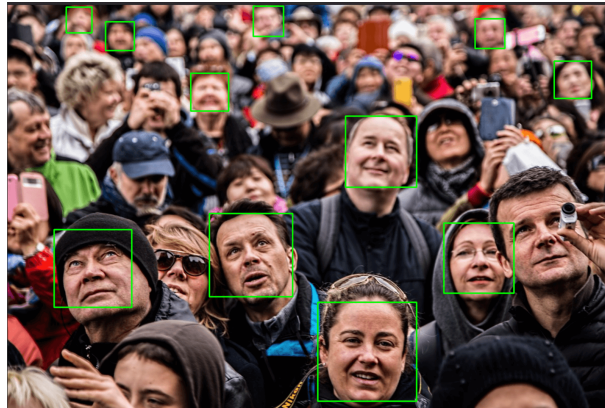


Fonte: (GURURAJ *et al.*, 2024)

De acordo com Ho *et al.* (2024), o principal objetivo de um detector facial é localizar e identificar rostos em imagens e, quando encontrados, retornar as dimensões (caixas delimitadoras, *bounding boxes*) de cada rosto. Realizar essa tarefa continua sendo um desafio em ambientes

não controlados. Diversos fatores, como variações de escala, ausência de componentes estruturais (oclusão), mudanças significativas de pose, expressões faciais, condições de iluminação variadas, baixa resolução das imagens, entre outros, podem comprometer o desempenho dos detectores faciais e de outros sistemas de análise facial, conforme ilustra a Figura 25 (HO *et al.*, 2024; RANJAN *et al.*, 2019).

Figura 25 – Desafios da detecção facial



Fonte: ( Internet )

Na literatura, diversas técnicas podem ser empregadas para detectar e localizar rostos humanos, incluindo a técnica Viola-Jones (VJ), o método do Histograma de Gradientes Orientados (HOG) e a Análise de Componentes Principais (PCA), entre outras.

O algoritmo Viola-Jones destaca-se por sua eficiência em imagens frontais, sendo frequentemente utilizado em aplicações em tempo real. Baseado em características Haar, a técnica analisa as imagens para identificar elementos-chave, como bordas, cantos e linhas, que são fundamentais para o reconhecimento facial. Além disso, algumas abordagens integram informações baseadas em cores para aprimorar a precisão do processo. A combinação dessas técnicas fortalece a identificação de rostos e contribui significativamente para a robustez dos sistemas de RF (NAPOLÉON; ALFALOU, 2017; GURURAJ *et al.*, 2024).

O algoritmo VJ é o detector de faces mais icônico e conhecido. Ele é referência até os dias atuais para a detecção facial em tempo real e pode ser implementado utilizando a biblioteca *OpenCV* em *Python*. Em questão de segundos, é capaz de detectar um rosto presente em uma imagem utilizando as características Haar e os recursos de aprendizado de máquina *AdaBoost*. Esse algoritmo apresenta uma taxa de detecção muito alta e uma taxa de falsos positivos muito baixa (SRIMAN *et al.*, 2021; REINOSO; BRESSAN, 2024). A Figura 26 apresenta o fluxo de trabalho do algoritmo VJ.

O algoritmo HOG é um descritor de características usado principalmente para

Figura 26 – Fluxograma do algoritmo de Viola Jones



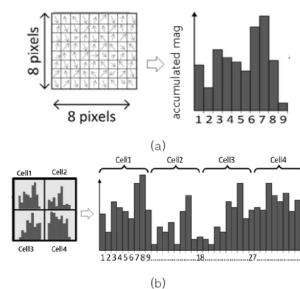
Fonte: (SRIMAN *et al.*, 2021)

detectar objetos, com destaque para a detecção de pessoas. Ele analisa a distribuição das direções das bordas ou dos gradientes de intensidade em blocos de *pixels*, representando a direção e a magnitude das variações. Os histogramas locais desses blocos são concatenados para formar um descritor. A normalização do contraste em blocos maiores melhora a precisão, enquanto a padronização ajuda a minimizar os efeitos de variações de iluminação e sombras, aumentando a eficácia da detecção (SRIMAN *et al.*, 2021).

De acordo com Jang (2021), as características HOG, que são compostas por histogramas, são robustas à variação geométrica e insensíveis aos efeitos da iluminação. No entanto, são muito sensíveis às transformações rotacionais porque só são utilizadas características locais para comparação. O HOG segmenta a imagem em unidades mais pequenas chamadas células e extrai características locais. Normalmente, cada célula é constituída por  $8 \times 8$  *pixels* e um bloco é constituído por  $2 \times 2$  células.

Jang (2021), argumenta que para cada célula, cria-se um histograma somando as magnitudes dos gradientes em seus bins de direção, como ilustra a Figura 27(a). Um bloco de  $2 \times 2$  células combina os histogramas dessas células em um único vetor, como ilustra a Figura 36(b). Esse bloco desliza pela imagem inteira em incrementos iguais ao tamanho de uma célula, e os histogramas de todos os blocos são unidos para formar o vetor final de características HOG.

Figura 27 – Caraterística HOG



Fonte: (JANG, 2021)

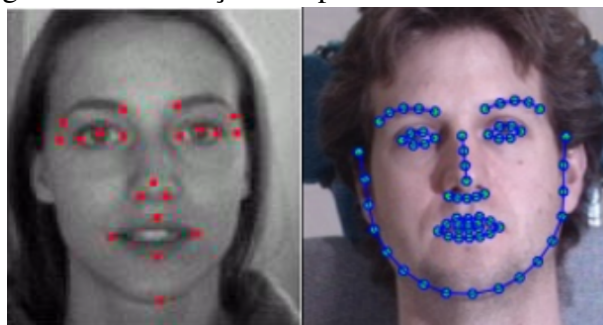
A Análise de Componentes Principais (PCA) foi inicialmente proposta por Karl Pearson em 1901 e posteriormente desenvolvida e nomeada por Harold Hotelling em 1933. Dependendo do campo de aplicação, a PCA é conhecida por diferentes denominações, como Decomposição Ortogonal Apropriada (POD) na engenharia mecânica e Transformada de Karhunen-

Loève (KLT) no processamento de sinais, entre outras. Turk e Pentland foram os primeiros a implementar a PCA no contexto da detecção de rostos (HASAN *et al.*, 2021).

Hasan *et al.* (2021). argumenta que a PCA reduz uma grande quantidade de dados ao capturar sua essência em poucas dimensões representativas. Esse processo matemático aplica uma transformação ortogonal para converter um conjunto de  $M$  variáveis possivelmente correlacionadas em  $K$  variáveis não correlacionadas, chamadas de componentes principais. Essas componentes são derivadas do conjunto de treinamento, com apenas as mais significativas sendo selecionadas, enquanto as demais são descartadas. As componentes principais resultantes também são conhecidas como "faces próprias". A detecção de um rosto é realizada projetando uma imagem de teste em um subespaço definido pelos *eigenspaces*.

Ainda na etapa de detecção, realiza-se o alinhamento facial, também conhecido como detecção de pontos de referência do rosto. Essa etapa é essencial para alinhar o rosto frontalmente, aumentando a precisão no reconhecimento facial. Os principais pontos de referência incluem olhos, nariz, boca, entre outros, cuja forma ou localização confere um padrão único a cada rosto. A escolha do modelo de pontos de referência depende do contexto de utilização, o modelo de 68 pontos é ideal para aplicações que exigem alta precisão, por exemplo: biometria, análise de expressões. Enquanto que o modelo de 20 pontos é mais leve, adequado para sistemas embarcados ou em tempo real onde a velocidade é prioridade (ex.: câmeras de vigilância) (SARAGIH *et al.*, 2022). A Figura 28 ilustra a detecção desses pontos de referência: o rosto à esquerda apresenta 20 pontos detectados, enquanto o rosto à direita exibe 68 pontos de referência.

Figura 28 – Detecção dos pontos de referência



Fonte: Adaptado de (SARAGIH *et al.*, 2022)

### 2.7.1 *Extração das características*

Nesta etapa, o objetivo é realizar a extração de características das imagens dos rostos detectados, utilizando métodos globais ou locais. No método global, considera-se o rosto como um todo, enquanto no método local a ênfase recai sobre características internas ou regiões faciais específicas de interesse. Os métodos locais são amplamente preferidos, pois capturam informações detalhadas de áreas específicas do rosto e são menos afetados por fatores como geometria facial, envelhecimento, variações de pose e rotação. Além disso, os métodos de extração de características podem ser organizados em três categorias principais: Métodos Genéricos, Métodos Baseados em Modelos de Características, Métodos de Correspondência Estrutural (PAYAL; GOYANI, 2020; GURURAJ *et al.*, 2024).

Métodos Genéricos baseiam-se na identificação de bordas, linhas e curvas em imagens faciais, utilizando medidas como tamanho e distância para diferenciar rostos de forma precisa. Diversas técnicas são amplamente empregadas na extração de características faciais, incluindo: HOG, *Eigenfaces*, Análise de Componentes Independentes (ICA), Transformada de Recursos Invariantes à Escala (SIFT), Filtros de Gabor, Quantização de Fase Local (LPQ) e Análise Discriminante Linear (LDA). Essas abordagens desempenham um papel fundamental na caracterização e reconhecimento de atributos faciais, contribuindo significativamente para a eficácia de sistemas de reconhecimento facial (GURURAJ *et al.*, 2024).

Os métodos baseados em modelos de características representam uma abordagem para o reconhecimento facial, que se concentra na identificação de elementos faciais específicos, como os olhos, através da utilização de modelos predefinidos. Neste contexto, um rosto é abstraído como um conjunto de atributos, denominado "assinatura", que descreve as características faciais primárias, incluindo a boca, o nariz e os olhos, bem como sua distribuição geométrica dentro da imagem. A identificação de cada indivíduo é realizada com base em sua estrutura facial única, que compreende o tamanho, a forma e a disposição relativa dos elementos faciais (KORTLI *et al.*, 2018; GURURAJ *et al.*, 2024).

Métodos de Correspondência Estrutural essas abordagens levam em conta as restrições geométricas das características faciais, assegurando a correspondência com padrões estruturais definidos. Técnicas como correspondência de grafos elásticos, arquitetura de ligação dinâmica e Padrão Binário Local (LBP) têm se destacado amplamente na literatura por seus resultados expressivos (HAMMOUCHE *et al.*, 2022; KARPAGAM *et al.*, 2023; GURURAJ *et al.*, 2024).

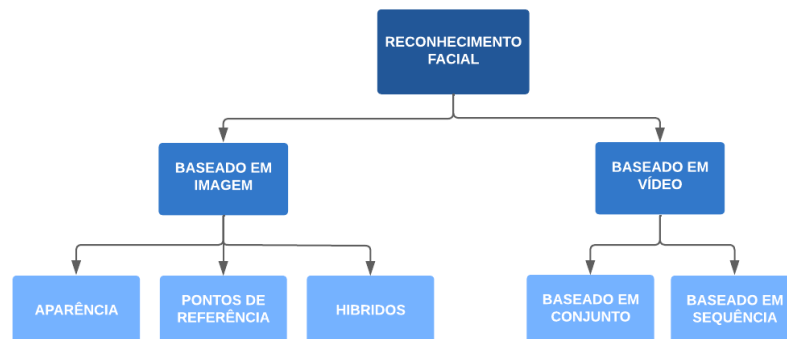
## 2.8 Reconhecimento facial

Esta é a etapa final, responsável por identificar as faces e realizar o reconhecimento facial automatizado. Para isso, é essencial a existência de um banco de dados facial, onde os dados relevantes são armazenados para comparação e identificação. Nesse processo, múltiplas imagens de cada indivíduo são capturadas, suas características distintas são extraídas e registradas no banco de dados. Quando uma imagem facial é submetida ao sistema, suas características extraídas são comparadas com as classes de rostos previamente armazenadas. Nessa etapa, tanto as características faciais extraídas anteriormente quanto os elementos faciais são analisados e comparados com os rostos conhecidos registrados no banco de dados (GURURAJ *et al.*, 2024).

Para Saragih *et al.* (2022), o reconhecimento facial por computador é uma técnica que permite a identificação automática de uma pessoa em uma imagem. Esse processo baseia-se na análise de características específicas que distinguem o rosto de um indivíduo dos demais.

No campo dos sistemas de reconhecimento facial, a literatura classifica esses sistemas em dois grupos principais: métodos baseados em imagens e métodos baseados em vídeos. Os sistemas baseados em imagens concentram-se no reconhecimento de indivíduos utilizando exclusivamente sua aparência física estática. Em contraste, os sistemas baseados em vídeos analisam não apenas características físicas, mas também alterações na aparência ao longo do tempo e movimentos faciais dinâmicos (SILVA *et al.*, 2023; GURURAJ *et al.*, 2024). A Figura 29 apresenta a classificação geral da literatura de RF, destacando a diferença fundamental entre essas duas abordagens.

Figura 29 – Classificação dos métodos de RF



Fonte: Elaborado pelo autor (2024)

### 2.8.1 Métodos baseados em imagens

De acordo com Gururaj *et al.* (2024), os métodos de reconhecimento facial baseados em imagem podem ser divididos em três categorias: métodos de aparência, métodos de pontos de referência e métodos híbridos.

#### 2.8.1.1 Métodos de Aparência

A abordagem de aparência, frequentemente chamada de método baseado em características globais, utiliza o rosto inteiro para a extração de características. Essa técnica busca reconhecer rostos com base em sua representação facial completa, em vez de focar em elementos individuais, como olhos, boca ou nariz (ADJABI *et al.*, 2020).

Normalmente, a imagem do rosto é representada como uma matriz de *pixels*, que é convertida em vetores de características para facilitar o processamento e a análise. Posteriormente, esses vetores são projetados em um espaço de menor dimensionalidade. No entanto, tanto as técnicas de aparência quanto as de subespaço são sensíveis a variações, como mudanças de expressão facial, iluminação e pose (TALAB *et al.*, 2020). Apesar dessa sensibilidade, suas vantagens têm garantido seu uso amplo em sistemas de reconhecimento facial. Essas abordagens podem ainda ser classificadas em técnicas lineares ou não lineares, dependendo do método empregado para representar o subespaço (JAYARAMAN *et al.*, 2020).

#### 2.8.1.2 Métodos de Pontos de Referência

No reconhecimento facial, os pontos de referência desempenham um papel fundamental na identificação e análise de características faciais. Esses pontos específicos, como os cantos dos olhos, a ponta do nariz e os cantos da boca, possuem significância anatômica e podem ser localizados com precisão no rosto de uma pessoa. Eles fornecem informações essenciais para os algoritmos de reconhecimento facial, permitindo uma análise detalhada e uma identificação mais precisa das características faciais (MOUSAVI *et al.*, 2021a).

No reconhecimento facial, a detecção de pontos de referência utiliza algoritmos para localizar e marcar pontos específicos no rosto de uma pessoa em imagens ou quadros de vídeo. Após a identificação desses pontos, eles são usados como base para extrair diversas características faciais, como a distância entre os olhos ou o ângulo da boca (MOUSAVI *et al.*, 2021b).

Segundo Jing *et al.* (2023), as características extraídas são empregadas para criar uma representação única do rosto de um indivíduo, frequentemente chamada de modelo facial ou assinatura facial. Um descritor facial em 3D pode ser desenvolvido combinando diversas medidas, como a largura da cabeça, altura e a largura do nariz, profundidade do nariz, distância entre os olhos e curvaturas faciais.

### 2.8.1.3 Métodos híbridos

As abordagens híbridas combinam características locais e globais para aproveitar os pontos fortes de ambas as técnicas. Esses métodos integram as duas abordagens em um único sistema, visando aprimorar o desempenho e a precisão no reconhecimento facial (GOEL *et al.*, 2020). A escolha da técnica mais adequada deve considerar a aplicação específica. Por exemplo, em sistemas que utilizam imagens muito pequenas, métodos baseados em características locais podem não ser a melhor opção. Outro fator importante na seleção do algoritmo é a quantidade de exemplos de treinamento necessária para alcançar resultados satisfatórios. Vale destacar que a tendência atual é o desenvolvimento de métodos híbridos, que unem as vantagens das abordagens locais e holísticas. No entanto, essas soluções geralmente apresentam maior complexidade e demandam mais tempo de processamento (KORTLI *et al.*, 2020).

## 2.8.2 Métodos baseados em vídeos

Em comparação com os métodos tradicionais de reconhecimento facial, o RF baseado em vídeo analisa dados de vídeo em vez de imagens estáticas. Assim como as técnicas convencionais, o RF baseado em vídeo tem como objetivo identificar indivíduos em um vídeo (identificação) ou verificar se dois sujeitos em vídeos diferentes possuem a mesma identidade (verificação), conforme argumenta Gururaj *et al.* (2024).

A popularidade dessa abordagem tem crescido devido à sua ampla gama de aplicações. Este campo atrai significativa atenção nas áreas de visão computacional e biometria, sendo utilizado em domínios como vigilância visual, controle de acesso e análise de conteúdo de vídeo. Pesquisadores e profissionais aproveitam esses avanços para melhorar a segurança, automatizar processos e obter informações valiosas a partir de dados visuais. Com a disseminação de câmeras em diversas localidades e o uso massivo de dispositivos móveis capazes de capturar vídeos, há um fluxo constante de grandes volumes de dados de vídeo. Essa disponibilidade tem acelerado a pesquisa e impulsionado avanços significativos nessa área (ZHENG *et al.*, 2020).

Na literatura, os métodos de reconhecimento facial baseados em vídeo são geralmente classificados em duas abordagens principais: métodos baseados em conjuntos e métodos baseados em sequências (GURURAJ *et al.*, 2024). Os detalhes dessas técnicas são discutidos nesta seção.

#### 2.8.2.1 *Métodos baseados em conjuntos*

Na abordagem baseada em conjuntos para reconhecimento facial, os quadros extraídos de um vídeo são tratados como uma coleção de amostras de imagem individuais, sem levar em conta sua sequência temporal. Esses métodos foram projetados especificamente para enfrentar os desafios associados a imagens de baixa resolução, comumente capturadas por câmeras de vigilância (ZHENG; CHELLAPPA, 2021).

As abordagens baseadas em conjuntos podem ser classificadas em dois tipos: fusão antes da correspondência e fusão após a correspondência. Na fusão antes da correspondência, as características extraídas de cada imagem facial são combinadas antes do processo de reconhecimento. Já na fusão após a correspondência, o reconhecimento é realizado de forma independente em cada quadro, e os resultados individuais são integrados posteriormente para gerar uma decisão final. Essa estratégia permite uma análise mais robusta ao considerar a redundância e a complementaridade entre os quadros, garantindo maior precisão mesmo em cenários desafiadores (BAI *et al.*, 2021).

#### 2.8.2.2 *Métodos baseados em sequências*

No reconhecimento facial baseado em sequência, as imagens extraídas de um vídeo são processadas considerando sua ordem temporal, permitindo a análise de dinâmicas faciais ao longo do tempo. Essa abordagem explora informações sequenciais para melhorar a precisão do reconhecimento, especialmente em condições desafiadoras, como variações de iluminação, poses e expressões faciais (KOMOROWSKI; ROKITA, 2013; HU *et al.*, 2018).

Gong *et al.* (2019), argumenta que um vídeo de rosto apresenta uma sequência de quadros que podem ser utilizados para realizar o reconhecimento facial. Essa abordagem possibilita a exploração de informações contextuais e temporais associadas a uma identidade, cujas imagens faciais são capturadas de forma sequencial ao longo do vídeo.

Para este trabalho, foi utilizada a abordagem baseada em imagens para implementar um sistema de reconhecimento facial.

### 3 METODOLOGIA

Este capítulo apresenta a metodologia utilizada para o desenvolvimento do estudo, detalhando os procedimentos adotados para a implementação e avaliação dos algoritmos de reconhecimento facial. Inicialmente, são descritas as tecnologias empregadas, incluindo bibliotecas e ferramentas essenciais para o processamento de imagens e modelagem dos algoritmos. Em seguida, é delineado o pipeline metodológico, destacando as principais etapas envolvidas, desde a aquisição e pré-processamento das imagens até o treinamento e avaliação dos modelos. Além disso, são abordadas as estratégias adotadas para a configuração dos parâmetros dos algoritmos *Eigenfaces*, *Fisherfaces* e LBPH, assim como o procedimento de treinamento e a usabilidade dos modelos no processo de reconhecimento facial. Por fim, são apresentadas as métricas e os critérios empregados na avaliação do desempenho dos algoritmos, garantindo uma análise comparativa consistente entre as abordagens estudadas.

#### 3.1 Tecnologias utilizadas

Os sistemas de reconhecimento facial baseados nos algoritmos *Eigenfaces*, *Fisherfaces* e LBPH utilizam bibliotecas comuns essenciais para o processamento de imagens e a construção de modelos de reconhecimento facial. Nesta seção, serão discutidas brevemente algumas dessas bibliotecas:

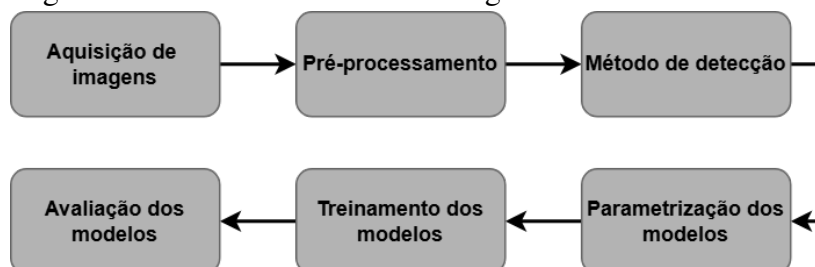
- a) *cv2* (*OpenCV - Open Source Computer Vision*) é uma ferramenta de código aberto amplamente utilizada para visão computacional e aprendizado de máquina em linguagens como C++, *python* e outras. Reconhecida como uma das bibliotecas mais populares no campo, ela oferece uma ampla gama de funções e algoritmos para manipulação de imagens, detecção e reconhecimento de objetos, processamento de vídeos, entre outros. Nos sistemas de reconhecimento facial, a *cv2* desempenha papéis fundamentais, como carregar imagens faciais, realizar operações de pré-processamento, detectar e recortar regiões faciais, converter imagens para escala de cinza e salvar imagens durante o cadastramento (CERVERA, 2020);
- b) *numPy*: é a principal ferramenta para programação de matrizes em *python*, oferecendo recursos poderosos e eficientes para computação científica. Nos sistemas de reconhecimento facial, o *NumPy* é utilizado para manipular e transformar matrizes de imagens, desempenhando funções como a normalização da intensidade

- dos pixels, cálculos estatísticos e outras operações numéricas essenciais durante o pré-processamento e o treinamento dos modelos (HARRIS *et al.*, 2020);
- c) *PIL*: (*Python Imaging Library*) é uma biblioteca amplamente utilizada na linguagem *python* para processamento de imagens. Ela oferece uma variedade de funcionalidades para manipulação, edição e processamento de imagens, permitindo que desenvolvedores trabalhem com imagens de forma eficiente e flexível (KIM *et al.*, 2020);
- d) *python*: A linguagem de programação *python* foi escolhida para criar o código-fonte do projeto devido à sua versatilidade, ampla comunidade de desenvolvedores e à existência de diversos projetos relacionados ao reconhecimento facial implementados nessa linguagem;
- e) *dataset Labeled Faces in the Wild* (LFW): contendo imagens coloridas de indivíduos distintos. As fotografias apresentam variações em expressões faciais (LI, 2018);
- f) *dataset Privado*: Contendo imagens do autor criado a partir de vídeo durante a etapa de captura de imagens, as mesmas apresentam variação de expressões faciais;
- g) *hardware*: Foi utilizado um desktop com as seguintes configurações: AMD *Ryzen 5* 3.60GHz 6-Core, RAM 16 GB, SSD 500 GB, Placa gráfica GTX 1050 TI 4GB. Tais configurações foram essenciais para garantir o processamento das imagens de forma fluida e eficiente.

### 3.2 Pipeline

O procedimento metodológico adotado para o desenvolvimento deste trabalho é dividido em seis etapas. Ele inicia com a aquisição de imagens, seguido pelas etapas de pré-processamento, detecção facial, reconhecimento facial, treinamento dos modelos e, por fim, a avaliação dos modelos, conforme ilustra a Figura 30.

Figura 30 – Procedimento metodológico adotado



Fonte: Elaborado pelo autor (2024)

### 3.3 Aquisição das imagens

Nesta etapa, são coletadas imagens faciais de um conjunto de pessoas que serão cadastradas no sistema a partir de dois *datasets*: o LFW e outro criado com imagens capturadas do próprio autor. As imagens apresentam uma variedade de características faciais, como variação de ângulos, aparência e expressões faciais. A qualidade das imagens também foi levada em consideração; todas as imagens são nítidas e bem iluminadas. Os dois *datasets* combinados contêm um total de 400 imagens. A qualidade e a diversidade de um *dataset* fornecem uma base sólida para a fase de treinamento dos algoritmos, assegurando a eficiência e o desempenho de um sistema de reconhecimento facial. A Figura 31 apresenta algumas imagens do *dataset*.

Figura 31 – Exemplos de imagens dos *datasets*

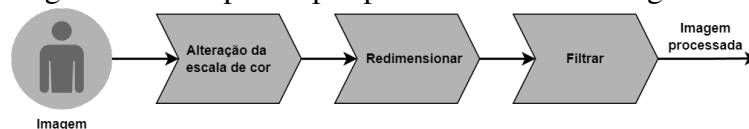


Fonte: Elaborado pelo autor (2024)

### 3.4 Pré-processamento

O pré-processamento é fundamental para aprimorar a qualidade das imagens e minimizar variações indesejadas que possam comprometer o desempenho do sistema de reconhecimento facial (GONZALEZ; WOODS, 2010). A Figura 32 apresenta o diagrama das etapas de pré-processamento de imagem do projeto.

Figura 32 – Etapas de pré-processamento de imagem



Fonte: Elaborado pelo autor (2024)

As imagens coletadas foram submetidas a um processo de pré-processamento para assegurar a qualidade e a uniformidade dos dados. A primeira etapa consistiu na conversão das imagens para a escala de cinza, eliminando informações de cor e reduzindo a complexidade computacional do processamento, o que é essencial para algoritmos de reconhecimento facial baseados em características estruturais.

Na segunda etapa, as imagens foram redimensionadas para um tamanho padronizado de  $100 \times 100$ <sup>1</sup> pixels, garantindo a consistência das dimensões entre os dados de entrada. Esse redimensionamento facilita o treinamento dos algoritmos e melhora a comparabilidade entre diferentes amostras, além de otimizar o uso de memória e o tempo de processamento.

A terceira etapa envolveu a aplicação de filtros para realce e remoção de ruídos, aprimorando a nitidez e os contornos faciais. Foi empregado um filtro de ajuste de luminosidade, isto porque alguns algoritmos apenas reconhecem imagens se a luminosidade delas for superior a 110. Essas etapas garantem que as imagens faciais utilizadas estejam em um formato consistente e de qualidade suficiente para garantir um bom desempenho no processo de detecção e posterior reconhecimento facial. A Figura 33 exemplifica as imagens processadas.

Figura 33 – Exemplo de Imagens pós pré-processamento



Fonte: Elaborado pelo autor (2024)

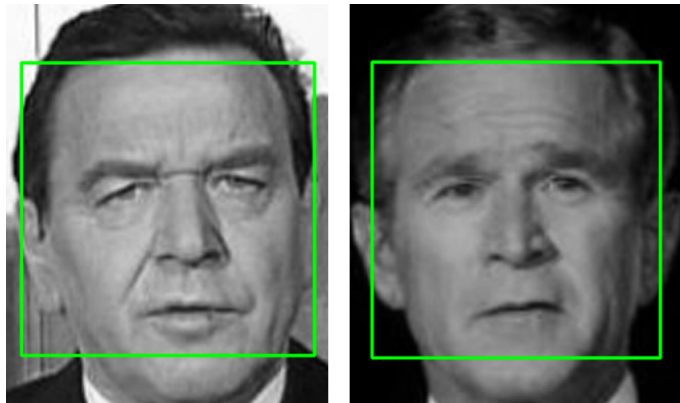
### 3.5 Método de detecção facial

Nesta etapa, utilizou-se a técnica de *Haar Cascade*, baseada no método Viola-Jones já mencionado no capítulo 2 para a detecção facial. Esse algoritmo foi selecionado por ser amplamente reconhecido por sua eficácia na identificação de regiões faciais (PEREIRA, 2017).

<sup>1</sup> O Haar Cascade original (treinado com o algoritmo Viola-Jones) foi otimizado para detectar faces em janelas relativamente pequenas (ex.:  $24 \times 24$  pixels durante o treinamento). Redimensionar para  $100 \times 100$  mantém detalhes suficientes para a detecção, sem sobrecarregar o sistema (OPENCV.ORG, 2024).

O processo de detecção passou por algumas etapas, iniciando com o pré-processamento das imagens, onde a imagem é preparada para a detecção facial, ajustando a iluminação, o contraste e removendo ruídos. Na próxima etapa, o algoritmo procura por características específicas do rosto, como olhos, nariz e boca. Uma vez detectadas as características, o algoritmo delimita a área do rosto na imagem, criando um retângulo ao redor dela. O algoritmo verifica se a delimitação do rosto está correta e a ajusta se necessário. A Figura 34 apresenta alguns exemplos.

Figura 34 – Demarcação da face detectada



Fonte: Elaborado pelo autor (2024)

### 3.6 Parametrização dos modelos

Os algoritmos utilizados neste trabalho possuem parâmetros que influenciam diretamente suas estratégias de extração de características e classificação. Assim, os principais parâmetros serão descritos nesta seção.

O *Eigenfaces* utiliza dois parâmetros para a análise do PCA, o primeiro é o número de componentes (*num\_components*) a serem mantidos no momento da redução da dimensionalidade da base. O segundo parâmetro é o limiar (*threshold*) aplicado no momento da classificação de uma face. O *Fisherfaces* assim como o *Eigenfaces*, também possui dois parâmetros: o número de componentes e o limiar. O número de componentes é utilizado para ajustar o LDA na redução da dimensionalidade. O limiar é aplicado na classificação de uma face.

Diferentemente dos outros dois algoritmos, o LBPH possui cinco parâmetros: raio (*radius*), vizinhos (*neighbors*), grade x (*grid\_x*), grade y (*grid\_y*) e limiar (*threshold*). O raio é usado para construir o LBP circular, que representa a distância ao redor do pixel central. Vizinhos é o número de pontos de amostra para construir o LBP circular. Grade x é o número de células na direção horizontal. Grade y é o número de células na direção vertical. O limiar é aplicado na

classificação.

Os valores dos parâmetros foram obtidos empiricamente por meio de testes exaustivos. As Tabelas 1, 2 e 3 apresentam as melhores combinações encontradas para os algoritmos *Eigenfaces*, *Fisherfaces* e LBPH respectivamente.

Tabela 1 – Parâmetros para o algoritmo Eigenfaces

Algoritmo	Num_Components	Threshold
Eigenfaces	200	13000
	60	8000
	45	9000

Fonte: Elaborado pelo autor (2024)

Tabela 2 – Parâmetros para o algoritmo Fisherfaces

Algoritmo	Num_Components	Threshold
Fisherfaces	2	1000
	3	2000
	25	2000

Fonte: Elaborado pelo autor (2024)

Tabela 3 – Parâmetros para o algoritmo LBPH

Algoritmo	Radius	Neighbors	grid_x	grid_y	Threshold
LBPH	3	3	7	7	45
	3	4	8	8	80
	4	4	6	6	50

Fonte: Elaborado pelo autor (2024)

Os três algoritmos apresentados utilizam o classificador KNN<sup>2</sup> para calcular a distância euclidiana e realizar a classificação durante o processo de reconhecimento facial. O cálculo da distância euclidiana é dado pela Equação (3.1).

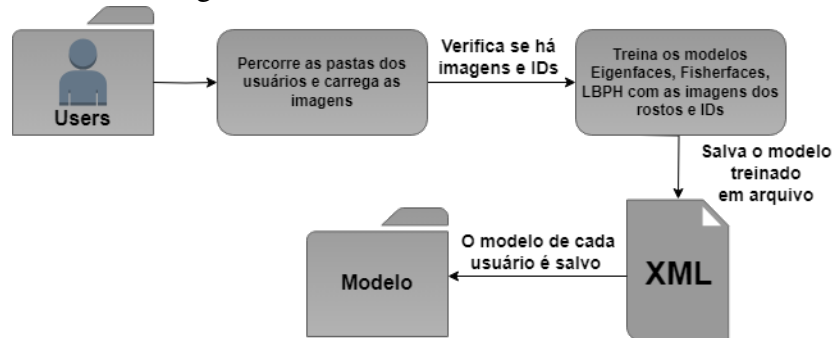
$$D(x,y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.1)$$

### 3.7 Treinamento dos modelos

A Figura 35 apresenta um diagrama que ilustra o fluxo de trabalho no processo de treinamento dos algoritmos *Eigenfaces*, *Fisherfaces* e LBPH, destacando visualmente as etapas e a sequência das operações realizadas.

<sup>2</sup> O K-Nearest Neighbor (KNN) é bastante usado em problemas de classificação, inclusive em reconhecimento facial. É um algoritmo abrangente que toma medidas de distâncias ou similaridades entre pares de padrões para realizar a classificação (JOSE *et al.*, 2012).

Figura 35 – Diagrama das etapas do processo de treinamento dos algoritmos



Fonte: Elaborado pelo autor (2024)

No diagrama acima, são apresentadas as etapas envolvidas no treinamento dos algoritmos de reconhecimento facial para um usuário específico. O processo inicia-se com a iteração pelo diretório de usuários, onde o sistema percorre as pastas individuais de cada usuário e carrega as imagens associadas a eles.

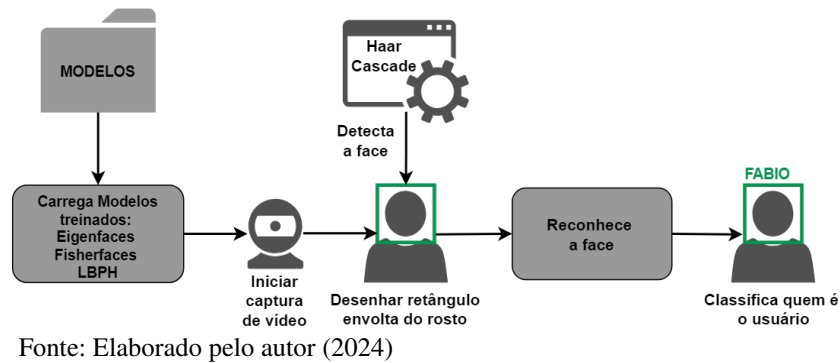
Cada usuário possui um conjunto de imagens faciais que compõem o *dataset* de treinamento, as imagens passam por um pré-processamento que inclui conversão de cor para escala de cinza, recorte e redimensionamento. Além disso, apresentam variações de iluminação, ângulos e expressões para garantir um treinamento robusto do classificador. O sistema então verifica se há imagens válidas e IDs corretamente atribuídos ao usuário. Caso positivo, o processo de treinamento é iniciado.

Após o treinamento de um algoritmo específico, os parâmetros extraídos são armazenados em um arquivo XML correspondente ao modelo treinado. Esse arquivo XML contém informações essenciais para a identificação facial futura, incluindo os vetores de características extraídos, os pesos ajustados pelo algoritmo e a relação entre os identificadores e as respectivas imagens de treino. Cada modelo treinado gera seu próprio arquivo XML, que é salvo em um diretório específico. Esse processo garante que, ao longo do tempo, o sistema possa reconhecer os usuários cadastrados de forma eficiente, utilizando os modelos previamente treinados e armazenados.

### 3.8 Usabilidade dos modelos

O diagrama da Figura 36 apresenta a sequência de operações no processo de detecção e reconhecimento facial dos algoritmos em estudo, proporcionando uma visão geral do fluxo de trabalho envolvido.

Figura 36 – Diagrama etapas do processo de detecção e reconhecimento facial



O diagrama acima ilustra as principais etapas do processo de detecção e reconhecimento facial implementado no projeto. Para viabilizar esse procedimento, foi desenvolvido um código em *python* que utiliza modelos treinados previamente, conforme descrito na seção anterior, para identificar a identidade do indivíduo presente na imagem ou vídeo.

O fluxo inicia-se com o carregamento dos modelos treinados dos algoritmos *Eigenfaces*, *Fisherfaces* e *LBPH*, armazenados em arquivos XML. Em seguida, a captura de vídeo é iniciada, permitindo a análise em tempo real das imagens adquiridas. A detecção de rostos é realizada utilizando o classificador *Haar Cascade*, que identifica padrões faciais na cena. Quando um rosto é detectado, um retângulo é desenhado ao seu redor para destacar a região de interesse.

Na etapa seguinte, cada rosto detectado é submetido à fase de reconhecimento facial, na qual o modelo preditivo analisa as características extraídas e calcula os níveis de confiança da correspondência. A nova face capturada é então comparada com as imagens previamente armazenadas no treinamento. Caso seja identificada uma correspondência suficientemente precisa, o sistema classifica o usuário e exibe sua identificação.

### 3.9 Avaliação dos modelos

Neste passo, é realizada a avaliação dos modelos apresentados na seção anterior. Neste projeto, optou-se por trabalhar com métricas confiáveis, como o *classification report*, que inclui acurácia, precisão, *recall* e *F1-score*, além da matriz confusão. Essas métricas refletem diretamente a capacidade do algoritmo de identificar corretamente indivíduos previamente registrados no treinamento (WEISS; KULIKOWSKI, 1991). A seguir, serão apresentadas as definições das métricas deste estudo, bem como as suas fórmulas matemáticas.

- a) acurácia (Accuracy): Mede a proporção de previsões corretas em relação ao total de previsões. Indica a performance geral do modelo (WEISS; KULIKOWSKI, 1991).

É calculada por:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN}$$

- b) precisão (Precision): Reflete a proporção de verdadeiros positivos em relação a todas as previsões positivas. Mostra a confiabilidade do modelo ao classificar uma instância como positiva (SCHÜTZE *et al.*, 2008). Sua fórmula é:

$$\text{Precisão} = \frac{TP}{TP + FP}$$

- c) recall (Revocação ou Sensibilidade): Indica a proporção de verdadeiros positivos em relação a todos os casos que são realmente positivos. Mede a capacidade do modelo de identificar todas as instâncias relevantes (GOODFELLOW *et al.*, 2016).

É calculado por:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- d) F1-Score: Combina precisão e recall em uma única métrica, sendo a métrica harmônica entre elas (JURAFSKY; MARTIN, 2020). Sua fórmula é:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

Onde:

- a) TP (True Positive): Casos positivos corretamente classificados;
- b) TN (True Negative): Casos negativos corretamente classificados;
- c) FP (False Positive): Casos negativos incorretamente classificados como positivos;
- d) FN (False Negative): Casos positivos incorretamente classificados como negativos.

Foi montado um *dataset* contendo imagens capturadas do autor, juntamente com imagens do *dataset* LFW, totalizando 400 amostras. Essas amostras foram divididas para compor os conjuntos de treino e teste. A divisão das imagens teve embasamento no método de validação cruzada (holdout)<sup>3</sup> com a separação de parte das imagens faciais da base de imagens original para a base de teste. Do conjunto total das imagens, foram retirados 12% das imagens aleatoriamente para compor o conjunto de teste, evitando que houvesse interseção das imagens entre os conjuntos de treino e teste (REFAEILZADEH *et al.*, 2009). Além disso, o método de validação cruzada

<sup>3</sup> Holdout é um método de validação cruzada que consiste em dividir o conjunto total de dados em dois subconjuntos mutuamente exclusivos, um para treinamento (estimação dos parâmetros) e outro para teste (validação). O conjunto de dados pode ser separado em quantidades iguais ou não (KOHAVI *et al.*, 1995).

é descrito na literatura como uma das formas mais eficazes de reduzir o *overfitting*<sup>4</sup>, que é um problema significativo em tarefas de detecção e reconhecimento facial (KRONBERGER *et al.*, 2011).

A base de treinamento contém 70 amostras de cada indivíduo, totalizando 350 amostras utilizadas para treinar os algoritmos. As imagens abrangem variações de posição e expressão facial, visando melhorar a identificação dos indivíduos. Para compor a base de teste, foram separadas 10 amostras de cada indivíduo, totalizando 50 amostras. As imagens da base de teste são distintas das da base de treinamento, variando em poses e expressões faciais para garantir que as métricas reflitam o desempenho real do modelo.

---

<sup>4</sup> Overfitting (sobreajuste) é um fenômeno que ocorre quando um modelo se ajusta tão bem aos dados específicos que foram usados para treiná-lo que ele perde a capacidade de generalizar para novos dados não vistos. Ele “memoriza” os exemplos de treinamento em vez de aprender os padrões subjacentes e gerais (YING, 2019).

## 4 RESULTADOS

Este capítulo apresenta e analisa os resultados obtidos a partir da implementação e avaliação dos algoritmos *Eigenfaces*, *Fisherfaces* e LBPH no reconhecimento facial. A avaliação baseia-se em métricas amplamente utilizadas na literatura, incluindo precisão, recall, f1-score, acurácia e matriz confusão, permitindo uma comparação justa entre os métodos.

Inicia-se esta seção apresentando os resultados individuais de cada algoritmo, destacando seus desempenhos e possíveis limitações. Em seguida, realiza-se uma análise comparativa para identificar o algoritmo mais eficiente. Além disso, são discutidos os principais fatores que influenciaram a acurácia, como variações de iluminação e expressões faciais.

### 4.1 Resultados *Eigenfaces*

Os resultados obtidos pelo algoritmo *Eigenfaces*, conforme apresenta a Tabela 4, mostram uma acurácia geral de 84%, demonstrando desempenho razoável. A classe Fabio destacou-se com precisão de 90,9%, *recall* de 100% e *F1-score* de 95,2%, apesar de um falso positivo que reduziu a precisão e *F1-score*. Gerhard Schroeder também teve um bom desempenho com 90% em todas as métricas. Essa baixa no percentual foi devido a incidências de falsos negativos e positivos. Colin Powell apresentou precisão de 88,9%, mas um *recall* inferior de 80%, indicando perdas de algumas instâncias. As classes George W. Bush e Tony Blair registraram os piores resultados, com *F1-scores* de 76,2% e 73,7%, respectivamente, sugerindo dificuldades na identificação correta, possivelmente devido a variações de iluminação e ângulo das imagens. O modelo foi eficaz em algumas classes, mas apresentou sensibilidade a variações visuais.

Tabela 4 – Métricas de desempenho *Eigenfaces*

EIGENFACES				
Parâmetros: 45, 9000				
Classes	Precisão	Recall	F1-Score	Acurácia Geral
Colin Powell	0.889	0.80	0.842	
Fabio	0.909	1.00	0.952	
George W. Bush	0.727	0.80	0.762	<b>0.84</b>
Gerhard Schroeder	0.90	0.90	0.90	
Tony Blair	0.778	0.70	0.737	

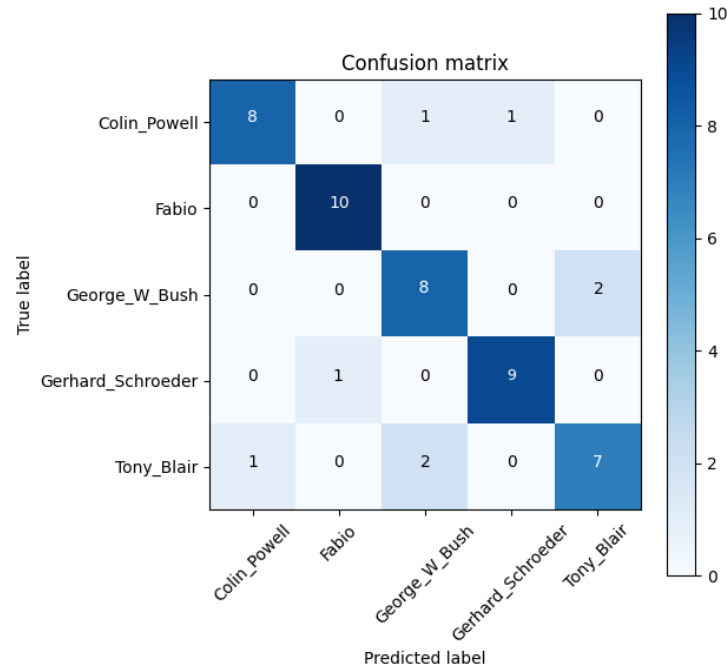
Fonte: Elaborado pelo autor (2024). Acurácia Geral: Trata-se de uma métrica global que expressa o percentual de acertos do algoritmo ao considerar todas as classes avaliadas no conjunto de teste.

A Figura 37 apresenta a matriz confusão do algoritmo *Eigenfaces*, que mostra uma avaliação detalhada da classificação das amostras. O modelo classificou corretamente todas as instâncias reais de Fabio sem nenhum falso negativo. Porém, uma instância de Gerhard Schroeder foi classificada como Fabio, gerando um falso positivo e afetando levemente a precisão e o *F1-score*, apresentando um desempenho quase perfeito. Já a classe Gerhard Schroeder teve nove instâncias reais de Gerhard Schroeder corretamente classificadas. Porém, uma instância real de Gerhard Schroeder foi classificada incorretamente como Fabio (falso negativo) e outra classe foi incorretamente classificada como Gerhard Schroeder, representando um falso positivo. O modelo concede bom desempenho para Gerhard Schroeder.

A Classe Colin Powell obteve oito instâncias classificadas corretamente como Colin Powell, duas instâncias reais de Colin Powell foram incorretamente classificadas como George W. Bush e Gerhard Schroeder (falsos negativos) e uma instância de outra classe foi incorretamente classificada como Colin Powell (falsos positivos). O modelo tem uma boa performance para Colin Powell, apesar de confundir algumas instâncias. A classe George W. Bush obteve oito instâncias corretamente classificadas como George W. Bush. Contudo, duas instâncias reais de George W. Bush foram incorretamente classificadas como Tony Blair, ou seja, falsos negativos. Além disso, três instâncias de outras classes foram incorretamente classificadas como George W. Bush (falsos positivos). No entanto, o desempenho para George W. Bush é considerado bom.

Por fim, a classe Tony Blair teve sete instâncias classificadas corretamente como Tony Blair e três instâncias reais de Tony Blair foram incorretamente classificadas como Colin Powell e George W. Bush (falsos negativos). Além disso, duas instâncias de outra classe foram incorretamente classificadas como Tony Blair (falsos positivos). Essa é a classe com o maior número de falsos negativos, o que justifica o seu menor *recall* (0.70) e *F1-score* (0.737). O modelo teve dificuldades em distinguir essas duas classes, o que pode indicar sensibilidade a variações de iluminação, como dito anteriormente, ou ainda características faciais muito parecidas entre os dois indivíduos no conjunto de dados.

Figura 37 – Matriz Confusão Eigenfaces



Fonte: Elaborado pelo autor (2024)

## 4.2 Resultados *Fisherfaces*

Os resultados obtidos pelo *Fisherfaces* indicam um bom desempenho global na classificação das imagens. O algoritmo alcançou uma acurácia de 94%, apresentando uma taxa de acerto elevada para a maioria dos indivíduos, mas ainda houve erros em algumas categorias, conforme observa-se na Tabela 5.

Tabela 5 – Métricas de desempenho *Fisherfaces*

FISHERFACES				
Parâmetros: 25, 2000				
Classes	Precisão	Recall	F1-Score	Acurácia Geral
Colin Powell	1.00	0.90	0.95	
Fabio	1.00	1.00	1.00	
George W. Bush	0.83	1.00	0.91	<b>0.94</b>
Gerhard Schroeder	0.91	1.00	0.95	
Tony Blair	1.00	0.80	0.89	

Fonte: Elaborado pelo autor (2024)

De acordo com o relatório, a classe Fabio obteve precisão e *recall* de 100%, indicando que todas as previsões para esta classe estavam corretas e que nenhuma instância foi omitida. Isso pode sugerir que as características dessa classe são bem separadas das demais no espaço de características. A classe Colin Powell teve alta precisão de 100%, mas um *recall* ligeiramente

menor de 90%, indicando que uma instância foi classificada incorretamente. A classe Gerhard Schroeder teve um *recall* perfeito de 100% e alta precisão de 91%, mostrando que todas as instâncias dessa classe foram identificadas, com apenas um falso positivo.

Apesar do bom desempenho da maioria das classes, duas apresentaram desempenho inferior. A classe Tony Blair, com um *recall* de 80%, indica que duas das dez instâncias dessa classe foram classificadas incorretamente, resultando em falsos negativos. Já a precisão de 83% para a classe George W. Bush indica que outras classes estão sendo confundidas com George W. Bush, gerando falsos positivos, possivelmente devido a similaridades nas características faciais.

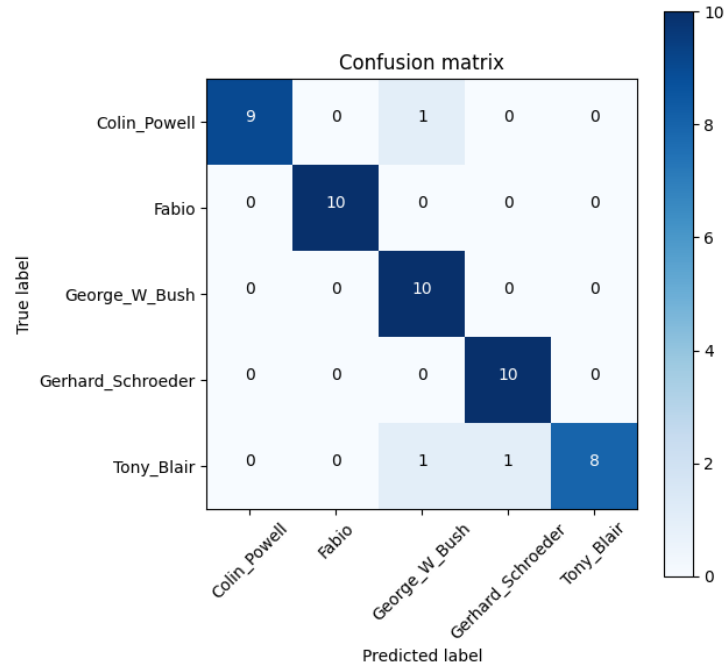
Embora o *Fisherfaces* seja mais robusto a variações na iluminação, ainda pode ser sensível a diferenças sutis na iluminação, inclinação da cabeça ou expressões faciais. Se as imagens de George W. Bush e Tony Blair apresentavam tais variações, isso pode ter influenciado os erros.

A matriz confusão do modelo Fisherfaces apresentada na Figura 38 revelou um desempenho geral muito bom, com acurácia de 94%. Destaca-se a classe Fabio, com 100% de acertos, sem a ocorrência de FP ou FN, indicando excelente discriminação de suas características faciais pelo modelo. Entretanto, a classe Tony Blair apresentou dois falsos negativos, com imagens classificadas como George W. Bush e Gerhard Schroeder, o que indica certa sobreposição entre essas representações faciais. Além disso, George W. Bush sem falsos negativos, foi erroneamente atribuído a imagens de Colin Powell e Tony Blair, tornando-se a classe com maior incidência de falsos positivos.

A classe Colin Powell teve um erro FN ao ser confundido com George W. Bush, enquanto Gerhard Schroeder sofreu apenas um FP oriundo de Tony Blair. Esses erros podem ter ocorrido possivelmente por similaridades faciais ou devido a expressões ou iluminação semelhantes.

De acordo com a análise da matriz confusão, o Fisherfaces tem um desempenho excelente para a maioria das classes, mas enfrenta desafios com Tony Blair e Colin Powell, principalmente devido às similaridades de expressões ou características faciais. A presença de FPs e FNs específicos sugere a necessidade de ajustes direcionados para melhorar a discriminação entre essas classes.

Figura 38 – Matriz Confusão Fisherfaces



Fonte: Elaborado pelo autor (2024)

### 4.3 Resultados LBPH

O modelo LBPH demonstrou desempenho geral bastante satisfatório, com acurácia total de 96% no conjunto de testes, conforme apresenta a Tabela 6. Todas as classes apresentaram valores elevados de precisão, *recall* e *F1-score*, evidenciando a eficácia do método para o problema de reconhecimento facial.

Tabela 6 – Métricas de desempenho *LBPH*

<b>LBPH</b>				
Parâmetros: 4, 4, 6, 6, 50				
<b>Classes</b>	<b>Precisão</b>	<b>Recall</b>	<b>F1-Score</b>	<b>Acurácia Geral</b>
Colin Powell	1.00	1.00	1.00	
Fabio	1.00	1.00	1.00	
George W. Bush	1.00	0.80	0.89	<b>0.96</b>
Gerhard Schroeder	0.91	1.00	0.95	
Tony Blair	0.91	1.00	0.95	

Fonte: Elaborado pelo autor (2024)

A classe Colin Powell e Fabio obtiveram desempenho perfeito, com precisão, *recall* e *F1-score* iguais a 100%, indicando que todas as amostras foram corretamente classificadas, sem ocorrência de falsos positivos ou falsos negativos. Esse resultado sugere que as representações faciais desses indivíduos foram bem capturadas pelas características locais extraídas pelo LBPH.

Por outro lado, a classe George W. Bush apresentou uma queda no *recall* para 80%, o que implica que duas imagens foram classificadas erroneamente como pertencentes a outras classes (falsos negativos). Obteve uma precisão de 100%, indicando que não houve a ocorrência de falsos positivos, o desempenho inferior no *recall* compromete sua *F1-score* de 89%. Esse resultado pode estar relacionado a variações de iluminação, expressões faciais ou ruído nas imagens associadas a essa identidade. As classes Gerhard Schroeder e Tony Blair apresentaram precisão de 91% e *recall* de 100%, sugerindo que, embora todas as suas imagens tenham sido reconhecidas corretamente, houve ao menos um falso positivo envolvendo essas classes. Esses erros podem indicar que algumas características locais dessas classes são semelhantes às de outras classes, levando a confusões pontuais.

Portanto, embora o LBPH tenha mostrado ser um método confiável em geral, uma análise mais aprofundada indica que ele ainda pode apresentar dificuldades em lidar com variações dentro da mesma classe, o que pode levar a erros. Isso é especialmente verdade para indivíduos com características faciais mais comuns ou quando as imagens estão sujeitas a ruído ou distorções.

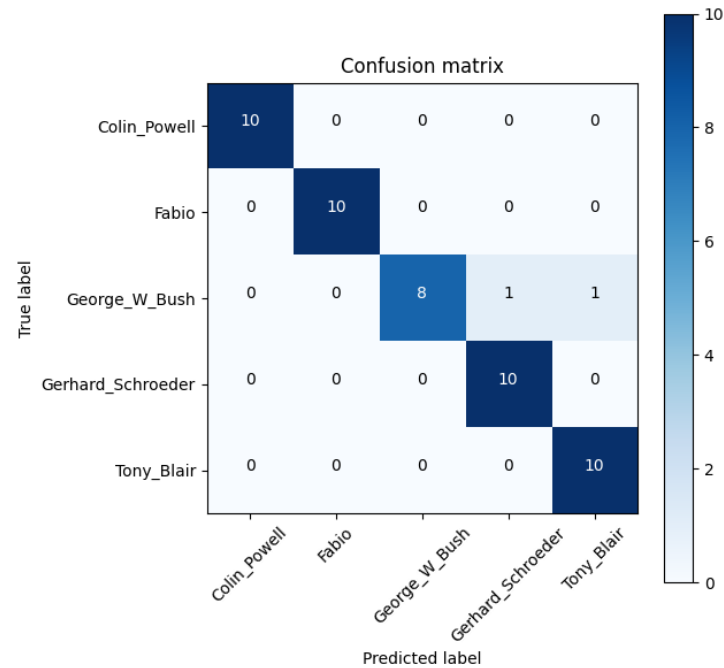
A matriz confusão da Figura 39, referente ao modelo LBPH, evidencia um desempenho robusto do algoritmo, que apresentou acertos perfeitos (10/10) para quatro das cinco classes: Colin Powell, Fabio, Gerhard Schroeder e Tony Blair. Esses resultados indicam zero falsos negativos nessas classes, refletindo a capacidade do modelo em distinguir corretamente as características faciais desses indivíduos. No entanto, a classe George W. Bush apresentou dois erros de classificação, revelando falsos negativos em duas instâncias: uma foi incorretamente classificada como Gerhard Schroeder e outra como Tony Blair. Assim, a classe teve um *recall* de 80% (8 acertos em 10), o que representa uma dificuldade do algoritmo em reconhecer corretamente todas as amostras de George W. Bush.

Por outro lado, tanto Gerhard Schroeder quanto Tony Blair, apesar de apresentarem 100% de acertos em suas próprias classes (*recall* = 1.00), receberam uma amostra incorretamente atribuída a elas, ou seja, tiveram um falso positivo cada, afetando a precisão e o *F1-score*. Esses falsos positivos e negativos indicam alguma sobreposição nas representações faciais entre George W. Bush e os demais indivíduos, possivelmente causadas por expressões faciais muito semelhantes, condições de iluminação ou até mesmo ângulo de captura das imagens.

Em síntese, o desempenho do algoritmo LBPH é altamente satisfatório, com mínima taxa de erro global, mas identificam-se dificuldades de classificação específicas na classe George

W. Bush, que merece atenção, sobretudo em cenários críticos onde a taxa de falsos negativos pode comprometer a confiabilidade do sistema de segurança.

Figura 39 – Matriz Confusão LBPH



Fonte: Elaborado pelo autor (2024)

#### 4.4 Comparação entre os algoritmos *Eigenfaces*, *Fisherfaces*, LBPH

Os três algoritmos testados apresentaram desempenhos distintos na tarefa de reconhecimento facial, evidenciados pelas métricas de acurácia, precisão, *recall* e *F1-score*. Em função disto, nesta seção será feita uma comparação dos resultados e uma análise crítica de suas vantagens e limitações.

O algoritmo LBPH se destacou como o método mais eficaz da comparação, alcançando uma acurácia geral de 96% e maior consistência entre as classes. Esse resultado reforça sua capacidade de lidar com variações faciais complexas e iluminação. As classes Fabio e Colin Powell obtiveram *F1-score* perfeito (1.00). A principal limitação foi o *recall* de 0.80 para George W. Bush, ocasionado por falsos negativos.

O Fisherfaces alcançou 94% de acurácia, apresentando uma boa relação entre precisão e *recall* em praticamente todas as classes. Destacou-se também na classe Fabio (F1=1.00), mas apresentou erros pontuais em George W. Bush e Tony Blair, com dois falsos positivos e um *recall* de 0.80, respectivamente. Isso evidencia que o algoritmo ainda sofre com a distinção

entre sujeitos com traços similares. Já o *Eigenfaces* apresentou o desempenho mais baixo entre os três métodos, com 84% de acurácia geral e maior instabilidade entre as classes. A principal limitação foi o alto número de falsos positivos, especialmente na classe George W. Bush, que foi erroneamente atribuída a três imagens de outros indivíduos. A classe Tony Blair também teve um desempenho insatisfatório com  $F1=0.737$ , destacando a dificuldade do algoritmo em separar corretamente as características faciais mais sutis.

O tempo de treinamento também foi levado em consideração como métrica para medir a eficiência computacional de cada método. Dentre os três, o *Fisherfaces* apresentou o menor tempo de treinamento de 3.66s, destacando-o como o método mais eficiente nessa fase. Esse resultado pode estar ligado à sua redução dimensional baseada em LDA. O *Eigenfaces*, teve tempo ligeiramente maior que o *Fisherfaces*, pois sua abordagem baseada em PCA depende da variabilidade da imagem, podendo demandar mais tempo em bases complexas. Já o LBPH, com 4.10s, apresentou o maior tempo de treinamento entre os três métodos. Esse resultado era esperado, pois o LBPH calcula histogramas locais para cada região da imagem, que exigem mais desempenho computacional. Por tanto, se a prioridade for um treinamento mais rápido, o *Fisherfaces* seria a melhor escolha. Entretanto, a escolha ideal deve considerar não apenas o tempo de treinamento, mas também a precisão e robustez do reconhecimento facial. A Tabela 7 apresenta o tempo de treinamento dos algoritmos.

Tabela 7 – Tempo de treinamento dos algoritmos

<b>Algoritmo</b>	<b>Tempo de Treinamento (s)</b>
Eigenfaces	3.84
Fisherfaces	3.66
LBPH	4.10

Fonte: Elaborado pelo autor (2024)

Em síntese, o *Eigenfaces* teve o pior desempenho devido à sua sensibilidade à iluminação e variações globais na imagem. O *Fisherfaces* apresentou uma melhoria, principalmente na diferenciação de classes que foram problemáticas no *Eigenfaces*. O LBPH foi o mais eficaz, possivelmente devido à sua abordagem baseada em padrões locais, o que o tornou mais robusto a variações faciais. Em um cenário real, o LBPH seria a melhor escolha para um sistema de reconhecimento facial mais confiável, considerando sua maior robustez e precisão. No entanto, a escolha final dependeria do contexto de aplicação.

## 5 CONCLUSÃO

Com o auxílio de trabalhos relacionados e estudos, foi possível realizar o presente trabalho, testar e analisar mais a fundo os métodos Eigenfaces, Fisherfaces e LBPH, e comprovar o funcionamento e resultados desses modelos. A eficácia e comportamento deles foram medidos e apresentados com diversos experimentos utilizando variação paramétrica e métricas de controle da qualidade, com análise dos resultados e testes.

De acordo com os experimentos realizados, o LBPH se destacou como o algoritmo mais eficiente para o reconhecimento facial na abordagem utilizada, ele apresentou os melhores resultados de precisão, recall, F1-score e acurácia, comprovando sua robustez mesmo em condições adversas, como variações de iluminação e expressões faciais. Embora seu tempo de treinamento tenha sido o mais alto entre os demais métodos, essa diferença não compromete a sua eficácia.

Em relação ao algoritmo Fisherfaces, que demonstrou ser um algoritmo bem equilibrado entre desempenho e eficiência, obtendo resultados consistentes e um tempo de treinamento menor do que os demais algoritmos, esse modelo pode ser uma opção viável para aplicações que necessitam de treinamento rápido sem grandes perdas de precisão.

Já o Eigenfaces apresentou um bom desempenho geral, porém mostrou-se mais sensível a variações de iluminação e expressões faciais, reduzindo sua capacidade de diferenciar algumas classes. Isso pode explicar sua performance inferior aos outros modelos.

Conclui-se que, embora o método LBPH tenha se saído melhor nos experimentos, a escolha do algoritmo ideal depende do contexto da aplicação. O LBPH é indicado como a melhor alternativa quando a prioridade é a precisão do reconhecimento, enquanto o Fisherfaces pode ser mais indicado onde o tempo de treinamento é um fator relevante, em um contexto com pouca variação facial. O Eigenfaces, apesar de suas limitações, ainda pode ser útil em cenários não tão exigentes.

Por fim, este estudo contribuiu para a compreensão comparativa dos métodos de reconhecimento facial, oferecendo um referencial para a escolha do método mais adequado para a aplicação de um sistema de controle de acesso para ambientes internos. Trabalhos futuros podem comparar o algoritmo LBPH com abordagens mais atuais, como Redes Neurais Profundas (DNN), ou realizar implementações híbridas, combinando o LBPH com outras técnicas a fim de aprimorar o processo de reconhecimento facial.

## REFERÊNCIAS

- ADJABI, I.; OUAHABI, A.; BENZAOU, A.; TALEB-AHMED, A. **Past, present, and future of face recognition: A review.** *Electronics (Basel)*, v. 9, n. 8, p. 1188, 2020. Disponível em: <<https://www.mdpi.com/2079-9292/9/8/1188>>. Acesso em: 14 jul. 2024.
- AHONEN, T.; HADID, A.; PIETIKAINEN, M. **Face description with local binary patterns: Application to face recognition.** *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 28, n. 12, p. 2037–2041, 2006. Disponível em: <<https://ieeexplore.ieee.org/document/1717463>>. Acesso em: 18 jun. 2024.
- ALMEIDA, D. d. O. **Métodos de visão computacional aplicados a extração de características de ambientes urbanos em imagens de satélite de baixa resolução.** 2018. Disponível em: <<http://repositorio.bc.ufg.br/tede/handle/tede/9036>>. Acesso em: 05 set. 2024.
- AMARAL, V.; THOMAZ, C. E. **Um estudo sobre o detalhamento espacial de descritores locais aplicados ao reconhecimento de gênero e expressões faciais.** *Anais do*, v. 3, 2013. Disponível em: <[https://fei.edu.br/~cet/VagnerAmaral\\_spgabc2013.pdf](https://fei.edu.br/~cet/VagnerAmaral_spgabc2013.pdf)>. Acesso em: 10 ago. 2024.
- BACKES, A. R.; JUNIOR, J. J. d. M. S. **Introdução à visão computacional usando Matlab.** [S.l.]: Alta Books Editora, 2019. Disponível em: <[https://altabooks.com.br/wp-content/uploads/2021/07/Capitulo\\_amostra\\_Introducao\\_a\\_visao\\_computacional\\_MATLAB.pdf](https://altabooks.com.br/wp-content/uploads/2021/07/Capitulo_amostra_Introducao_a_visao_computacional_MATLAB.pdf)>. Acesso em: 30 set. 2024.
- BAI, Z.; WANG, R.; SHAN, S.; CHEN, X. **Local feature enhancement network for set-based face recognition.** In: **2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021).** [S.l.]: IEEE, 2021. Disponível em: <<https://ieeexplore.ieee.org/document/9667020>>. Acesso em: 03 out. 2024.
- BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. **Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.** *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 19, n. 7, p. 711–720, 1997. Disponível em: <<https://ieeexplore.ieee.org/document/598228>>. Acesso em: 12 dez. 2024.
- BETTER PHOTOGRAPHY TUTORIALS. **BETTER PHOTOGRAPHY TUTORIALS.** 2011. Disponível em: <<https://betterphotographytutorials.com/2011/11/28/image-histograms-part-1/>>. Acesso em: 11 dez. 2024.
- BISSI, T. *et al.* **Reconhecimento facial com os algoritmos eigenfaces e fisherfaces.** Universidade Federal de Uberlândia, 2018. Disponível em: <<https://repositorio.ufu.br/handle/123456789/22158>>. Acesso em: 22 out. 2024.
- CERVERA, E. **GPU-accelerated vision for robots: Improving system throughput using OpenCV and CUDA.** *IEEE Robot. Autom. Mag.*, v. 27, n. 2, p. 151–158, 2020. Disponível em: <<https://ieeexplore.ieee.org/document/9047171>>. Acesso em: 25 ago. 2024.
- COSTA, V. J. d. **Reconhecimento de padrões faciais: uma síntese.** Universidade Tecnológica Federal do Paraná, 2019. Disponível em: <<http://repositorio.utfpr.edu.br/jspui/handle/1/15627>>. Acesso em: 08 jul. 2024.

DAVE, G.; CHAO, X.; SRIADIBHATLA, K. **Face recognition in mobile phones. Department of Electrical Engineering Stanford University, USA**, p. 7, 2010. Disponível em: <[https://stacks.stanford.edu/file/druid:xd064wr9736/Sriadibhatla\\_Davo\\_Chao\\_FaceRecognition.pdf](https://stacks.stanford.edu/file/druid:xd064wr9736/Sriadibhatla_Davo_Chao_FaceRecognition.pdf)>. Acesso em: 20 jul. 2024.

FENG, Y.; YU, S.; PENG, H.; LI, Y.-R.; ZHANG, J. **Detect faces efficiently: A survey and evaluations. IEEE Trans. Biom. Behav. Identity Sci.**, v. 4, n. 1, p. 1–18, 2022. Disponível em: <<https://ieeexplore.ieee.org/document/9580485>>. Acesso em: 16 dez. 2024.

FIGUEREDO, M. B. **Reconhecimento de faces aplicado ao problema de pessoas desaparecidas-Estudo de caso do Eigenface**. Faculdade de Tecnologia SENAI CIMATEC, 2011. Disponível em: <<http://repositoriosenaiba.fieb.org.br/handle/fieb/719>>. Acesso em: 14 ago. 2024.

FILHO, O. M.; NETO, H. V. **Processamento digital de imagens**. [S.l.]: Brasport, 1999. Disponível em: <<http://projetoaprendizagemgrupo4.pbworks.com/w/file/attach/96395952/Processamento%20Digital%20de%20Imagens.pdf>>. Acesso em: 23 jun. 2024.

FONSECA, J. J. S. d. **Apostila de metodologia da pesquisa científica**. [S.l.]: João José Saraiva da Fonseca, 2002. Disponível em: <<http://www.ia.ufrj.br/ppgea/conteudo/conteudo-2012-1/1SF/Sandra/apostilaMetodologia.pdf>>. Acesso em: 25 nov. 2024.

GIRON-SIERRA, J. M. **Digital signal processing with matlab examples, volume 2: Decomposition, recovery, data-based actions**. 1. ed. Singapore, Singapore: Springer, 2016. Disponível em: <<https://link.springer.com/book/10.1007/978-981-10-2537-2>>. Acesso em: 21 set. 2024.

GOEL, L.; NEOG, A.; AMAN, A.; KAUR, A. **Hybrid nature-inspired optimization techniques in face recognition**. In: **Transactions on Computational Science XXXVI**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020. p. 99–126. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-662-61364-1\\_6](https://link.springer.com/chapter/10.1007/978-3-662-61364-1_6)>. Acesso em: 28 jul. 2024.

GONG, S.; SHI, Y.; JAIN, A. K. **Recurrent Embedding Aggregation Network for video face recognition. arXiv preprint arXiv:1904.12019**, 2019. Disponível em: <<https://arxiv.org/abs/1904.12019>>. Acesso em: 01 jul. 2024.

GONZALEZ, R. C.; WOODS, R. E. **Processamento digital de imagens. 3.ed**. São Paulo: Pearson Prentice Hall, 2010. Disponível em: <<https://pdfcoffee.com/processamento-digital-de-imagens-pdf-free.html>>. Acesso em: 08 dez. 2024.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing Ed. 4**. New York, USA: Pearson, 2018. Disponível em: <<https://www.cl72.org/090imagePLib/books/Gonzales,Woods-Digital.Image.Processing.4th.Edition.pdf>>. Acesso em: 19 nov. 2024.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A.; BENGIO, Y. **Deep learning**. [S.l.]: MIT press Cambridge, 2016. v. 1. Disponível em: <<https://shre.ink/DeepLearning>>. Acesso em: 16 nov. 2024.

GURURAJ, H. L.; SOUNDARYA, B. C.; PRIYA, S.; SHREYAS, J.; FLAMMINI, F. A **comprehensive review of face recognition techniques, trends, and challenges. IEEE Access**, v. 12, p. 107903–107926, 2024. Disponível em: <<https://ieeexplore.ieee.org/document/10589377>>. Acesso em: 20 nov. 2024.

- HAMMOUCHE, R.; ATTIA, A.; AKHROUF, S.; AKHTAR, Z. **Gabor filter bank with deep autoencoder based face recognition system.** *Expert Syst. Appl.*, v. 197, n. 116743, p. 116743, 2022. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0957417422002123>>. Acesso em: 06 nov. 2024.
- HARRIS, C. R.; MILLMAN, K. J.; WALT, S. J. V. D.; GOMMERS, R.; VIRTANEN, P.; COURNAPEAU, D.; WIESER, E.; TAYLOR, J.; BERG, S.; SMITH, N. J. *et al.* **Array programming with NumPy.** *Nature*, v. 585, n. 7825, p. 357–362, 2020. Disponível em: <<https://www.nature.com/articles/s41586-020-2649-2>>. Acesso em: 30 out. 2024.
- HASAN, M. K.; AHSAN, M. S.; Abdullah-Al-Mamun; NEWAZ, S. H. S.; LEE, G. M. **Human face detection techniques: A comprehensive review and future research directions.** *Electronics (Basel)*, v. 10, n. 19, p. 2354, 2021. Disponível em: <<https://www.mdpi.com/2079-9292/10/19/2354>>. Acesso em: 13 ago. 2024.
- HO, H.-T.; NGUYEN, L. V.; LE, T. H. T.; LEE, O.-J. **Face detection using eigenfaces: A comprehensive review.** *IEEE Access*, v. 12, p. 118406–118426, 2024. Disponível em: <<https://ieeexplore.ieee.org/document/10614584>>. Acesso em: 17 jul. 2024.
- HU, W.; HUANG, Y.; ZHANG, F.; LI, R.; LI, W.; YUAN, G. **SeqFace: Make full use of sequence information for face recognition.** 2018. Disponível em: <<https://arxiv.org/abs/1803.06524>>. Acesso em: 04 set. 2024.
- ISHIDA, T. H. C. **Sistema de controle de acesso por reconhecimento facial utilizando sistemas embarcados.** Universidade Estadual Paulista (Unesp), 2023. Disponível em: <<https://repositorio.unesp.br/entities/publication/f0034fce-6f9d-461f-bacf-7824333fc84b>>. Acesso em: 30 jun. 2024.
- JANG, K.-S. **Rotation Invariant Face Detection Using HOG and Polar Coordinate Transform.** *Journal of The Korea Society of Computer and Information*, Korean Society of Computer Information, v. 26, n. 11, p. 85–92, 2021. Disponível em: <<https://koreascience.kr/article/JAKO202101355744167.page>>. Acesso em: 19 jun. 2024.
- JAYARAMAN, U.; GUPTA, P.; GUPTA, S.; ARORA, G.; TIWARI, K. **Recent development in face recognition.** *Neurocomputing*, v. 408, p. 231–245, 2020. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0925231220304951>>. Acesso em: 23 out. 2024.
- JING, Y.; LU, X.; GAO, S. **3D face recognition: A comprehensive survey in 2022.** *Computational Visual Media*, Springer, v. 9, n. 4, p. 657–685, 2023. Disponível em: <<https://ieeexplore.ieee.org/document/10897726>>. Acesso em: 07 out. 2024.
- JOSE, J. P.; POORNIMA, P.; KUMAR, K. M. **A novel method for color face recognition using KNN classifier.** In: IEEE. **2012 International Conference on Computing, Communication and Applications.** [S.l.], 2012. p. 1–3. Disponível em: <<https://ieeexplore.ieee.org/document/6179151>>. Acesso em: 03 dez. 2024.
- JR, D. P.; CARVALHO, A. S. A. de. **ANÁLISE DE ALGORITMOS DE RECONHECIMENTO FACIAL: HAAR CASCADE (VIOLA-JONES) E CNN (CONVOLUTIONAL NEURAL NETWORKS).** 2023. Disponível em: <<https://shre.ink/AnaliseDeAlgoritmosDeReconhecimentoFacialHaarcascade>>. Acesso em: 21 dez. 2024.

JUNIOR, C.; MARTINS, L. F. **Reconhecimento de faces utilizando um modelo conexcionista baseado em populações de neurônios**. São Carlos, SP; Universidade de São Paulo, 2015. Disponível em: <[https://www.teses.usp.br/teses/disponiveis/55/55134/tde-04082015-104017/publico/LuisFernandoMartinsCarlosJunior\\_disseracao\\_revisada.pdf](https://www.teses.usp.br/teses/disponiveis/55/55134/tde-04082015-104017/publico/LuisFernandoMartinsCarlosJunior_disseracao_revisada.pdf)>. Acesso em: 07 nov. 2024.

JUNIOR, J. C. S. **Processamento de Imagens: Fundamentos. Material Didático**. Rio Grande do Sul: Escola Politécnica PUCRS, 2010. Disponível em: <[https://www.inf.pucrs.br/~smusse/Simulacao/PDFs/aula\\_02\\_Fundamentos\\_PI.pdf](https://www.inf.pucrs.br/~smusse/Simulacao/PDFs/aula_02_Fundamentos_PI.pdf)>. Acesso em: 27 nov. 2024.

JURAFSKY, D.; MARTIN, J. H. **Speech and Language Processing (Draft)**. 2021. 2020. Disponível em: <<https://web.stanford.edu/~jurafsky/slp3/>>. Acesso em: 18 nov. 2024.

KARPAGAM, M. *et al.* **A novel face recognition model for fighting against human trafficking in surveillance videos and rescuing victims**. *Soft Comput.*, v. 27, n. 18, p. 13165–13180, 2023. Disponível em: <<https://link.springer.com/article/10.1007/s00500-022-06931-1>>. Acesso em: 24 out. 2024.

KIM, J.-G. *et al.* **A study on object recognition using deep learning for optimizing categorization of radioactive waste**. *Prog. Nuclear Energy*, v. 130, n. 103528, p. 103528, 2020. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0149197020302766>>. Acesso em: 29 nov. 2024.

KITANI, E. C. **Análise de discriminantes lineares para modelagem e reconstrução de imagens de faces**. Centro Universitário da FEI, São Bernardo do Campo, 2007. Disponível em: <[https://fei.edu.br/~cet/dissertacao\\_EdsonKitani\\_2007.pdf](https://fei.edu.br/~cet/dissertacao_EdsonKitani_2007.pdf)>. Acesso em: 28 set. 2024.

KOHAVI, R. *et al.* **A study of cross-validation and bootstrap for accuracy estimation and model selection**. In: MONTREAL, CANADA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Disponível em: <[https://www.researchgate.net/publication/2352264\\_A\\_Study\\_of\\_Cross-Validation\\_and\\_Bootstrap\\_for\\_Accuracy\\_Estimation\\_and\\_Model\\_Selection](https://www.researchgate.net/publication/2352264_A_Study_of_Cross-Validation_and_Bootstrap_for_Accuracy_Estimation_and_Model_Selection)>. Acesso em: 25 dez. 2024.

KOMOROWSKI, J.; ROKITA, P. **Face Recognition Based on Sequence of Images**. In: SPRINGER. **Proceedings of the 8th International Conference on Computer Recognition Systems CORES 2013**. [S.l.], 2013. p. 207–216. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-319-00969-8\\_20](https://link.springer.com/chapter/10.1007/978-3-319-00969-8_20)>. Acesso em: 16 ago. 2024.

KORTLI, Y.; JRIDI, M.; FALOU, A. A.; ATRI, M. **Face recognition systems: A survey**. *Sensors (Basel)*, v. 20, n. 2, p. 342, 2020. Disponível em: <<https://www.mdpi.com/1424-8220/20/2/342>>. Acesso em: 15 set. 2024.

KORTLI, Y. *et al.* **A novel face detection approach using local binary pattern histogram and support vector machine**. In: **2018 International Conference on Advanced Systems and Electric Technologies (IC\_ASET)**. [S.l.]: IEEE, 2018. Disponível em: <<https://ieeexplore.ieee.org/document/8379829>>. Acesso em: 02 dez. 2024.

KRONBERGER, G.; KOMMENDA, M.; AFFENZELLER, M. **Overfitting detection and adaptive covariant parsimony pressure for symbolic regression**. In: **Proceedings of the 13th annual conference companion on Genetic and evolutionary computation**. [S.l.: s.n.], 2011. p. 631–638. Disponível em: <<https://dl.acm.org/doi/abs/10.1145/2001858.2002060>>. Acesso em: 09 jul. 2024.

- LI, J. **Labelled faces in the Wild (LFW) dataset**. 2018. Disponível em: <<https://www.kaggle.com/datasets/jessicali9530/lfw-dataset>>. Acesso em: 10 jun. 2024.
- LÓPEZ, L. S. **Local Binary Patterns applied to Face Detection and Recognition**. Universitat Politècnica De Catalunya, 2010. Disponível em: <<https://upcommons.upc.edu/entities/publication/0f6025a5-2e3d-49cf-a768-16cc5bc12806>>. Acesso em: 21 nov. 2024.
- MEDEIROS, L. X.; CARRIJO, G. A.; FLORES, E. L.; VEIGA, A. C. P. **Genetic algorithms applied in face recognition**. *IEEE Latin America Transactions*, IEEE, v. 10, n. 6, p. 2280–2285, 2012. Disponível em: <<https://ieeexplore.ieee.org/document/6418133>>. Acesso em: 13 jul. 2024.
- MENDES, K. B. **Comparativo de algoritmos clássicos de aprendizado de máquina em um problema de reconhecimento de faces**. Universidade Federal de Uberlândia, 2019. Disponível em: <<https://repositorio.ufu.br/handle/123456789/28614>>. Acesso em: 11 set. 2024.
- MOREIRA, G. R. **Reconhecimento facial: um estudo comparativo sobre métodos cruzados de PCA e LDA**. Universidade Federal de São Carlos, 2021. Disponível em: <<https://repositorio.ufscar.br/items/2071efc7-e301-45a3-9cfb-3b18852e1c45>>. Acesso em: 08 ago. 2024.
- MOUSAVI, S.; CHARMI, M.; HASSANPOOR, H. **A distinctive landmark-based face recognition system for identical twins by extracting novel weighted features**. *Computers & Electrical Engineering*, Elsevier, v. 94, p. 107326, 2021. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0045790621002986>>. Acesso em: 17 out. 2024.
- MOUSAVI, S.; CHARMI, M.; HASSANPOOR, H. **Recognition of identical twins based on the most distinctive region of the face: Human criteria and machine processing approaches**. *Multimedia Tools and Applications*, Springer, v. 80, n. 10, p. 15765–15802, 2021. Disponível em: <<https://link.springer.com/article/10.1007/s11042-020-10360-3>>. Acesso em: 04 dez. 2024.
- NAPOLÉON, T.; ALFALOU, A. **Pose invariant face recognition: 3D model from single photo**. *Optics and Lasers in Engineering*, Elsevier, v. 89, p. 150–161, 2017. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0143816616301245>>. Acesso em: 06 ago. 2024.
- OPENCV.ORG. **OpenCV: OpenCV modules**. 2024. Disponível em: <<https://docs.opencv.org/4.x/index.html>>. Acesso em: 04 dez. 2024.
- ORVALHO, V. **Reconhecimento facial**. *Revista de Ciência Elementar*, Casa das Ciências, v. 7, n. 4, 2019. Disponível em: <<https://rce.casadasciencias.org/rceapp/art/2019/073/>>. Acesso em: 09 set. 2024.
- PADILHA, A. J.; BARBOSA, J.; MORAIS, C. N. **Simulation of data distribution strategies for LU factorization on heterogeneous machines**. In: *Proceedings International Parallel and Distributed Processing Symposium*. [S.l.]: IEEE Comput. Soc., 2003. Disponível em: <<https://ieeexplore.ieee.org/document/1213211>>. Acesso em: 11 jul. 2024.
- PAPAGEORGIOU, C. P.; OREN, M.; POGGIO, T. **A general framework for object detection**. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. [S.l.]: Narosa Publishing House, 2002. Disponível em: <<https://ieeexplore.ieee.org/document/710772>>. Acesso em: 26 ago. 2024.

PAYAL, P.; GOYANI, M. M. **A comprehensive study on face recognition: methods and challenges.** *The Imaging Science Journal*, Taylor & Francis, v. 68, n. 2, p. 114–127, 2020. Disponível em: <<https://www.tandfonline.com/doi/abs/10.1080/13682199.2020.1738741>>. Acesso em: 13 dez. 2024.

PEREIRA, R. C. **Técnica de rastreamento e perseguição de alvo utilizando o algoritmo Haar Cascade aplicada a robôs terrestres com restrições de movimento.** 2017. Disponível em: <<https://repositorio.ufrn.br/server/api/core/bitstreams/6e3d37d5-32ce-4c2b-bbac-eeb09bb4ca88/content>>. Acesso em: 24 jul. 2024.

PETROU, M. C. **Image processing: The fundamentals.** 2. ed. Hoboken, NJ: Wiley-Blackwell, 2010. Disponível em: <[https://www.researchgate.net/publication/267127313\\_Image\\_processing\\_The\\_fundamentals\\_With\\_CD-ROM\\_2nd\\_ed](https://www.researchgate.net/publication/267127313_Image_processing_The_fundamentals_With_CD-ROM_2nd_ed)>. Acesso em: 28 set. 2024.

PRADO, K. S. d. **Comparação de técnicas de reconhecimento facial para identificação de presença em um ambiente real e semicontrolado.** Universidade de São Paulo, 2018. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/100/100131/tde-07012018-222531/en.php>>. Acesso em: 27 out. 2024.

RANJAN, R. *et al.* **A fast and accurate system for face detection, identification, and verification.** *IEEE Trans. Biom. Behav. Identity Sci.*, v. 1, n. 2, p. 82–96, 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8680708>>. Acesso em: 01 ago. 2024.

REFAEILZADEH, P.; TANG, L.; LIU, H. **Cross-validation.** *Encyclopedia of database systems*, Springer US, v. 5, p. 532–538, 2009. Disponível em: <[https://link.springer.com/rwe/10.1007/978-0-387-39940-9\\_565](https://link.springer.com/rwe/10.1007/978-0-387-39940-9_565)>. Acesso em: 24 nov. 2024.

REINOSO, L. L. R.; BRESSAN, G. **A real-time face analysis system for head poses classification.** Dissertação (Mestrado) – Universidade de São Paulo, São Paulo, 2024. Disponível em: <<https://www.teses.usp.br/teses/disponiveis/3/3141/tde-26032024-112307/pt-br.php>>. Acesso em: 02 nov. 2024.

ROSEBROCK, A. **Local binary patterns with python & opencv.** 2015. Disponível em: <<https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/>>. Acesso em: 23 dez. 2024.

SARAGIH, R. E.; TO, Q. H. *et al.* **A survey of face recognition based on convolutional neural network.** *Indonesian Journal of Information Systems*, v. 4, n. 2, 2022. Disponível em: <<https://ojs.uajy.ac.id/index.php/IJIS/article/view/5439>>. Acesso em: 29 ago. 2024.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval.** [S.l.]: Cambridge University Press Cambridge, 2008. v. 39. Disponível em: <<https://www.cis.uni-muenchen.de/~hs/teach/14s/ir/pdf/19web.pdf>>. Acesso em: 14 out. 2024.

SCURI, A. E. **Fundamentos da imagem digital.** Pontifícia Universidade Católica do Rio de Janeiro, p. 13, 2002. Disponível em: <[https://www2.tecgraf.puc-rio.br/ftp\\_pub/curso-cgi/Notas\\_de\\_Aula/scuri.pdf](https://www2.tecgraf.puc-rio.br/ftp_pub/curso-cgi/Notas_de_Aula/scuri.pdf)>. Acesso em: 20 dez. 2024.

SILVA, A. L. **Redução de características para classificação de imagens de faces.** Tese (Doutorado) — Master's thesis, Dissertação de Mestrado, Universidade Federal Rural do Semi . . . , 2016. Disponível em: <<https://shre.ink/ufersa>>. Acesso em: 10 dez. 2024.

SILVA, B. R. d. A. I. **Sistema de contagem automática de objetos utilizando processamento digital de imagens em dispositivos móveis.** Master's thesis. Universidade do Estado do Rio Grande do Norte, 2014. Disponível em: <<https://ufersa.edu.br/wp-content/uploads/sites/42/2014/09/bruno-ramon-de-almeida-silva.pdf>>. Acesso em: 26 set. 2024.

SILVA, S. L. *et al.* **Análise de comportamento e desempenho dos algoritmos computacionais de reconhecimento facial.** 2023. Disponível em: <<http://repositorio.ufc.br/handle/riufc/76319>>. Acesso em: 17 nov. 2024.

SRIMAN, K. P.; KUMAR, P. R.; NAVEEN, A.; KUMAR, R. S. **Comparison of Paul Viola-Michael Jones algorithm and HOG algorithm for face detection.** In: IOP PUBLISHING. **IOP conference series: materials science and engineering.** [S.l.], 2021. v. 1084, n. 1, p. 012014. Disponível em: <<https://ui.adsabs.harvard.edu/abs/2021MS%26E.1084a2014P/abstract>>. Acesso em: 31 jul. 2024.

SURVE, M.; JOSHI, P.; JAMADAR, S.; VHARKATE, M. **Automatic attendance system using face recognition technique.** **International Journal of Recent Technology and Engineering (IJRTE)**, v. 9, n. 1, p. 2134–2138, 2020. Disponível em: <<https://shre.ink/AutomaticAttendanceSystemUsingFaceRecognition>>. Acesso em: 19 dez. 2024.

TALAB, M. A.; AWANG, S.; ANSARI, M. D. **A Novel Statistical Feature Analysis-Based Global and Local Method for Face Recognition.** **International Journal of Optics**, Wiley Online Library, v. 2020, n. 1, p. 4967034, 2020. Disponível em: <<https://onlinelibrary.wiley.com/doi/full/10.1155/2020/4967034>>. Acesso em: 05 jun. 2024.

TURK, M.; PENTLAND, A. **Eigenfaces for recognition.** **Journal of cognitive neuroscience**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 3, n. 1, p. 71–86, 1991. Disponível em: <<https://direct.mit.edu/jocn/article/3/1/71/3025/Eigenfaces-for-Recognition>>. Acesso em: 12 nov. 2024.

VIOLA, P.; JONES, M. J. **Robust real-time face detection.** **International journal of computer vision**, Springer, v. 57, p. 137–154, 2004. Disponível em: <<https://link.springer.com/article/10.1023/b:visi.0000013087.49260.fb>>. Acesso em: 18 dez. 2024.

WEISS, S. M.; KULIKOWSKI, C. A. **Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems.** [S.l.]: Morgan Kaufmann Publishers Inc., 1991. Disponível em: <<https://dl.acm.org/doi/abs/10.5555/102700>>. Acesso em: 22 ago. 2024.

YING, X. **An Overview of Overfitting and its Solutions.** **J. Phys. Conf. Ser.**, v. 1168, p. 022022, 2019. Disponível em: <<https://iopscience.iop.org/article/10.1088/1742-6596/1168/2/022022/meta>>. Acesso em: 12 out. 2024.

ZHENG, J.; CHELLAPPA, R. **Video-based face recognition.** In: **Computer Vision: A Reference Guide.** [S.l.]: Springer, 2021. p. 1331–1336. Disponível em: <[https://link.springer.com/rwe/10.1007/978-3-030-63416-2\\_816](https://link.springer.com/rwe/10.1007/978-3-030-63416-2_816)>. Acesso em: 09 dez. 2024.

ZHENG, J.; RANJAN, R.; CHEN, C.-H.; CHEN, J.-C.; CASTILLO, C. D.; CHELLAPPA, R. **An automatic system for unconstrained video-based face recognition.** **IEEE Trans. Biom. Behav. Identity Sci.**, v. 2, n. 3, p. 194–209, 2020. Disponível em: <<https://ieeexplore.ieee.org/document/8999558>>. Acesso em: 15 jul. 2024.

ZHOU, H.; WU, J.; ZHANG, J. **Digital image processing: part II**. [S.l.]: Bookboon, 2010. Disponível em: <<https://www.cl72.org/090imagePLib/books/digital-image-processing-part-two.pdf>>. Acesso em: 22 dez. 2024.

## APÊNDICE A – TREINAMENTO DOS MODELOS

```

File - C:\Users\proadmin\Videos\TCC Reconhecimento Facial\recon_prep.py
Page 1 of 5
1 import os
2 import cv2
3 import numpy as np
4 import matplotlib.pyplot as plt
5 import itertools
6 from sklearn.metrics import confusion_matrix
7 from sklearn.metrics import classification_report
8 import time
9
10 def show_dataset(images_class, label):
11
12     plt.figure(figsize=(14,5))
13     k = 0
14     for i in range(1,6):
15         plt.subplot(1,5,i)
16         try :
17             plt.imshow(images_class [k] [ :, :, :: - 1 ] )
18         except :
19             plt.imshow(images_class[ k ], cmap='gray' )
20         plt.title( label )
21         plt.axis( 'off' )
22         plt.tight_layout()
23         k += 1
24
25
26
27 dataset_folder = "dataset/"
28
29 names = []
30 images = []
31 for folder in os.listdir(dataset_folder):
32     for name in os.listdir(os.path.join(dataset_folder,
33         folder))[ :70 ]: # limit only 70 face per class
34         if name.find(".jpg") > -1 :
35             img = cv2.imread(os.path.join(dataset_folder
36                 + folder, name))
37             images.append(img)
38             names.append(folder)
39
40 labels = np.unique(names)
41 #labels
42 for label in labels:

```

File - C:\Users\proadmin\Videos\TCC Reconhecimento Facial\recon\_prep.py  
Page 2 of 5

```
43     ids = np.where(label == 43 np.array(names))[0]
44     images_class = images[ids[0]: ids[-1] + 1]
45
46 face_cascade = cv2.CascadeClassifier('haarcascades/
    haarcascade_frontalface_default.xml')
47
48 def detect_face(img, idx):
49     img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
50     faces = face_cascade.detectMultiScale(img, 1.3, 5)
51     try :
52         x, y, w, h = faces[ 0 ]
53         img = img[y:y+h, x:x+w]
54         img = cv2.resize(img, (100, 100))
55     except :
56         print("Face not found in image index", i)
57         img = None
58     return img
59
60 cropped_images = []
61 for i, img in enumerate( images ) :
62     img = detect_face( img, i )
63     if img is not None :
64         cropped_images.append( img )
65     else :
66         del names[ i ]
67
68 len(cropped_images)
69
70 len(names)
71
72 for label in labels:
73     ids = np.where(label == np.array( names ))[ 0 ]
74     images_class = cropped_images[ ids [ 0 ]: ids [ -1 ] + 1]
75
76
```

File - C:\Users\proadmin\Videos\TCC Reconhecimento Facial\recon\_prep.py  
Page 3 of 5

```
84 name_vec = np.array([np.where(name == labels)[0][0] for
    name in names])
85
86
87 model = cv2.face.EigenFaceRecognizer_create(45,9000)
88 #model = cv2.face.FisherFaceRecognizer_create(25,2000)
89 #model = cv2.face.LBPHFaceRecognizer_create(4,4,6,6,50)
90 start_time = time.time()
91 print("Treinando Modelo...")
92 model.train(cropped_images, name_vec)
93
94 model.save("eigenface_model.xml")
95 model.read("eigenface_model.xml")
96 #model.save("fisherface_model.xml")
97 #model.read("fisherface_model.xml")
98 #model.save("lbph_model.xml")
99 #model.read("lbph_model.xml")
100 end_time = time.time()
101
102 test_folder = "test/"
103 actual_names = []
104 predicted_names = []
105 confidences = []
106 for filename in os.listdir(test_folder):
107     if filename.find(".jpg") > -1:
108         path = os.path.join(test_folder, filename)
109
110         img = cv2.imread(path)
111         img = detect_face(img, 0)
112         if img is not None:
113             idx, confidence = model.predict(img)
114             actual_names.append(np.where(filename[:-9] == labels)[0][0])
115             predicted_names.append(idx)
116             confidences.append(confidence)
117         else:
118             print(f"Face not detected in image: {filename}")
119
120     assert len(actual_names) == len(predicted_names
121 ), "Desalinhamento entre rótulos reais e previstos"
122
```

## Trecho do código para à avaliação dos modelos

```

File - C:\Users\proadmin\Videos\TCC Reconhecimento Facial\recon_prep.py
123 # confusion_matrix
124
125 def plot_confusion_matrix(cm, classes,
126                          normalize=False,
127                          title='Confusion matrix',
128                          cmap=plt.cm.Blues):
129     if normalize:
130         cm = cm.astype('float') / cm.sum(axis=1)[:, np.
131             newaxis]
132
133     plt.figure(figsize=(6, 6))
134     plt.imshow(cm, interpolation='nearest', cmap=cmap)
135     plt.title(title)
136     plt.colorbar()
137     tick_marks = np.arange(len(classes))
138     plt.xticks(tick_marks, classes, rotation=45)
139     plt.yticks(tick_marks, classes)
140
141     fmt = '.2f' if normalize else 'd'
142     thresh = cm.max() / 2.
143     for i, j in itertools.product(range(cm.shape[0]),
144         range(cm.shape[1])):
145         plt.text(j, i, format(cm[i, j], fmt),
146             horizontalalignment="center",
147             color="white" if cm[i, j] > thresh else "black")
148
149     plt.tight_layout()
150     plt.ylabel('True label')
151     plt.xlabel('Predicted label')
152     plt.show()
153
154 # Compute confusion matrix
155 cnf_matrix = confusion_matrix(actual_names, predicted_names)
156 np.set_printoptions(precision=3)
157
158 # Plot non-normalized confusion matrix plot_confusion_matrix
159 (cnf_matrix, classes=labels, normalize=False,
160
161                          title='Confusion matrix')
162
163 # classification_report
164 print("== Classification Report for Test Dataset ==\n")
165 print(classification_report(actual_names,
166                             predicted_names,
167                             target_names=labels))
168
169 training_time = end_time - start_time
170 print(f"Tempo de Treinamento: {training_time} Segundos")

```

## APÊNDICE B – TESTANDO OS MODELOS

```

1 import cv2
2 from dataset_prep import face_cascade, model, labels
3
4
5 def draw_ped(img, label, x0, y0, xt, yt, color=(255,127,0
  ), text_color=(255,255,255)):
6
7     (w, h), baseline = cv2.getTextSize(label, cv2.
FONT_HERSHEY_SIMPLEX, 0.5, 1)
8     cv2.rectangle(img,
9         (x0, y0 + baseline),
10        (max(xt, x0 + w), yt),
11        color,
12        2)
13    cv2.rectangle(img,
14        (x0, y0 - h),
15        (x0 + w, y0 + baseline),
16        color,
17        -1)
18    cv2.putText(img,
19        label,
20        (x0, y0),
21        cv2.FONT_HERSHEY_SIMPLEX,
22        0.5,
23        text_color,
24        1,
25        cv2.LINE_AA)
26    return img
27
28
29 cap = cv2.VideoCapture("George W Bush.mp4")
30 while cap.isOpened():
31     ret, frame = cap.read()
32     if ret:
33         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
34         faces = face_cascade.detectMultiScale(gray, 1.1, 5
  )
35         for (x, y, w, h) in faces:
36             face_img = gray[y:y + h, x:x + w]
37             face_img = cv2.resize(face_img, (100, 100))
38             idx, confidence = model.predict(face_img)
39             label_text = "%s (%.2f %%) " % (labels[idx],
confidence)
40
41             frame = draw_ped(frame, label_text, x, y, x +
w, y + h, color=(0, 255, 0), text_color=(50, 50, 50))
42
43             cv2.imshow('Detect Face', frame)
44         else:
45             break
46         if cv2.waitKey(10) == ord('q'):
47             break
48
49 cv2.destroyAllWindows()
50 cap.release()
51
52

```

## APÊNDICE C – CRIAR DATASET

File - C:\Users\proadmin\Videos\TCC Reconhecimento Facial\create\_dataset.py

```
1 import cv2
2
3
4 cap = cv2.VideoCapture("fabio2.mp4")
5
6 my_name = "Fabio"
7 i = 0
8 while cap.isOpened():
9     ret, frame = cap.read()
10
11     if ret:
12         cv2.imshow("Capture Photo", frame)
13         cv2.imwrite("my_face/%s_%04d.jpg" % (my_name, i),
14                 frame)
15         if cv2.waitKey(100) == ord('q') or i == 200:
16             break
17         i += 1
18 cap.release()
19 cv2.destroyAllWindows()
```