

**UNIVERSIDADE FEDERAL DE ALFENAS - UNIFAL-MG  
ESTATÍSTICA APLICADA E BIOMETRIA**

**MICHEL LINO ALVES**

**COMPARAÇÃO DE MÉTODOS DE RECONHECIMENTO FACIAL UTILIZANDO  
FISHERFACE**

**ALFENAS / MG  
2023**

**MICHEL LINO ALVES**

**COMPARAÇÃO DE MÉTODOS DE RECONHECIMENTO FACIAL UTILIZANDO  
FISHERFACE**

Dissertação apresentada ao Programa de Pós-Graduação em Estatística Aplicada e Biometria, área de concentração em Estatística Aplicada e Biometria da Universidade Federal de Alfenas-MG, como parte dos requisitos para a obtenção do título de Mestre.

Linha de Pesquisa: Modelagem Estatística e Estatística Computacional.

Orientadora: Profa. Dra. Natália da Silva Martins  
Fonseca

Coorientador: Prof. Dr. Carlos Pereira da Silva

ALFENAS / MG  
2023

Sistema de Bibliotecas da Universidade Federal de Alfenas

Alves, Michel Lino.

Comparação de métodos de reconhecimento facial utilizando Fisherface / Michel Lino Alves. - Alfenas, MG, 2023.

83 f. : il. -

Orientador(a): Natália da Silva Martins Fonseca.

Dissertação (Mestrado em Estatística Aplicada e Biometria) -  
Universidade Federal de Alfenas, Alfenas, MG, 2023.

Bibliografia.

1. Análise de Componentes Principais (ACP). 2. Abordagem segmentada.  
3. Análise de Discriminantes Lineares (LDA). 4. Normalização de imagens. I.  
Fonseca, Natália da Silva Martins, orient. II. Título.

Ficha gerada automaticamente com dados fornecidos pelo autor.

**MICHEL LINO ALVES**

**COMPARAÇÃO DE MÉTODOS DE RECONHECIMENTO FACIAL UTILIZANDO  
FISHERFACE**

O Presidente da banca examinadora abaixo assina a aprovação da Dissertação apresentada como parte dos requisitos para a obtenção do título de Mestre em Estatística Aplicada e Biometria pela Universidade Federal de Alfenas. Área de concentração: Estatística Aplicada e Biometria

Aprovada em: 21 de julho de 2023.

Profa. Dra. Natália da Silva Martins Fonseca

Presidente da Banca Examinadora

Instituição: Universidade Federal de Alfenas - UNIFAL-MG

Profa. Dra. Ana Patrícia Bastos Peixoto

Instituição: Universidade Estadual da Paraíba - UEPB

Profa. Dr. Eric Batista Ferreira

Instituição: Universidade Federal de Alfenas - UNIFAL-MG



Documento assinado eletronicamente por **Natalia da Silva Martins Fonseca, Professor do Magistério Superior**, em 31/07/2023, às 14:04, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [https://sei.unifalmg.edu.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](https://sei.unifalmg.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1042016** e o código CRC **2DE31B01**.

---

Dedico esse estudo a Deus,  
a minha mãe Ivone e minha  
esposa Doroth.

## **AGRADECIMENTOS**

Em primeiro lugar, gostaria de agradecer a Deus por me permitir chegar até aqui. Agradeço aos meus pais pela minha vida e pelo apoio em todas as situações. Agradeço também à minha esposa, Doroth, por estar sempre ao meu lado e me motivar em todos os momentos.

Gostaria de expressar minha gratidão à minha orientadora, Nátalia, pelas instruções desde o início do curso. Agradeço também ao meu coorientador, Carlos, pelos inúmeros pedidos de correção e suas contribuições valiosas por meio de seu feedback.

Sou grato à Unifal pelos excelentes professores que me proporcionaram os resultados deste estudo.

Agradeço a todos que de alguma forma contribuíram para tornar este estudo uma realidade.

Este trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

## RESUMO

Os seres humanos têm a habilidade natural de reconhecer rostos de pessoas conhecidas e identificá-las pelo nome. Esse processo de reconhecimento facial é realizado diariamente, seja ao cumprimentar pessoas conhecidas na rua pelo nome ou ao visualizar uma imagem de um rosto familiar. Embora seja uma tarefa trivial para os seres humanos, o reconhecimento facial automatizado é uma tarefa extremamente complexo para um computador executar. Ao longo dos anos, diversos autores propuseram vários algoritmos de reconhecimento facial com o objetivo de aumentar a taxa de acerto na predição. Portanto, a presente dissertação de mestrado explora duas abordagens para o algoritmo de reconhecimento facial Fisherface: (i) abordagem segmentada e (ii) realização do alinhamento da face para cada imagem do banco de faces. As duas abordagens foram testadas analisando três indicadores de desempenho: acurácia, tempo total de processamento e carga total gasta com a execução do algoritmo. Os resultados demonstraram que é essencial realizar a abordagem de alinhamento de faces para a obtenção de melhores taxas de acurácia. A abordagem segmentada necessitou de baixo consumo de memória na realização dos testes, mas apresentou menor taxa média de acurácia quando executada em um banco de faces normalizado.

Palavras-chave: Análise de Componentes Principais (ACP); Abordagem segmentada; Análise de Discriminantes Lineares (LDA), Normalização de imagens.

## **ABSTRACT**

Humans have the natural ability to recognize the faces of familiar individuals and identify them by name. This process of facial recognition occurs daily, whether when greeting known individuals on the street by name or when viewing an image of a familiar face. While it is a trivial task for humans, automated facial recognition is an extremely complex task for a computer to execute. Over the years, various authors have proposed multiple facial recognition algorithms with the goal of increasing prediction accuracy. Therefore, this master's dissertation explores two approaches to the Fisherface facial recognition algorithm: (i) a segmented approach and (ii) performing face alignment for each image in the face database. Both approaches were tested by analyzing three performance indicators: accuracy, total processing time, and the overall computational load during algorithm execution. The results demonstrated that it is essential to perform the face alignment approach to achieve better accuracy rates. The segmented approach required low memory consumption during testing but exhibited a lower average accuracy rate when executed on a normalized face database.

**Keywords:** Principal Component Analysis (PCA); Approach by parts; Linear Discriminant Analysis (LDA); Facial Recognition.

## LISTA DE FIGURAS

Figura 1 -	Ficha com as principais imagens dos presidiários.....	18
Figura 2 -	Foto dos irmãos West e William. ....	19
Figura 3 -	Etapas básicas de um processo genérico de Reconhecimento facial. ....	23
Figura 4 -	Representação matricial de uma imagem em escala cinza.....	25
Figura 5 -	Representação da imagem como um vetor. ....	29
Figura 6 -	A imagem de uma mesma pessoa com diferentes aspectos de iluminação. ....	35
Figura 7 -	Projeção do espaço com a LDA. ....	36
Figura 8 -	Redução da dimensionalidade com a ACP. ....	36
Figura 9 -	Projeção do espaço da LDA.....	37
Figura 10 -	As Fisherfaces adquiridas após aplicação da ACP e da LDA. ....	39
Figura 11 -	Comparação dos algoritmos de reconhecimento facial Eigenface, Fisherface, LBPH.....	40
Figura 12 -	Particionamento das características da face. ....	41
Figura 13 -	Fluxograma do algoritmo da Fisherface com abordagem segmentada.....	42
Figura 14 -	Erro ao predizer a classe mais semelhante.....	45
Figura 15 -	Uma face das oito classes do banco de face ORL. ....	47
Figura 16 -	Exemplo de oito classes do banco Georgia Tech Face. ....	47
Figura 17 -	Exemplo de imagem ajustada no banco de face da Georgia Tech Face.....	48
Figura 18 -	Exemplo de particionamento em $k = 5$ folds. ....	49
Figura 19 -	Representação da divisão do banco de dados ORL. ....	50
Figura 20 -	Processo de alinhamento da face pelo ângulo dos olhos.....	51
Figura 21 -	Comparação da métrica de acurácia nos testes de reconhecimento do banco de face ORL. ....	54
Figura 22 -	Comparação da métrica acurácia nos testes de reconhecimento entre Fischerface e Fisherface com abordagem segmentada no banco de face Georgia Tech Face. ....	56
Figura 23 -	Exemplo de faces dos bancos de Face Georgia Tech Face e	

	ORL. ....	57
Figura 24 -	Resultado da métrica acurácia nos testes de reconhecimento no banco de face Georgia Tech Face com faces alinhadas. ....	57
Figura 25 -	Tempo de processamento do algoritmo de reconhecimento facial Fisherface e Fisherface com abordagem segmentada no banco ORL. ....	59
Figura 26 -	Tempo de processamento do algoritmo de reconhecimento facial Fisherface e Fisherface com abordagem segmentada no banco Georgia Tech Face. ....	60
Figura 27 -	Carga de processamento do algoritmo de reconhecimento facial Fisherface e Fisherface com abordagem segmentada no banco ORL. ....	61
Figura 28 -	Carga de processamento do algoritmo de reconhecimento facial Fisherface e Fisherface com abordagem segmentada no banco Georgia Tech Face. ....	63
Figura 29 -	Comparação das métricas avaliadas. ....	64

## LISTA DE TABELAS

Tabela 1 - Critério com a divisão dos dados. ....	50
Tabela 2 - Resolução adotada para cada característica.....	52
Tabela 3 - Resultado do teste Mann-Whitney para comparação acurácia média.....	55
Tabela 4 - Resultado do teste Mann-Whitney para comparação da acurácia média entre alinhamento de faces e face original.....	58
Tabela 5 - Resultado do teste Mann-Whitney para comparação do tempo de processamento. ....	59
Tabela 6 - Resultado do teste Mann-Whitney para comparação da carga de processamento. ....	62

## LISTA DE ABREVIATURAS E SIGLAS

ACP	Análise de Componentes Principais
ACP avançado	Algoritmo Análise de Componentes Principais Avançado
ACP Segmentado	Algoritmo Análise de Componentes Principais Segmentado
CALTECH	Instituto de Tecnologia da Califórnia
COSINE	Distância baseado por ângulos
DWT	Discrete Wavelet Transform
GEORGIA TECH	Instituto de Tecnologia da Geórgia
JPGE	Joint Photographic Experts Group
LDA	Linear Discriminant Analysis
LBP	Local Binary Patterns
MIB	MebiByte
MIT	Instituto de Tecnologia de Massachusetts da cidade de Cambridge do país Estados Unidos
ORL	Olivetti Research Limited
OpenCV	Open Source Computer Vision Library
RGB	Red, Green, Blue

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	<b>17</b>
2.1	CONTEXTO HISTÓRICO	17
2.2	PROCESSO DE RECONHECIMENTO FACIAL	22
2.2.1	Visão Computacional	25
2.3	ANÁLISE DE COMPONENTES PRINCIPAIS	26
2.4	<i>EIGENFACE</i>	28
2.5	ANÁLISE DE COMPONENTES PRINCIPAIS AVANÇADO	31
2.6	ANÁLISE DE COMPONENTES PRINCIPAIS SEGMENTADO	33
2.7	<i>FISHERFACE</i>	34
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>41</b>
3.1	<i>FISHERFACE</i> COM ABORDAGEM SEGMENTADA	41
3.2	PADRÃO DE AVALIAÇÃO	44
3.3	BANCO DE FACES	46
3.3	O ALINHAMENTO DE FACE	50
<b>4</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>54</b>
4.1	ACURÁCIA	54
4.2	TEMPO DE PROCESSAMENTO	58
4.3	CARGA DE PROCESSAMENTO	61
<b>5</b>	<b>CONCLUSÕES</b>	<b>67</b>
	<b>REFERÊNCIAS</b>	<b>68</b>
	<b>APÊNDICE A – IMPLEMENTAÇÃO LINGUAGEM R</b>	<b>73</b>

## 1 INTRODUÇÃO

O reconhecimento facial automatizado é um procedimento informatizado em que, dada uma imagem de uma face humana, é desejável a identificação da pessoa a partir de um treinamento prévio das pessoas conhecidas. Esse processo funciona de maneira similar ao processo de reconhecimento dos seres humanos. A diferença é que os seres humanos conseguem facilmente reconhecer faces de humanos, independente do ambiente, das posições do rosto, do uso de disfarces. Uma das justificativas é que estes contam com uma parte do cérebro encarregada especificamente para realizar este trabalho, utilizando-se, a seu favor no processo de reconhecimento, do contexto em que a face está inserida.

Bledsoe (1964) apresenta um dos primeiros sistemas de reconhecimento facial, em que a face humana era reconhecida a partir da comparação das medidas de diversas características da face (olhos, boca, nariz, cabeça, olhos etc.). A estratégia usada apresentou alto número de erros de identificação, uma vez que pessoas diferentes podem apresentar medições das características de faces semelhantes. Mesmo assim, a pesquisa despertou interesse da comunidade científica em aprimorar a proposta de Bledsoe.

Durante muitos anos, a senha foi o método predominante para a confirmação da identidade de usuários em um sistema informatizado. Apesar de seu constante uso, é importante destacar algumas de suas fragilidades, como o fato de o sistema utilizado poder estar vulnerável a ataques *hacker*. Um exemplo clássico de ataque *hacker* que busca descobrir a senha do usuário é *chamado de* força bruta. O ataque de força bruta consiste em um processo exaustivo de tentativa e erro auxiliado por computador, com o objetivo de descobrir a senha alheia, e com isso, obter acesso privilegiado. Outro ponto é que um usuário com acesso privilegiado poderia conceder acesso a um terceiro apenas compartilhando a sua senha; uma outra prática comum é que um usuário anote a senha de acesso e guarde em um local físico, e diante disso, um indivíduo mal-intencionado poderia ter acesso ao local, e com isso obter a senha para ter acesso privilegiado.

Dentre as várias características do ser humano, a face é uma em que pode ser comparada unicamente entre várias pessoas. Consequentemente, o processo de reconhecimento facial é uma técnica biométrica que se utiliza como meio de

comparação. A palavra "biometria" vem do grego: "Bios" (vida) e "Metron" (medida). É uma tecnologia que está sendo usada para superar as deficiências do uso de senha para identificação, uma vez que se baseia em características físicas únicas do indivíduo, tais como: impressão digital, íris, face e palma da mão.

Uma característica positiva de utilizar o reconhecimento facial é o seu baixo custo de *hardware* necessário em comparação ao reconhecimento da impressão digital. Para realizar o reconhecimento pela impressão digital, é necessária a aquisição de um escâner de alta precisão para leitura das digitais que estão localizadas nas pontas dos dedos. Para o método de reconhecimento facial, no entanto, é necessário apenas o uso de uma câmera fotográfica ou *webcam*, ou até mesmo a câmera de um celular para capturar a imagem da face e iniciar o processo de reconhecimento. Outra característica do reconhecimento facial é que ela não é invasiva, isto é, diferente do reconhecimento por íris e por impressão digital, que necessitam do contato próximo com um dispositivo que realiza a leitura, para realizar o reconhecimento facial, basta somente a visualização da pessoa pela câmera.

A aplicação do reconhecimento facial tem sido utilizada em diversas áreas, tais como: identificação de passaporte, identificação e autenticação biométrica, aplicações de segurança, controle de usuários em computadores e dispositivos móveis. Dentre essas áreas, a que mais se destaca é a segurança, na qual o reconhecimento facial pode ajudar na identificação de criminosos que circulam junto às pessoas. As câmeras podem ser instaladas em vias públicas de alto tráfego e ligadas a um *software* de reconhecimento facial, no qual cada imagem capturada do sistema de vigilância será analisada e comparada em um banco de dados de faces de criminosos.

Desse modo, torna-se evidente a importância de um método de reconhecimento facial eficiente e seguro no que tange à identificação. Na literatura, existem dois algoritmos baseados em métodos estatísticos multivariado que são amplamente utilizados no processo de reconhecimento facial: *Eigenface* e *Fisherface*. Os algoritmos *Eigenface* e *Fisherface* tiveram respostas diferentes quanto à qualidade das predições em testes realizados em um mesmo banco de faces ; COSTA *et al.*, 2021).

O objetivo principal deste estudo é realizar a comparação dos indicadores de desempenho entre a abordagem original e os métodos: abordagem segmentada e alinhamento de faces, utilizando o algoritmo de reconhecimento facial Fisherface. Com isso, é desejável que os métodos propostos aumentem o poder de predição do algoritmo, diminuam o tempo total gasto com processamento e reduzam a quantidade de recursos computacionais necessários para a execução.

Para atender a esse objetivo, as seguintes tarefas serão realizadas:

- a) pesquisar, preparar e adaptar o banco de faces para testar o algoritmo de reconhecimento facial com a abordagem segmentada e o alinhamento de faces. Nesta dissertação, a face será a característica biométrica usada para o processo de distinção entre pessoas;
- b) implementar a funcionalidade para o particionamento aleatório e automatizado do banco de faces em imagens de teste e treinamento, utilizando a abordagem *k-folds*;
- c) para o banco de face *Georgia Tech Face*, realizar o alinhamento da face a partir da localização dos olhos;
- d) implementar o algoritmo *Fisherface* no R;
- e) propor o algoritmo *Fisherface* com abordagem segmentada e realizar a implementação na linguagem R;
- f) realizar a comparação dos indicadores de desempenho entre a utilização da imagem original, a abordagem de segmentação da face e o alinhamento das faces em relação aos olhos.

Esse estudo foi organizado em 5 seções. A primeira seção conceitua o que é reconhecimento facial e onde pode ser aplicado. Também são apresentados os objetivos deste estudo.

Na segunda seção, foi realizada a revisão da literatura, começando com o histórico que motivou o processo de reconhecimento facial. Em seguida, foi apresentado um processo básico de reconhecimento facial, que pode ser utilizado para qualquer algoritmo de reconhecimento facial. Por fim, foram apresentados vários algoritmos de reconhecimento facial.

Na terceira seção, foram conduzidos os experimentos para alcançar os objetivos propostos por este estudo. No início, é apresentada a metodologia dos testes de reconhecimento, os bancos de faces utilizados e o ambiente de execução dos testes.

Na quarta seção, são apresentados os resultados obtidos com a execução dos testes propostos.

Por fim, na seção 5, são apresentadas as conclusões finais.

## 2 REFERENCIAL TEÓRICO

Neste tópico, será apresentado, inicialmente, um contexto histórico da necessidade de identificação de um indivíduo a partir de uma característica biométrica (Face, Íris, impressão digital, entre outros). Apesar de todas serem características importantes para reconhecimento de pessoa, a face é objeto utilizado neste estudo para reconhecimento de pessoas.

### 2.1 CONTEXTO HISTÓRICO

Existem áreas da sociedade em que há uma maior necessidade de identificar uma pessoa automaticamente, sem intervenção humana, por meio de suas características faciais, como, por exemplo, a segurança. Diante dessa necessidade, cresce a demanda pela busca de ferramentas de reconhecimento biométrico para a identificação de pessoas.

Entretanto, essa preocupação não é algo recente, pois existem relatos históricos de 800 d.C. (depois de Cristo), em que os chineses identificavam os envolvidos com a realização de contratos de transação comercial. Os indivíduos eram identificados com a marcação das mãos em tábuas de barro (FERREIRA, 2016). Com isso, as pessoas envolvidas não poderiam alegar o desconhecimento do contrato, uma vez que a marca da mão estava gravada em um objeto. Apesar de ser muito rudimentar, é um dos primeiros relatos do uso de uma aplicação de identificação através da gravação do formato da mão.

O primeiro relato científico de um método de identificação de pessoas foi apresentado pelo pesquisado Bertillon (1893). O objetivo desse método era identificar criminosos presos. Em 1893, o pesquisador Bertillon realizou um dos primeiros estudos para a identificação de pessoas a partir das dimensões do corpo (BERTILLON, 1893). O método foi chamado de Sinalética, que se baseava em registrar a fotografia de frente e de perfil dos criminosos, detalhar textualmente as características individuais como: cor dos olhos, tatuagens, altura, peso entre outras características. Posteriormente, caso necessário, seria possível consultar detalhes das características identificadas desse criminoso ao comparar as características da ficha com as características atual, e com isso, afirmar se são da mesma pessoa.

A proposta consistia em criar fichas em papel, nas quais seriam descritas as características físicas dos indivíduos, como: dimensões das orelhas, boca, olhos, nariz, sobrancelha, comprimento dos principais ossos e características individuais, como tatuagem, cicatrizes (SPANHOL; MENDONÇA LUNARDI; VIEIRA DE SOUZA, 2016). Além das anotações das características corporais, eram registradas fotografias das faces dos criminosos de frente e de lado, conforme ilustrado na Figura 1.

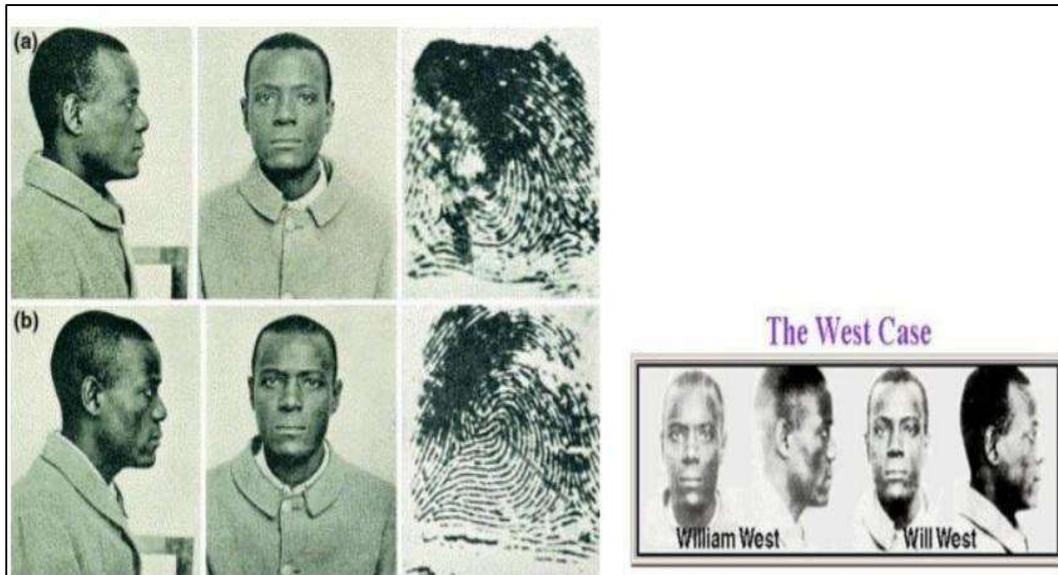
Figura 1 - Ficha com as principais imagens dos presidiários



Fonte: Scorsato (2012, p. 8)

Em 1903, o método de Bertillon foi desacreditado por terem ocorrido erros como, por exemplo, o caso dos dois irmãos norte-americanos chamados de William West e Will West, que eram gêmeos idênticos com medições corporais semelhantes quase que imperceptíveis (ROSA, 2016). Essa falha comprovou que as dimensões do corpo não apresentam dimensões e características únicas. Na Figura 2 podem-se visualizar as semelhanças entre West, representado no item (a), e seu irmão gêmeo William West, representado no item (b). Neste caso a polícia pôde diferenciá-los através de suas impressões digitais.

Figura 2 - Foto dos irmãos West e William



Fonte: Adaptado de ROSA 2016, p. 21.

Diante do problema dos irmãos William West, entre outros casos, Galton (1892) propõe a identificação da pessoa a partir da sua impressão digital. Galton publicou o livro intitulado *Impressão Digital*, no qual foi possível estabelecer a prova da unicidade e individualidade da impressão digital do ser humano, coletando mais de 8000 conjuntos de impressões em seu laboratório de antropologia (ASHBOURN, 2014).

Após os avanços na identificação de pessoas por impressão digital, surgem os primeiros estudos de identificação por reconhecimento facial. Entre eles, um que se destacou foi o estudo de Bledsoe (1964), com a técnica denominada "*machine facial recognition*". A técnica consistia em medir as dimensões das características da face, como olhos, boca, nariz e tamanho da cabeça, com o auxílio do *Rand tablet*. O *Rand tablet*, também conhecido como *Grafacon*, é um dispositivo gráfico de baixo custo, sendo o primeiro a utilizar uma caneta como instrumento altamente prático.

O sistema desenvolvido para o *Rand tablet* consistia em projetar a imagem da face em um mosaico. O usuário então marcava manualmente os pontos de cada característica, e o sistema armazenava as dimensões correspondentes. Com isso, era criado um banco de dados com o registro de todas as medições para cada face registrada.

Com o banco de dados das medições, realizava-se a identificação de uma face desconhecida, comparando suas medições com as correspondentes mais próximas armazenadas no banco de dados. A face que apresentasse o maior número de medidas semelhantes era definida como a mais semelhante.

Entretanto, o método proposto foi ineficaz devido a problemas clássicos de reconhecimento facial, como a rotação da cabeça, expressão facial e envelhecimento. Essas características impactaram nas medições e, assim, prejudicaram a verificação correta da face desconhecida (FERREIRA, 2016).

Kanade (1977) propôs um processo de automatização para auxiliar na identificação de um rosto. O método computacional localiza um rosto em uma fotografia com o objetivo de reconhecê-lo automaticamente por meio de suas características. Essas características foram utilizadas para desenvolver um *software* que buscava padrões de semelhança com base em parâmetros faciais, como o formato do nariz, a posição dos olhos, a distância entre a boca e o cabelo. O algoritmo extraía automaticamente 16 parâmetros faciais, alcançando uma taxa de acerto na identificação de aproximadamente 45% a 75% (FERREIRA, 2016).

Um marco para os estudos de reconhecimento facial foi a proposta Kirby e Sirovich (1987), que consistia em reduzir a dimensionalidade do espaço das imagens das faces por meio do método estatístico análise de componentes principais (ACP), e posteriormente, reconstruí-las para, então, realizar a comparação. Essa técnica foi denominada de *Eigenface*, e diferente das abordagens anteriores, o método proposto considera a face integralmente, sem subdividi-la, para realizar o processo de identificação. Os métodos baseados nesta proposta foram classificados como métodos holísticos (DE SOUZA, 2014).

Turk e Pentland (1991) desenvolvem um sistema computacional baseado no algoritmo *Eigenface* capaz de realizar o reconhecimento facial. O processo foi feito quase em tempo real. Este processo de reconhecimento é realizado projetando as faces em um espaço de baixa dimensionalidade, de modo, que quanto maior a diferença entre as faces, maior a dispersão. O reconhecimento facial é realizado projetando todas as imagens de faces em um espaço de faces reduzidas pela ACP. Por meio de uma métrica, é realizada a comparação da menor distância entre a projeção de uma face desconhecida em relação às demais faces conhecidas. Com

isso, as projeções de faces semelhantes estão dispostas mais próximas no espaço de face (SILVA, 2016).

Ojala, Pietikäinen e Harwood (1996) propuseram um método de reconhecimento facial baseado em classificação de texturas da imagem da face chamado de *Local Binary Patterns (LBP)*. Porém, com o tempo, passou a ser utilizado para extração de características no processo de reconhecimento e classificação de imagens faciais.

Diferente dos algoritmos que se utilizam da ACP no processo de reconhecimento facial, Belhumeur (1997) propôs um que usa a *Linear Discriminant Analysis (LDA)* com o objetivo de melhorar o desempenho dos algoritmos de reconhecimento facial, que foi denominado *Fisherface*. A ideia básica é treinar um conjunto de faces organizados por classe, em que cada classe é representada por uma pessoa. Com isso, ao realizar uma análise por *LDA*, é possível redimensionar os dados originais, para uma escala menor. Na geração desse conjunto de dados, a *LDA* busca minimizar a dispersão entre as projeções das faces de uma mesma classe (intraclasse) e maximizar a dispersão das projeções das faces de classes diferentes (interclasse). Conseqüentemente quanto maior a semelhança entre as faces, menor será a distância entre as projeções no espaço de face, e quanto mais diferentes forem as características das faces, mais distantes estarão as suas projeções (BRAGA, 2013).

Viola e Jones (2001) propuseram uma abordagem para identificação de face entre objetos na imagem. A proposta se baseia em três conceitos: representação integral de imagem, treinamento de classificadores com *AdaBoost* e criação de uma estrutura de árvore. *Adaboost* é um algoritmo de aprendizado de máquina que tem o objetivo de aumentar a velocidade de outro algoritmo de aprendizado de máquina. O algoritmo tem um ponto forte, que é a velocidade de identificação da face em uma fotografia. A alta velocidade é uma característica essencial para um método de reconhecimento facial automatizado, isto é, com processamento em tempo real.

Com o objetivo de melhorar o algoritmo de reconhecimento facial que faz uso do método ACP, Duan (2008) desenvolveu um novo algoritmo chamado de análise de componentes avançado. Um método de projeção que utiliza de cluster, na qual, considera a informação da classe no cálculo dos componentes principais, visto que a

ACP não realiza a discriminação por classe. Conseqüentemente, ao projetar as faces por classe no espaço de face, houve uma aproximação das projeções das faces de uma mesma classe enquanto distanciou as projeções das faces de classes distintas. Foram realizados testes de reconhecimento facial utilizando o método onde foi observado um aumento no número de acerto na predição em comparação com o algoritmo de reconhecimento facial baseado na ACP padrão.

Medeiros *et al.* (2012) propôs a abordagem segmentada para o processo de reconhecimento facial baseada na análise de componentes principais avançados. A proposta é utilizar o reconhecimento facial a partir de regiões faciais (testa, olhos, nariz, boca e queixo) ao invés de utilizar a face como um todo. Com o uso desta abordagem, o algoritmo de reconhecimento facial baseado na Análise de componentes principais avançando obteve maiores taxas de reconhecimento facial em comparação com o algoritmo original.

Com os avanços dos estudos biométricos na área reconhecimento de faces permitiu-se a utilização prática do reconhecimento facial no mundo real. Com resultado, a Apple, a maior empresa produtora de celular do mundo, desenvolveu um método de reconhecimento facial para incorporar em seus produtos, chamado de *FaceID* (BAQEEL; SAEED, 2019). Assim, as câmeras dos celulares, *tablets* e *notbooks* são utilizadas para identificar o rosto do proprietário. Esse procedimento é um aperfeiçoamento da ideia de Bledsoe (1964), em que uma câmera digital com um sensor de infravermelho captura a face e a projeta tridimensionalmente usando cerca de 30.000 pontos característicos. Em seguida, o mapa característico é comparado com o do verdadeiro proprietário (BAEZA ARGÜELLO *et al.*, 2021).

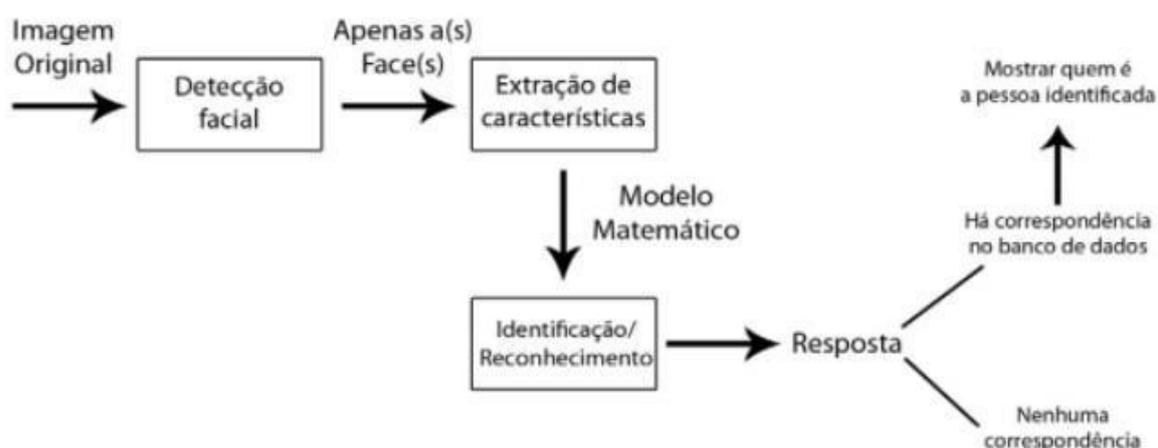
## 2.2 PROCESSO DE RECONHECIMENTO FACIAL

O objetivo do processo de reconhecimento facial é identificar uma pessoa a partir da imagem de uma face desconhecida. O processo se inicia com uma fotografia ou imagem obtida de uma gravação de vídeo. Verifica-se, na imagem de entrada, que o processo contém características de uma face humana. A face encontrada é isolada de todo o cenário da fotografia. Na última etapa do processo, aplica-se um método de reconhecimento facial o qual relaciona a face desconhecida à face de uma pessoa armazenada que contenha o maior número de características

semelhantes. O processo tem êxito quando a pessoa supostamente reconhecida pelo método é a mesma pessoa da face desconhecida.

Na figura 3 é ilustrado um processo genérico de reconhecimento facial automatizado, o qual é subdividido em três etapas: i) detecção facial, ii) extração de características e iii) identificação/reconhecimento (MARQUÉS, 2010).

Figura 3 - Etapas básicas de um processo genérico de Reconhecimento facial.



Fonte: BRAGA 2013, p. 23.

A primeira etapa do processo (Figura 3) é a detecção facial em que, numa dada a imagem (fotografia) original, verifica-se que existe pelo menos uma face na imagem dada. Um algoritmo é utilizado para que, dada uma imagem de entrada, ele retorne com as coordenadas da região da face, no caso de encontrar uma face. Viola-Jones (2001) é um exemplo de algoritmo de detecção de face amplamente usado. Esta etapa é importante, uma vez que, com o recorte da face identificada na imagem, eliminam-se informações desnecessárias (BRAGA, 2013).

Na segunda etapa, realiza-se a extração das características relevantes das faces identificadas (CARLOS JUNIOR, 2015, p. 9). É importante ressaltar que a execução desta etapa depende da identificação de pelo menos uma face encontrada no processo anterior. Posteriormente, é feito um filtro das características relevantes da face original, de modo que permita a um modelo matemático diferenciá-la (BRAGA, 2013). É nesta etapa em que é aplicado um algoritmo de reconhecimento facial com o objetivo de extrair somente características relevantes da face.

Por fim, é realizada a comparação entre as características obtidas da etapa anterior com as faces armazenadas em banco de dados. Na comparação, avalia-se o nível de similaridade entre as duas faces. O nível de similaridade é definido como a distância  $x$  que representa o quão similares são as faces, sendo que quanto menor a distância, maior a similaridade (BRAGA, 2013).

Yang, Kriegman e Ahuja (2002) classificam os algoritmos de reconhecimento facial em quatro categorias: métodos baseados em conhecimento, características, padrões e aparência.

- I. **Métodos baseados em conhecimento:** consistem em uma estratégia de reconhecimento a qual é realizada com base na geometria da face. Isto é, necessita-se do conhecimento das características da face humana como a posição dos olhos e a simetria deles.  
O trabalho de Yang e Huang (1994) é um exemplo de método baseado em conhecimento que se utilizava de regras hierárquicas para reconhecer regiões da face.
- II. **Métodos baseados em características:** representam as técnicas de reconhecimento que analisam as partes invariantes da face, como por exemplo a cor da pele, formato dos lábios e da bochecha.  
No trabalho de Buhiyan *et al.* (2003) utilizou-se a cor da pele para identificação da região da face.
- III. **Métodos baseados em padrões:** o reconhecimento se baseia em um conhecimento prévio da família de curva de uma face. Isto é, a partir de uma forma, procura-se na imagem uma correspondência desta forma. No caso da face este método procura um objeto com forma de uma elipse, por apresentar semelhanças com a face humana (LOPES, 2005).
- IV. **Métodos baseados na aparência:** esses métodos baseiam-se em algoritmos de aprendizado. A partir de várias imagens treina-se o modelo com o objetivo de ensiná-lo a reconhecer faces. Nestes métodos não é necessário o conhecimento a priori sobre o objeto ou a característica que se deseja comparar. Este método é considerado o mais bem-sucedido na tarefa de reconhecimento facial. Exemplos de utilização desse método: *Eigenface* e *Fisherface* (LOPES, 2005).

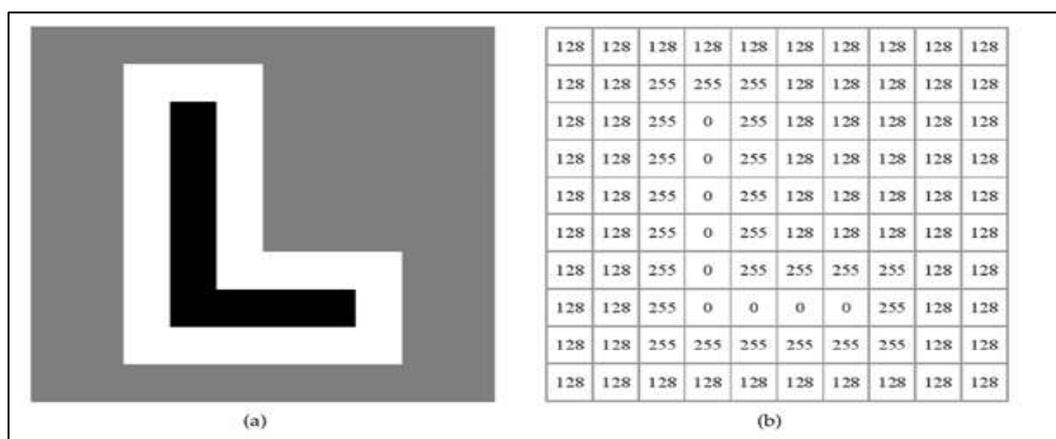
A entrada do processo de reconhecimento facial é uma imagem de face de uma pessoa, sendo que esta imagem é representada computacionalmente como uma matriz de *pixel*, que é um conceito relacionado à área de visão computacional.

### 2.2.1 Visão Computacional

Crowley e Christensen (1995) definem visão computacional como sendo um ramo de estudo de processamento de imagens do mundo real executadas por um computador. Com isso, a visão computacional consegue replicar determinadas funções da visão dos seres humanos por meio de software, como: identificação de objetos em imagens, carros autônomos, sistemas de reconhecimento facial, entre outros.

Na visão computacional, uma imagem digital é representada como uma função bidimensional, em que as coordenadas espaciais  $(x, y)$  fornecem intensidade de cor naquele ponto (MEDEIROS, L X *et al.*, 2012). A função  $F(x, y)$  pode ser representada como uma matriz, em que as coordenadas  $x$  e  $y$  representam, respectivamente, as linhas e colunas da matriz. Os elementos da matriz são chamados de *pixel*, que representa o menor ponto quadrado ao qual é possível atribuir uma cor (MACHADO; SOUKI, 2004). A palavra *pixel* é formada pela combinação das palavras “*picture*” e “*element*”, ou seja, “elemento da imagem” sendo a unidade de medida padrão para imagens digitais.

Figura 4 - Representação matricial de uma imagem em escala cinza



A Figura 4 ilustra a representação de uma imagem monocromática da letra L (a) de tamanho de 10 pixels de altura por 10 pixels de largura. No lado (b) é apresentada a matriz dos pixels da imagem de (a), cada elemento da matriz representa uma cor da imagem.

Como a imagem é monocromática, temos que cada elemento da matriz assumirá um valor do intervalo entre 0 para uma cor completamente escura (preto), até 255 para uma cor totalmente clara (branco). A escala de variação da cor para cada elemento da matriz representado pela Figura 4 (b) é chamado de quantização.

A quantização representa a escala de variação da intensidade da luminosidade, e é expressa por valores reais e positivos. No contexto de sistemas digitais, a quantização é comumente representada como potências de dois. Por exemplo, uma imagem monocromática com uma resolução de 8 bits (padrão) oferece ( $2^8$ ) 256 níveis de cores, variando de branco ao preto (CUNHA, 2013).

Em imagens coloridas, não se tem apenas um valor de cor, mas sim um vetor de 3 valores, em que cada elemento representa uma cor: vermelho, verde e azul. Isso resulta em um total de 24 bits, o que equivale 16 milhões de cores (MEDEIROS, L X *et al.*, 2012)

A *Open Source Computer Vision Library (OpenCV)* é uma biblioteca de código fonte aberto desenvolvida pela Intel, que oferece funções para visão computacional e aprendizado de máquina. A Biblioteca possui mais de 2.500 algoritmos otimizados para visão computacional e inteligência (CUNHA, 2013). A biblioteca *OpenCV* oferece operações fundamentais de visão computacional, como translação, rotação, alteração de resolução, alteração do sistema de cores das imagens e representação da imagem como matrizes de pixels, entre outras funcionalidades.

A biblioteca foi desenvolvida em C++, linguagem de baixo nível que oferece melhor desempenho, e também está disponível para as linguagens C, Java e *Python*, podendo ser executado em qualquer sistema operacional.

### 2.3 ANÁLISE DE COMPONENTES PRINCIPAIS

Johnson (2007) definiu a Análise de Componentes Principais como uma estrutura de variância-covariância de um conjunto de variáveis por meio de pequenas combinações lineares dessas variáveis, com o objetivo de reduzir e interpretar os dados. Em outras palavras, a partir da ACP, um conjunto inicial de

dados com  $p$  variáveis é reduzido para  $k$  componentes principais não correlacionados, sem perda significativa de informação. Essas combinações lineares são chamadas de componentes principais (HONGYU; SANDANIELO; JUNIOR, 2016). Os componentes principais dependem apenas da matriz de covariância (ou matriz de correlação).

A ACP é utilizada em várias áreas do conhecimento para análise de dados, incluindo aplicações de reconhecimento facial. Para utilizar a ACP em estudos de reconhecimento facial, é necessário calcular os componentes principais projetando os vetores de imagem das faces no espaço gerado pelos autovetores da matriz de covariância do conjunto de imagens (SILVA, Alexandre Fieno Da, 2006).

A definição do autovetor de uma matriz é a seguinte:

Dada uma matriz  $A_{(n \times n)}$ , então seja um vetor  $\vec{x}$  não nulo em  $R^n$  é chamado autovetor de  $A$  (ou do operador matricial  $T_A$ ) se  $A\vec{x}$  for múltiplo escalar de  $\vec{x}$ , isto é:

$$A\vec{x} = \lambda\vec{x}. \quad (2.3.1)$$

Com algum escalar  $\lambda$ , denominado autovalor da matriz  $A$  ou  $(T_A)$ , e dizemos que  $\vec{x}$  é um autovetor associado a  $\lambda$  (ANTON, 2001). Isto implica que, no caso em que  $\vec{x}$  seja um autovetor da matriz  $A$ , a imagem resultante da multiplicação  $A\vec{x}$  não altera a direção.

Para encontrar os autovalores e autovetores de uma matriz  $A_{(n \times n)}$ , dada a relação  $A\vec{x} = \lambda\vec{x}$ , ela pode ser reescrita  $A\vec{x} = \lambda I\vec{x}$ , ou equivalente:

$$(\lambda I - A)\vec{x} = 0. \quad (2.3.2)$$

Para que  $\lambda$  seja um autovalor de  $A$ , a equação deve ter uma solução  $\vec{x}$  não nula.

Teorema: Se  $A$  for uma matriz  $n \times n$ , então  $\lambda$  é um autovalor de  $A$  se, e somente se, e somente se,  $\lambda$  satisfaz a equação:

$$\det(\lambda I - A) = 0. \quad (2.3.3)$$

Essa é a equação característica de  $A$ .

Um aspecto importante da ACP é que a distribuição de probabilidade do vetor aleatório não precisa ser necessariamente normal, mas caso o seja, os componentes principais resultantes da combinação linear terão distribuição normal (MINGOTI, 2005).

O cálculo dos componentes principais pode ser realizado considerando um vetor aleatório  $X = \{X_1, X_2, \dots, X_p\}^T$  com a matriz de covariância  $\Sigma$  com os pares de autovalores e autovetores  $(\lambda_1, \vec{e}_1), (\lambda_2, \vec{e}_2), \dots, (\lambda_p, \vec{e}_p)$ , onde  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$ .

Considere a seguinte combinação linear:

$$\begin{aligned} Y_1 &= \vec{e}_1 X = \vec{e}_{11} X_1 + \vec{e}_{12} X_2 + \dots + \vec{e}_{1p} X_p \\ Y_2 &= \vec{e}_2 X = \vec{e}_{21} X_1 + \vec{e}_{22} X_2 + \dots + \vec{e}_{2p} X_p \\ &\vdots \\ Y_p &= \vec{e}_p X = \vec{e}_{p1} X_1 + \vec{e}_{p2} X_2 + \dots + \vec{e}_{pp} X_p. \end{aligned} \quad (2.3.4)$$

Os componentes principais resultantes dessa combinação linear são  $Y_1, Y_2, \dots, Y_p$ , estes não são correlacionados, isto é  $Cov(Y_i, Y_k) = \vec{e}_i \Sigma \vec{e}_k = 0$  para  $i \neq k$ . Com isso a

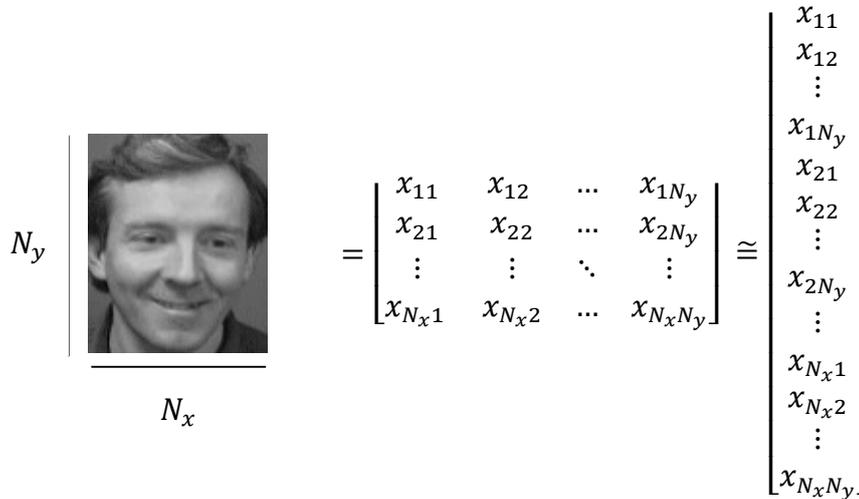
$$Var(Y_i) = \vec{e}_i \Sigma \vec{e}_i = \lambda_i \quad i = 1, 2, \dots, p. \quad (2.3.5)$$

## 2.4 EIGENFACE

Turk e Pentland (1991) propuseram o algoritmo de reconhecimento facial chamado *Eigenface*, (1991) que foi o primeiro algoritmo de reconhecimento facial a utilizar um método estatístico multivariado, a ACP, para realizar a comparação das características da face. No algoritmo proposto, não era realizada a comparação direta das faces originais, mas sim de suas projeções, que foram produzidas com a redução da dimensionalidade do espaço das faces. A proposta do algoritmo é obter uma matriz de transformação das imagens não correlacionadas das faces armazenadas no banco de imagem, realizando o reconhecimento (MEDEIROS, L X *et al.*, 2012).

A representação de uma imagem computacionalmente é definida como uma matriz de dimensões  $N_x$  e  $N_y$ , em que a largura da imagem é  $N_x$  e a altura  $N_y$ . Por exemplo, uma imagem de face tem dimensão de 256 pixels de larguras por 256 pixels de altura, então ela poderia ser representada por um vetor coluna de  $N^2 = 65.536$  posições, conforme ilustrado na Figura 5.

Figura 5 - Representação da imagem como um vetor.



Fonte: Própria.

Ao selecionar algumas imagens para treinamento de um banco de faces, representadas pelo conjunto  $X = \{X_1, X_2, \dots, X_M\}$ , onde  $M$  é a quantidade total de imagens, é importante observar que todas as faces possuem o mesmo tamanho,  $N^2 = 65.536$ , e, conseqüentemente, cada imagem de face é um ponto no espaço de 65.536-dimensional. Um conjunto de imagem é, então, mapeado nesse enorme espaço de pontos denominado espaço de face. Com isso, a ideia de utilizar a ACP para reduzir a dimensão do espaço de faces em uma estrutura mais simples, sem que sejam perdidas informações importantes.

O primeiro passo do algoritmo é calcular a média:

$$\bar{X} = \frac{1}{M} \cdot \sum_{n=1}^M X_n, \quad (2.4.1)$$

em seguida, é realizado o cálculo do desvio em relação à média de cada face:

$$\phi_i = X_i - \bar{X}, \quad i = 1, 2, 3 \dots M. \quad (2.4.2)$$

O vetor  $\phi_i$  contém todos os desvios de uma determinada face  $X_i$  em relação à média  $\bar{X}$ . Dessa forma, ao calcular os desvios em relação à média para todo o conjunto de dados, será obtida a matriz.

$$A = \{\phi_1, \phi_2, \dots, \phi_M\}, \quad (2.4.3)$$

o cálculo da matriz de covariância  $S$  é obtido pela equação 2.3.4:

$$S = \frac{1}{M} \sum_{i=1}^M (X_i - \bar{X})(X_i - \bar{X})^T = \frac{1}{M} \sum_{i=1}^M \phi_i \phi_i^T = (A \cdot A^T). \quad (2.4.4)$$

No entanto, a matriz  $S$  é grande ( $N^2 \times N^2$ ), e calcular os autovetores e os autovalores desta matriz é computacionalmente inviável. Para resolver esse problema, considere  $\vec{v}_i$  os autovetores de  $A^T A$ , de forma que:

$$A^T A \vec{v}_i = \bar{x}_i \vec{v}_i, \quad (2.4.5)$$

Pré-multiplicando ambos os lados por  $A$ :

$$A A^T A \vec{v}_i = \bar{x}_i A \vec{v}_i. \quad (2.4.6)$$

Pode-se observar que  $A \vec{v}_i$  são os mesmos autovetores da matriz de covariância  $S = A A^T$ .

Sempre que  $N^2 < M$ , é preferível obter os autovalores e autovetores de  $L = A^T A$ , uma vez que o cálculo desses autovalores e autovetores tem dimensionalidade reduzida de  $N^2$  para  $M$  (TURK; PENTLAND, 1991). No mundo real, na maioria das vezes, o tamanho do vetor de características das faces é maior que o número de faces de treinamento, portanto, normalmente é preferível utilizar esse artifício matemático para calcular os autovetores e autovalores a partir da matriz  $A^T A$ .

Assim, será obtido o  $w_i$ , que representa uma “*autoface*”, sendo escrita como combinação linear dos autovetores que representam as imagens das faces dos dados de treinamento, ou seja:

$$w_i = \sum_{j=1}^M v_{ji} \phi_j. \quad (2.4.7)$$

Depois de realizar os cálculos de todas *autofaces*, projeta-se o conjunto de faces no espaço denominado subespaço ACP, através da relação:

$$w_i = w_i^T (X_i - \bar{X}), \text{ em que } i = 1, 2, 3 \dots M, \quad (2.4.8)$$

onde  $M$  é o número total de autovetores obtidos da matriz de covariância  $L = A^T A$ , e  $x_i$  representa a contribuição de cada *autoface* em relação à imagem da face de treinamento.

Para verificar se uma face de teste  $X_{teste}$  pertence a uma pessoa do banco de faces, é necessário projetar a face no espaço de faces. A face de teste  $X_{teste}$  tem dimensão  $N^2 = N_x \cdot N_y$ , e a diferença entre a face teste e a face média é calculada:

$$\phi_{teste} = X_{teste} - \bar{X}. \quad (2.4.9)$$

A projeção no subespaço da imagem de teste é dada por:

$$w_{teste} = w_i^T \phi_{teste}. \quad (2.4.10)$$

Ao projetar todas as imagens dos dados de treinamento e a imagem de teste  $X_{teste}$  no espaço, é possível realizar a busca pela imagem mais similar à imagem de teste. Para isso, utiliza-se da distância euclidiana entre todos os pesos das equações (2.4.8) e a (2.4.10).

$$\varepsilon_i = \|(w_{teste} - w_i) \|, \quad (2.4.11)$$

em que  $w_{teste}$  representa o vetor de peso resultante da projeção da imagem  $x_{teste}$  no subespaço ACP, e  $w_i$  representa o peso da imagem do banco de faces de treinamento. Por fim, a imagem teste será atribuída à classe que tiver o menor valor para  $e_i$ .

Uma desvantagem de utilizar o algoritmo *Eigenface* é que, quando é necessário adicionar uma nova imagem de face a uma classe, é necessário recalculas as matrizes de covariância e os autovetores (BRAGA, 2013). Esse inconveniente foi resolvido por Duan (2008), com a proposta do algoritmo análise de componentes principais avançado, que utiliza a informação da classe para realizar as projeções das imagens no espaço de face. Com isso, ao adicionar uma nova classe ou uma nova face de uma pessoa existente, será necessário recalculas apenas para a classe afetada.

## 2.5 ANÁLISE DE COMPONENTES PRINCIPAIS AVANÇADO

Duan (2008) propôs um algoritmo de reconhecimento facial chamado Análise de Componentes Principais Avançado. O algoritmo proposto utiliza a ACP para reduzir a dimensionalidade do espaço das faces e ainda propõe um método de projeção baseado em *clusters*, com o objetivo de agrupar as faces por classes.

Dado um banco de imagens de faces com  $K$  classes, o método ACP avançado é utilizado em um conjunto treinamento  $C = \{C_1, C_2, \dots, C_K\}$  onde  $C_i$  é definido como subconjunto de treinamento da  $i$ -ésima classe, definido como:  $X_i = \{x_1^{(i)}, x_2^{(i)}, \dots, x_{N_i}^{(i)}\}$ , sendo  $x_1^{(i)}$  a primeira face da  $i$ -ésima classe, sendo  $x_2^{(i)}$  a segunda face da  $i$ -ésima classe, e assim por diante, e  $N_i$  a quantidade de imagens de faces utilizadas no treinamento da  $i$ -ésima classe, e  $N$  é o total de imagens de faces, ou seja,  $N = N_1 + N_2 + \dots + N_K$ .

Os passos do algoritmo de reconhecimento facial são:

1. Realize o cálculo da média de cada classe ( $X_i$ ) do conjunto de imagem  $X$  ( $i = 1, 2, \dots, K$ ), utilizando a equação:

$$\bar{x}_i = \frac{1}{N} \sum_{j=1}^{N_i} x_j^{(i)}. \quad (2.5.1)$$

2. Determine a matriz de covariância  $S_{x_i}$ , para cada classe, utilizando a equação:

$$S_{x_i} = \frac{1}{N-1} \sum_{j=1}^{N_i} (x_j^{(i)} - \bar{x}_i)(x_j^{(i)} - \bar{x}_i)^T. \quad (2.5.2)$$

Resumindo, temos que a matriz de covariância é obtida por:

$$S_{x_i} = \frac{1}{N-1} \sum_{j=1}^{N_i} AA^T. \quad (2.5.3)$$

Porém a matriz  $S$  é muito grande ( $M^2 \times M^2$ ), e calcular os autovetores desta matriz é computacionalmente inviável. Utiliza-se o mesmo artifício algébrico utilizado no algoritmo *Eigenface* para o cálculo da matriz de covariância. Assim, utiliza-se  $L = A^T A$  para obter os autovetores a partir dos  $m$  maiores autovalores.

Seja a matriz  $L$  definida a partir dos autovetores da matriz de covariância a partir dos  $m$  maiores autovalores,  $L_{x_i} = \{u_1^{(i)}, u_2^{(i)}, \dots, u_m^{(i)}\}^T$ . Todos os autovetores da matriz  $L$  devem ser normalizados (TURK; PENTLAND, 1991). Com essa matriz  $L$ , projeta-se a imagem desejada em um espaço onde é possível descrever a face e, assim, realizar o processo de reconhecimento facial.

Dada uma imagem de face desconhecida  $q$ , projete-a em cada autoespaço das classes ( $i = 1, 2, \dots, K$ ), onde  $w_i$  indica a projeção da imagem teste  $q$  na  $i$ -ésima classe.

$$w_i = L_{x_i}^T (q - \bar{x}_i). \quad (2.5.4)$$

Para cada imagem da projeção  $w_i$ , realiza-se a reconstrução  $q_i = L_i w_i$  com o objetivo de comparar as faces. Para isso, utiliza-se de uma métrica de similaridade ou um classificador. Neste caso, foi utilizada a métrica mais comum, a distância euclidiana, a que determina a distância  $d$  entre o vetor característico da imagem teste  $q$  e todos os vetores das imagens reconstruídas.

$$d_i = \|q - q_i\|, \quad (2.5.5)$$

como  $i$  é definido como a  $i$ -ésima classe, o menor valor  $d_i$  indica que a imagem teste pertence à classe ( $i$ ).

Apesar de o algoritmo de reconhecimento facial utilizar a métrica euclidiana para realizar a classificação, existem outras métricas de similaridade na literatura que poderiam ser usadas para o mesmo propósito. Estudos apontam que a métrica euclidiana, uma das mais utilizadas, não é a mais eficaz para a classificação, conforme testes de reconhecimento facial realizados por Perlibakas (2004) e Borade *et al.* (2016) com algoritmo *Eigenface*. Neste estudo, foi realizada uma comparação de desempenho entre 14 métricas de classificação, incluindo a distância euclidiana. A métrica que obteve um dos melhores resultados foi a distância baseada em ângulos (cosene), que também foi apontada como a melhor métrica (PERLIBAKAS, 2004).

## 2.6 ANÁLISE DE COMPONENTES PRINCIPAIS SEGMENTADO

Medeiros *et al.* (2012) propõem uma abordagem para o algoritmo de reconhecimento facial baseado na Análise de componentes principais avançado, chamada de Análise de componentes principais segmentado (ACP Segmentado). Nesta proposta, a estratégia é dividir a face em cinco regiões (testa, olhos, nariz, boca e queixo) e aplicar o método tradicional da ACP em cada uma dessas regiões, como se fossem imagens completas. A delimitação das regiões foi feita manualmente, utilizando um editor de imagens para separar as regiões de interesse.

Cada região é uma pequena parte que compõe a face como um todo, resultando em vetores de dimensão menor. Isso, por sua vez, leva a matrizes de covariância de menor dimensão, o que contribui para simplificar o cálculo dos autovalores e autovetores. Como resultado, há um aumento significativo na velocidade do processamento do método (MEDEIROS, *at al.*, 2012).

Os passos do algoritmo ACP segmento são descritos a seguir:

1. O processo inicia-se com a segmentação das regiões das faces em 5 grupos:  $T$ ,  $O$ ,  $V$ ,  $B$  e  $Q$ , formados por imagens que representam as regiões da testa, olhos, nariz, boca e queixo, respectivamente.
2. Subdivide-se a imagem da face desconhecida  $q$  em 5 regiões:  $q_T, q_O, q_V, q_B$  e  $q_Q$ , que são imagens das regiões da testa, olhos, nariz, boca e queixo, respectivamente.

3. Aplica-se o método ACP avançado nas cinco regiões geradas no passo anterior. Assim, obtêm-se os valores das distâncias para as 5 regiões  $d_T, d_O, d_V, d_B$  e  $d_Q$ .
4. Para fins de comparação, a distância da face é dada pela seguinte equação:  $d_i = d_{T_i} + d_{O_i} + d_{V_i} + d_{B_i} + d_{Q_i}$ , onde  $i$  é a classe.
5. Assim, a imagem da face desconhecida é identificada pelo menor valor obtido entre as classes  $d_i$ , identificada no passo 4.

Medeiros *et al* (2012) realizaram diversos testes de reconhecimento facial com o banco de face *ORL*, utilizando as 20 classes disponíveis. A taxa de reconhecimento encontrada com o algoritmo proposto foi de 98,33%, valor superior ao obtido pelo método ACP avançado, que foi de 95%.

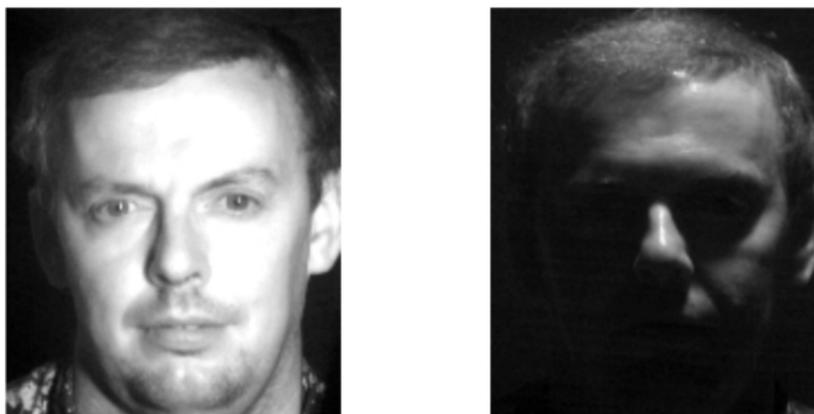
Apesar de os algoritmos de reconhecimento facial que utilizam o método estático da Análise de componentes principais terem uma boa taxa de reconhecimento em alguns testes, isto não significa garantia de poder discriminatório, uma vez que os componentes são encontrados com base na direção de maior variância e não de maior discriminabilidade (BRAGA, 2013).

## **2.7 Fisherface**

O algoritmo *Fisherface* foi proposto por Belhumeur *et al.* (1997) como uma forma de minimizar a variância dentro das classes e maximizar a variância entre classe, sendo um método útil para contornar as distorções do efeito de sensibilidade de iluminação nas expressões faciais, que é enfrentado pelo método *ACP Avançado* (PRADO, 2018).

Na Figura 6, são apresentadas duas faces da mesma pessoa, sendo que a segunda imagem da face apresenta baixa luminosidade. Para esses casos, o algoritmo *Fisherface* consegue obter melhores resultado nas predições devido à sua menor sensibilidade para efeitos de iluminação.

Figura 6 - A imagem de uma mesma pessoa com diferentes aspectos de iluminação.



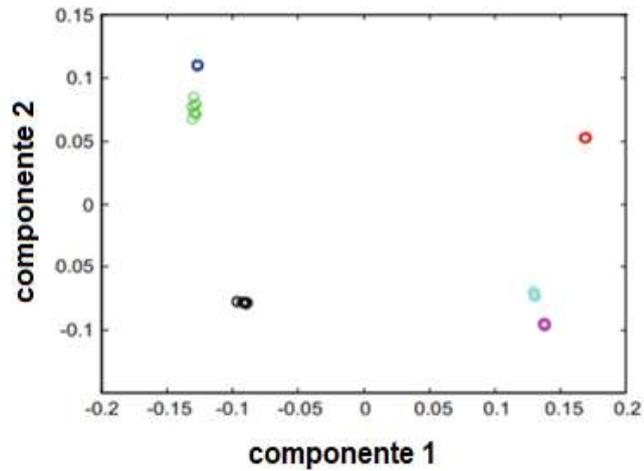
Fonte: BELHUMEUR *et al.* 1997, p. 711.

É um método estatístico usado com o objetivo de obter um subespaço  $W$  que melhor caracterize as variações intraclasse e interclasse de um conjunto de amostras das faces, com intuito de obter uma melhor capacidade discriminativa.

Para redução da dimensionalidade do espaço de face, o *Fisherface* utiliza o método estatístico *Linear discriminant analysis (LDA)*. A *LDA* foi proposta por Fisher (1936) para resolver o problema de separação e classificação de população. O trabalho de Kitani e Thomaz (2007) fundamentou que a *LDA* trabalha com as projeções das amostras em uma base dimensional e determina uma base vetorial que melhor discrimina as amostras, propondo um critério de maximização da separação entre duas ou mais classes de amostras.

A *LDA* foi proposta como uma abordagem superior à *ACP* em relação à direção da máxima variância, proporcionando um critério de discriminação que busque direções onde as classes são melhores separadas (KITANI; THOMAZ, 2007). Isso difere da *ACP*, que otimiza a extração das características com base na máxima variância de todas as amostras. Ambas as técnicas multivariadas, *ACP* e *LDA*, são usadas com o mesmo objetivo, que é reduzir a dimensionalidade do espaço de dados (CARLOS JUNIOR, 2015). A *LDA* pode ser entendida como uma técnica que busca maximizar a separação entre as classes e minimiza a dispersão dentro das classes (PRADO, 2018).

Figura 7 - Projeção do espaço com a LDA.



Fonte: GIRON-SIERRA, 2017, p. 810.

Legenda: Cada cor do círculo representa uma classe de amostra.

A Figura 7 exibe a posição das faces por classe no espaço projetado pela LDA. Em que, as cores iguais representam faces de uma mesma classe. Como esperado, a LDA afasta as faces de classes diferentes e aproxima as faces da mesma classe. Ao contrário da ACP, em que a projeção linear segue a direção da máxima variância (Figura 8), a LDA procura maximizar a variância entre classes e minimizar a variância intraclasses (Figura 9).

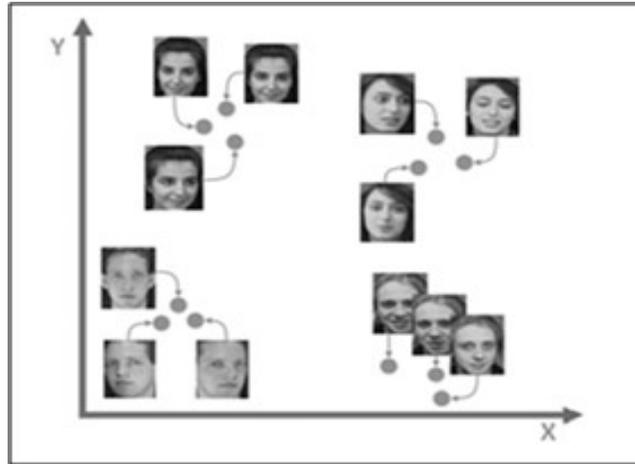
Figura 8 - Redução da dimensionalidade com a ACP.



Fonte: MENDES, 2019, p. 20

Legenda: Projeção do espaço gerado pela ACP, a linha apresenta a direção de maior variabilidade dos dados.

Figura 9 - Projeção do espaço da LDA.



Fonte: SILVA 2016, p. 33.

Legenda: Projeção do espaço gerado pela LDA, onde a projeção das faces da mesma classe está mais próxima.

Assim, todas as faces estão dispostas no espaço projetado pela *Fisherface*, seguindo a relação de que quanto mais diferente forem duas pessoas, mais distantes estarão as suas projeções, e quanto mais semelhante forem as faces de uma pessoa, mais próximas estarão suas projeções (PRADO, 2018). Como resultado, o algoritmo *Fisherface* apresenta, na maioria das vezes, uma taxa de reconhecimento superior à do método *Eigenface* (BISSI, 2018).

Considere  $N$  como a quantidade de amostras de imagens de faces  $X = \{X_1, X_2, \dots, X_K\}$ , sendo  $K$  o número de classes das amostras. O subconjunto da amostra é definido como  $X_i = \{x_1^{(i)}, x_2^{(i)}, \dots, x_{N_i}^{(i)}\}$  em que  $x_1^{(i)}$  é a primeira face da  $i$ -ésima classe,  $x_2^{(i)}$  é a segunda face da  $i$ -ésima classe, e assim por diante.

A matriz de dispersão interclasse é definida como:

$$S_B = \sum_{i=1}^K N_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T. \quad (2.7.1)$$

E a matriz de dispersão intraclasse definida como:

$$S_W = \sum_{i=1}^K \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T, \quad (2.7.2)$$

sendo  $\bar{x}_i$  a média da imagem da classe  $X_i$ , e  $N_i$  é o número de amostra da classe  $X_i$ , e  $\bar{x}$  é a média de todo o conjunto de faces.

O critério de *Fisher* define a separação entre as classes como a razão entre os determinantes das matrizes  $S_B$  e  $S_W$ , conforme equação (2.6.3).

$$Fisher_{criterion} = \frac{\det(S_B)}{\det(S_W)} = \frac{|S_B|}{|S_W|}. \quad (2.7.3)$$

Quanto maior a razão dos determinantes, maior a separação entre as classes. Portanto, a *LDA* busca uma matriz de projeção ótima, que maximize o critério de *Fisher*. A matriz de projeção ótima ( $W_{opt}$ ) é dada por:

$$W_{opt} = \arg \max_w \frac{|W^T S_B W|}{|W^T S_W W|} = [\vec{w}_1, \vec{w}_2, \dots, \vec{w}_m]. \quad (2.7.4)$$

Onde  $\{w_i | i = 1, 2, \dots, m\}$  é o conjunto dos autovetores generalizados das matrizes de dispersão  $S_w$  e  $S_B$ , correspondentes aos  $m$  maiores autovalores  $\{\lambda_i | i = 1, 2, \dots, m\}$ .

$$S_B \vec{w}_i = \lambda_i S_W \vec{w}_i, \quad i = 1, 2, 3, \dots, m. \quad (2.7.5)$$

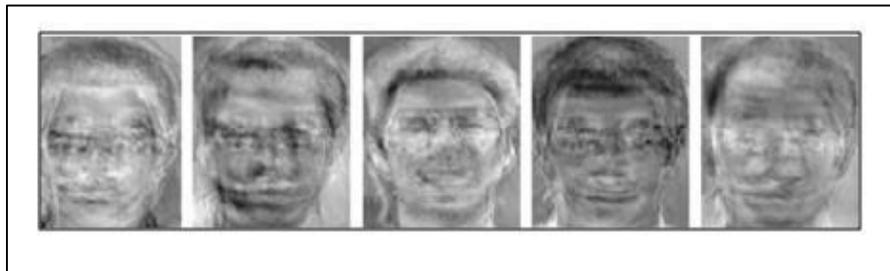
No entanto, há um problema neste processo de reconhecimento, em que a matriz de dispersão intraclasse  $S_w \in \mathbb{R}^{N \times N}$  nem sempre é não-singular. Isso ocorre por que, normalmente, o número total de imagens de treinamento é muito menor do que a dimensão resultante da concatenação do número de pixels de largura vezes o número de pixel do comprimento de cada imagem. Isso significa que pode ser escolhida uma matriz  $W$  para qual a matriz de dispersão intraclasse seja exatamente zero (BELHUMEUR; HESPANHA; KRIEGMAN, 1997).

Para contornar o problema de singularidade da matriz  $S_w$ , Belhumeur (1997) propôs uma alternativa que consistia em projetar um conjunto de imagens em um espaço de dimensão reduzida, tornando a matriz de dispersão  $S_w$  não-singular. Para isso, utiliza-se a ACP para reduzir a dimensão dos dados em  $N \times (N - K)$ , com colunas ortonormais. Com isso, o posto da matriz  $S_w$  será no máximo  $N - K$ , tornando a matriz  $S_w$  não-singular. Em seguida, a *LDA* é aplicada sobre o espaço reduzido pela ACP. Assim, a matriz de autovetores generalizados resultante terá dimensão  $N \times (N - K)$ , sendo a matriz  $W_i$  representada pelos autovetores associados aos  $K - 1$  maiores autovalores.

$$W_F = W_l W_p. \quad (2.7.6)$$

Com isso, é obtida a matriz  $W_F$  que é utilizada para projetar as faces no espaço de face da *Fisherface*. Essa matriz é chamada de '*Fisherfaces*' e contém os componentes obtidos a partir da ACP e posteriormente pela LDA. Giron-Sierra (2017) apresenta um exemplo no qual foi utilizado um banco de face de 54 faces de 6 classes, sendo que cada classe tinha igualmente 9 imagens de face. Cada face tinha dimensão fixa de 112x92, resultando em um vetor coluna com tamanho de 10304. Ao aplicar a ACP com o objetivo de reduzir a dimensionalidade dos dados e evitar o problema de singularidade de  $S_w$ , a dimensão da matriz de dados das faces foi reduzida para 48 componentes, ou seja,  $N - K$ . Em seguida, foi aplicada a LDA sobre o espaço projetado pela ACP, obtendo-se 48 autovetores generalizados, mas somente os 5 maiores foram utilizados, resultando na matriz  $W_F$ . Uma representação visual da matriz  $W_F$  está ilustrada na Figura 10.

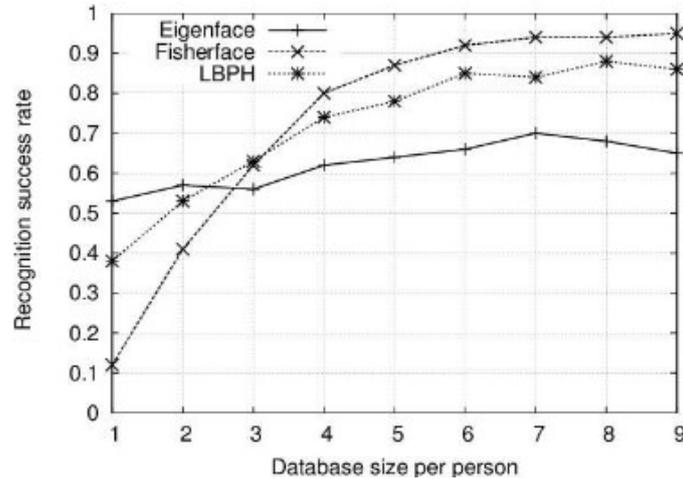
Figura 10 - As *Fisherfaces* adquiridas após aplicação da ACP e da LDA.



Fonte: GIRON-SIERRA 2017, p. 807.

Para a escolha do número mínimo de face para treinamento, Chen *et al.* (2018) realizam a comparação entre a taxa de sucesso de vários algoritmos de reconhecimento facial utilizando um número de faces de treinamento de 1 a 9. Cada simulação foi repetida 10 vezes, e em cada interação as amostras eram escolhidas aleatoriamente. Ao final dos testes de reconhecimento para todas as repetições, foi obtida a taxa média de sucesso, que é a relação entre número de acertos e o número total de testes. Os resultados foram ilustrados na Figura 11. Quando o número de imagens de treinamentos for superior a sete faces, todos os algoritmos alcançaram as melhores taxas de reconhecimento facial.

Figura 11 - Comparação dos algoritmos de reconhecimento facial *Eigenface*, *Fisherface*, *LBPH*.



Fonte: CHEN *et al.*;2018, p. 8.

Legenda: *Database size per person* - número de faces utilizadas para o treinamento.

*Recognition success rate* – taxa de sucesso de reconhecimento = número total de acerto / número total de testes de reconhecimento facial.

Em comparação com o *Eigenface*, o algoritmo *Fisherface* projeta as faces em um espaço de menor dimensionalidade. É importante destacar o estudo realizado por COSTA *et al.* (2021), que realizou uma análise comparativa de desempenho entre os métodos de reconhecimento facial *Eigenface* e *Fisherface* em relação aos bancos de faces *Caltech*, *Georgia Tech Face* e *Yale*, no qual, foi observada uma taxa de acerto superior ao algoritmo *Fisherface* em relação ao *Eigenface*. No entanto, para a base de dados *Georgia Tech Face*, o algoritmo *Fisherface* apresentou uma taxa de acerto inferior em comparação ao algoritmo *Eigenfaces*. É importante ressaltar que, embora a *LDA* seja um classificador mais robusto do que a *ACP*, existem banco de faces em que as imagens não estão alinhadas, consequentemente nesses casos o algoritmo da *Fisherface* não apresenta uma boa taxa de acerto em comparação com o algoritmo *Eigenfaces*.

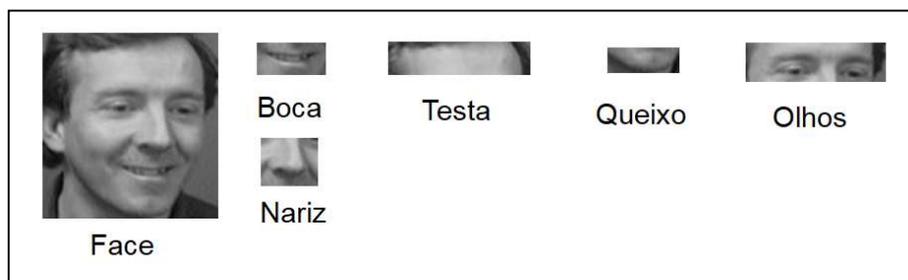
### 3 MATERIAIS E MÉTODOS

Nesta seção, será apresentada a estratégia de segmentação para o algoritmo Fisherface e o método de alinhamento da cabeça para bancos de faces que não estejam alinhados. Também serão apresentados os critérios de avaliação utilizados para a comparação do desempenho entre as estratégias para o algoritmo de reconhecimento facial *Fisherface*.

#### 3.1 FISHERFACE COM ABORDAGEM SEGMENTADA

A abordagem segmentada foi utilizada por Medeiros *et al.* (2012), na qual a imagem da face é dividida nas principais características da face (testa, olhos, nariz, boca e queixo), conforme exemplificado na Figura 12.

Figura 12 - Particionamento das características da face.



Fonte: Própria.

Legenda: Exemplo de particionamento das características de uma face do banco de face da *ORL*, com cortes das características individuais: boca, testa, queixo, olhos e nariz.

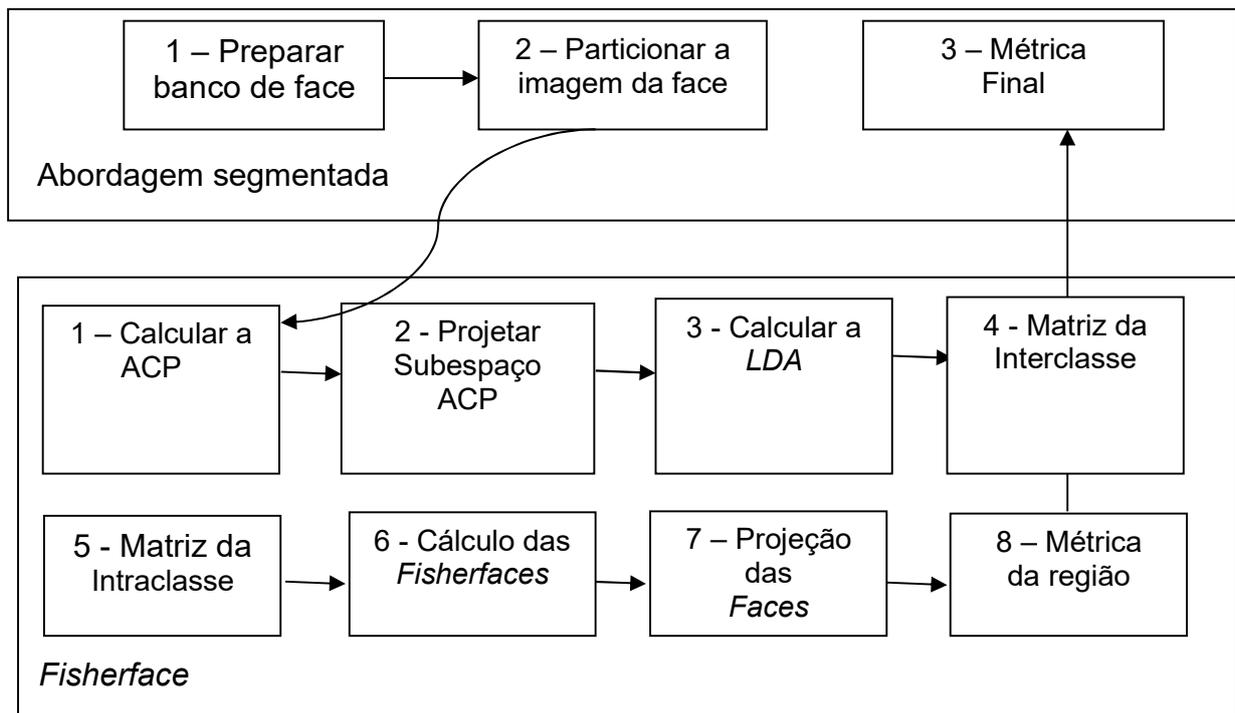
Em seguida, cada uma dessas características é submetida à análise de ACP. Como as faces são convertidas em vetores coluna, a conversão da face nessas regiões citadas proporciona a redução do tamanho dos elementos dos vetores coluna. Conseqüentemente, ocorrerá uma redução no cálculo das matrizes de dispersão da ACP e da matriz ótima da razão entre as matrizes  $S_w$  e  $S_B$  da *LDA* no cálculo dos autovetores generalizados, contribuindo para a redução do tempo de processamento do método.

Para comparar as regiões de interesse com as imagens da face no banco de dados de faces conhecidas, o processo é executado de forma interativa até que todas as distâncias de todas as regiões sejam calculadas. Os passos do processo do algoritmo *Fisherface* com abordagem segmentada são subdivididos em dois

processos: o primeiro é a separação das características da face e segundo é a execução do algoritmo da *Fisherface* para cada característica.

Na Figura 13, é apresentado um fluxograma com as sequências de atividades-chave para o processo de reconhecimento facial *Fisherface* com abordagem segmentada.

Figura 13 - Fluxograma do algoritmo da *Fisherface* com abordagem segmentada.



Fonte: Própria.

Processo de segmentação:

1. Para cada imagem de treinamento, é realizado um particionamento da imagem em cinco regiões ( $T = testa, O = olhos, V = Nariz, B = Boca, Qu = Queixo$ ), conforme ilustrado na imagem 12.
2. Uma imagem  $q$ , desconhecida, também foi segmentada nas regiões da face, formando 5 novas imagens:  $q_t, q_o, q_v, q_b, q_{qu}$ ;

Processo de execução do algoritmo *Fisherface* para cada região:

1. Cálculo da ACP – O primeiro passo do processo da *Fisherface* é reduzir o espaço de face do conjunto de dados para contornar o problema de singularidade da matriz de dispersão  $S_w$ . Utiliza-se da ACP para reduzir a dimensionalidade das faces em  $N - K$ . Com isso, a matriz  $S_w$  será sempre não-singular.

2. Projetar Subespaço ACP – Com os autovetores obtidos pela ACP, realizar a projeção das faces no subespaço gerado pela ACP.
3. Calcular a LDA – Calcula-se a média geral e a média por classe das faces no subespaço reduzido pela ACP.
  - a. Matriz da Interclasse - Calcula-se a matriz de dispersão da interclasse (equação 2.7.1).
  - b. Matriz da Intraclasse - Calcula-se a matriz de dispersão da intraclasse (equação 2.7.2).
  - c. Cálculo da *Fisherface* – A escolha da matriz de projeção ótima que maximize a razão entre o determinante das matrizes dispersão 3(a) e 3(b) é solucionado pelo conjunto generalizado dos autovetores, obtidos pela equação (2.7.4). Existirão somente  $C - 1$  autovalores diferentes de zeros. A matriz resultante será utilizada para projetar o espaço de face. Está matriz e chamada de *Fisherface*.
4. Projeção das Faces - Projetar todas as faces de treinamento no subespaço gerado pelas *Fisherfaces*.
5. Métrica de similaridade - Utilizar uma métrica de similaridade, por exemplo, a distância euclidiana para encontrar a classe do segmento da face desconhecida. Para isso, dada um segmento da face desconhecida  $s_q$ , projeta-se no subespaço da *Fisherface*. O segmento da face treinada que estiver mais próxima no subespaço, será considerada o mais similar, ou seja, ambos os segmentos de face pertencem à mesma classe.

Ao finalizar o cálculo da métrica (passo 5) para cada região da face, será obtido  $d_j = d_T^{(j)} + d_O^{(j)} + d_V^{(j)} + d_B^{(j)} + d_{Qu}^{(j)}$ , que representa a distância para a face inteira da classe  $j$ .

Para alcançar o objetivo proposto neste estudo, foi utilizada a linguagem R (R DEVELOPMENT CORE TEAM, 2009), e o ambiente de desenvolvimento *RStudio* (2023) para a implementação dos algoritmos. Os *scripts* produzidos estão disponíveis no Apêndice. Para a execução do algoritmo de reconhecimento facial, foi necessário a instalação dos pacotes *Geigen* (2009) e *OpenImageR* (2021) que são

utilizados para realizar os cálculos dos autovetores generalizados e a conversão das imagens em vetores, respectivamente. Além disso, o pacote *peakRAM* (2017) foi utilizado para avaliar as métricas do tempo e consumo de memória na execução dos algoritmos de reconhecimento facial. Com o objetivo de manter a imparcialidade na comparação dos métodos de reconhecimento facial, a execução de todos os testes de reconhecimento foram realizados na mesma máquina. Um computador com processador Intel(R) Core (TM) i7-10510U CPU 2.30 GHz com 16 GB de memória RAM e Sistema operacional Windows 10 de 64 bits [versão 10.0.22621.1265]. A biblioteca *OpenCV* foi utilizada para realizar a padronização do tamanho das imagens, a conversão de escala de cores das faces além de rotacionar imagem.

### 3.2 PADRÃO DE AVALIAÇÃO

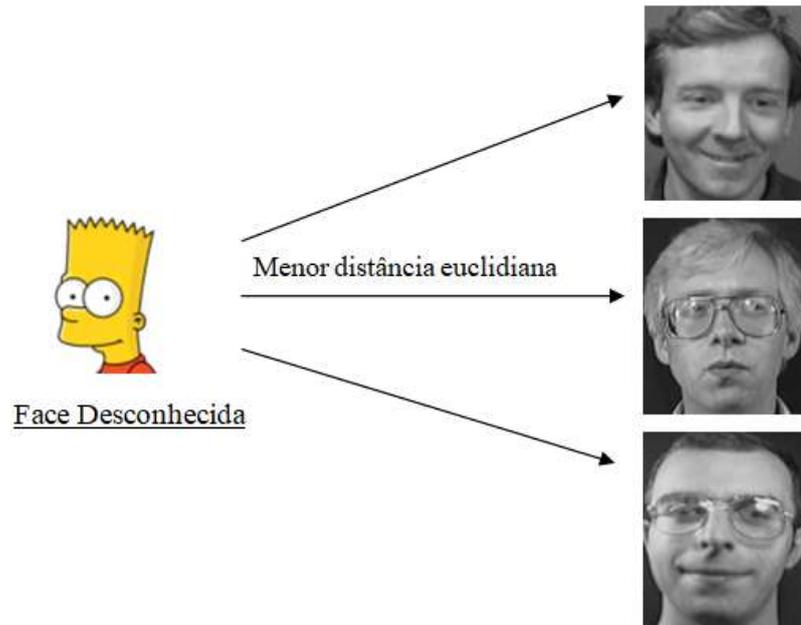
Para avaliar a qualidade de um algoritmo de reconhecimento facial, é importante entender os possíveis acertos e erros na predição. No resultado da execução do algoritmo de reconhecimento facial é possível acertar de duas formas: A primeira forma é chamado de verdadeiro positivo  $V_p$ , no qual o algoritmo identifica corretamente a classe de uma imagem desconhecida, dentre as classes que foram treinadas. A segunda possibilidade de acerto chamada de verdadeiro negativo  $V_n$ , em que o algoritmo corretamente rejeita uma face desconhecida, concluindo que a mesma não foi treinada pelo algoritmo, ou seja, não há semelhança entre a face desconhecida com nenhuma classe treinada.

O resultado da predição dos algoritmos podem estar errado de duas formas: a primeira possibilidade de errar e chamada de falso positivo  $F_p$ , em o algoritmo erra ao associar semelhança da face desconhecida com uma classe que não é verdadeiramente a sua. E a última forma possível de erro é chamado de falso negativo  $F_n$ , que ocorre quando o algoritmo de reconhecimento facial rejeita a face desconhecida, por não encontrar semelhança com nenhuma classe treinada sendo que ela é verdadeiramente de uma classe que foi treinada.

Quando o algoritmo de reconhecimento facial é testado somente com faces que pertencem a classes que foram previamente treinadas, não será necessário preocupar com o erro de falso negativo. Neste caso, ocorrerá apenas o erro na

escolha da classe correta. No entanto, ao avaliar faces de classes que podem não estar presente nas faces de treinamento, é necessário estabelecer limites para rejeitar as faces desconhecidas (Figura 14).

Figura 14 - Erro ao predizer a classe mais semelhante.



Fonte: Própria.

Para estabelecer um valor para o limiar, é necessário normalizar a distância euclidiana encontradas no resultado do algoritmo de reconhecimento facial. Com isso, as distâncias dos resultados são normalizadas para um intervalo de  $[0,100]$ , de modo que a distância máxima de uma classificação correta seja 50 e a mínima seja 0.

$$d_{x_{teste}}^{norm} = \begin{cases} 100 \times \frac{(d_{x_{teste}} - \text{Mínimo})}{2 \times (\text{Máximo} - \text{Mínimo})}, & \text{se } \text{Mínimo} < d_{x_{teste}} < \text{Máximo} \\ 0 & , \text{se } d_{x_{teste}} \leq \text{Mínimo} \\ 100 & , \text{se } d_{x_{teste}} \geq \text{Máximo} \end{cases} \quad 3.1$$

Sendo  $Q_{teste}$  a projeção da imagem de teste,  $X_{próxima}$  a projeção da imagem de treinamento disposta com a menor distância da  $Q_{teste}$ ,  $C_{próxima}$  a classe de  $X_{próxima}$  e  $d_{x_{teste}}$  a distância entre imagem de teste  $Q_{teste}$  da  $X_{próxima}$ . Na equação 3.1 é apresentada regra para normalização da distância euclidiana. Na qual o valor *Mínimo* é o valor da menor distância entre a classe da face desconhecida e a provável classe da face treinada, enquanto *Máximo* é o maior valor encontrado da

distância entre a classe da projeção da face desconhecida e a classe da face de treinamento.

Para aceitar que a face desconhecida é de uma classe que foi realizado o treinamento, é necessário estabelecer um limiar de separação, na qual será base para rejeitar a similaridade entre duas faces. É importante ressaltar que não existe uma fórmula para calcular um valor para o limiar, alguns estudos propuseram estratégia para estabelecer o valor do limiar: Gupta et. all (2010) estabeleceu o valor do limiar em 80% da maior distância normalizada, Da Silva (2015) utilizou o valor do limiar tal que despreze 5% das maiores distâncias normalizadas. Ao estabelecer o valor do limiar de separação de classe, será possível aceitar ou rejeitar a escolha da classe de uma face desconhecidas.

O aceite da classe de uma face desconhecida é considerado se:

$$d_{norm_{x_{teste}}} \leq \text{limiar}, e C_{proxima} = C_{teste}. \quad 3.2$$

Apesar da  $C_{teste}$  ser a provável classe, por apresentar a menor distância dentre as demais, com base no valor do limiar, pode ser rejeitada por apresentar uma distância acima do valor do limiar estabelecido.

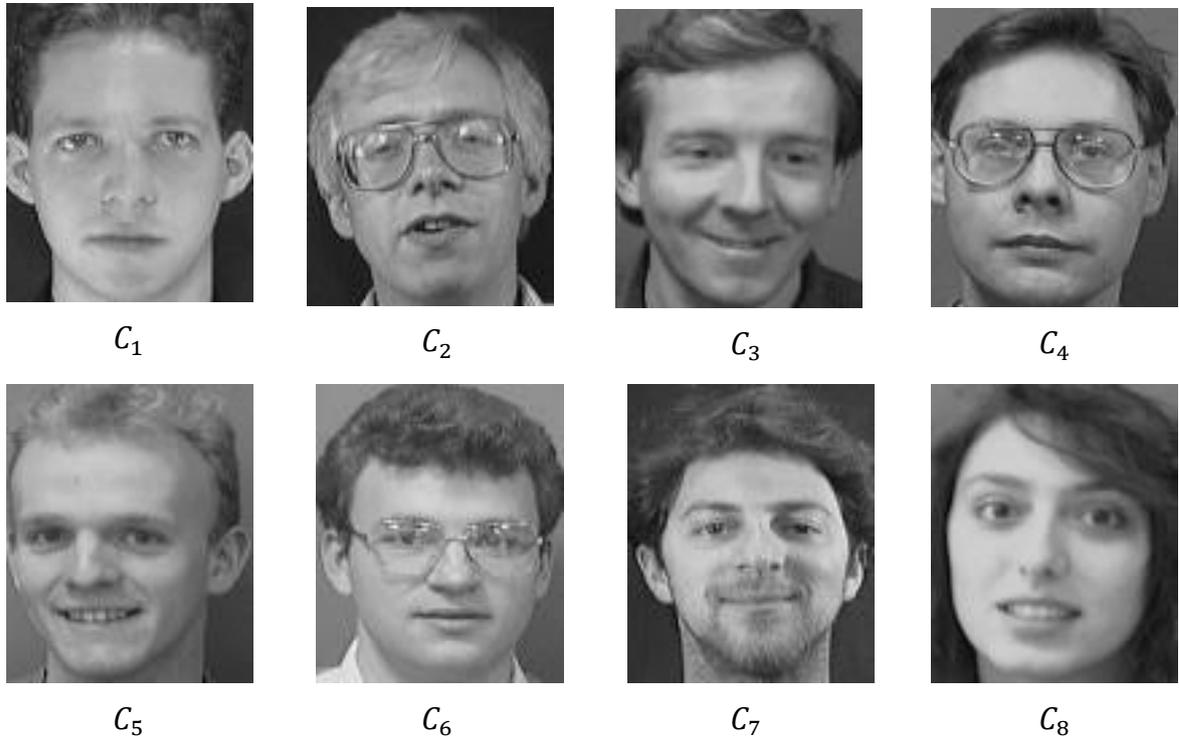
$$d_{norm_{x_{teste}}} > \text{limiar}. \quad 3.3$$

### 3.3 BANCO DE FACES

Várias instituições ao redor do mundo criaram bancos de faces para simplificar o processo de teste de algoritmos de reconhecimento facial. Cada banco de face possui características visuais distintas, além do número variável de faces por pessoas. Neste estudo, serão utilizados o banco de face *ORL* (SPEECH, 1994) e da *Georgia Tech Face* (GEORGIA TECH FACE DATABASE, 2000).

O banco de face *ORL* foi disponibilizado pela Universidade de Cambridge dos Estados Unidos. O *ORL* é um repositório de imagens de faces capturado em 1994, que contém 400 faces com 40 classes. Cada face possui uma resolução de  $92 \times 112$  pixels em escala cinza (SAMARIA; HARTER, 1994). Na Figura 15 são apresentadas oito classes ( $C_i$ ) do banco de face *ORL*.

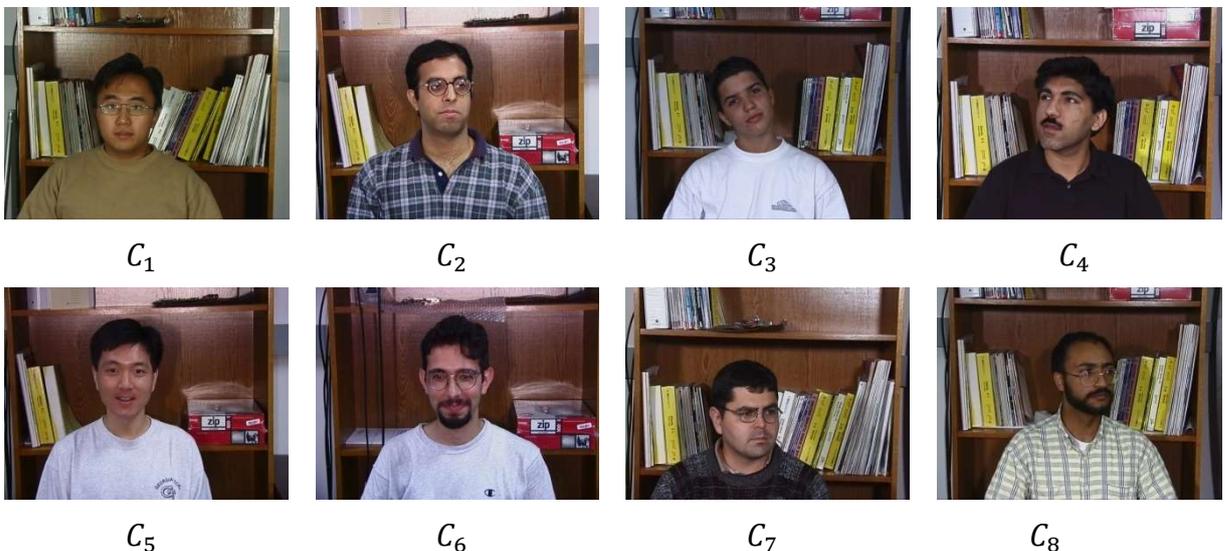
Figura 15 - Uma face das oito classes do banco de face *ORL*.



Fonte: Própria.

O *Georgia Tech Face* é um banco de dados de faces produzido pelo Instituto da Geórgia na cidade de Atlanta, nos Estados Unidos. O banco de face é composto de 15 faces de 50 pessoas, obtidas entre 01 de junho de 1999 até 15 de novembro de 1999

Figura 16 - Exemplo de oito classes do banco *Georgia Tech Face*.



Fonte: Própria.

As imagens das faces da Figura 16 estão no formato JPGE colorido, com resolução de  $640 \times 480$  pixels.

Cada face capturada apresenta diferentes condições de expressão, iluminação e posicionamento da face. Na figura 16 são apresentados exemplos de oito classes ( $C_i$ ) do banco *Georgia Tech Face*. Para realizar os testes de reconhecimento facial com os algoritmos propostos por esse estudo, é necessário que todas as faces sejam monocromáticas. Portanto, foi realizada a conversão de todas as imagens do banco *Georgia Tech Face*.

No banco de face da *Georgia Tech Face* as imagens contêm elementos que não são relacionados à face, sendo necessário isolar a face do restante da imagem. Para isso, foi realizado um corte em cada imagem do banco, removendo características irrelevantes. Na Figura 17 é ilustrado um exemplo de conversão realizado antes de executar os algoritmos de reconhecimento facial. Após realizar os ajustes em todas as imagens de face do banco de face, ficou definido-se a resolução padrão de  $233 \times 233$ .

Figura 17 - Exemplo de imagem ajustada no banco de face da *Georgia Tech Face*.



Fonte: Própria.

Para realizar a extração das características do método *LDA*, é fundamental que todas as imagens das faces utilizadas tenham a mesma dimensão, inclusive às faces que serão testadas. Caso alguma face não possua a mesma dimensão, será necessário realizar um pré-processamento na imagem, ajustando a resolução para que todas as imagens tenham a mesma medida.

Para realizar um teste de reconhecimento facial com um banco de dados de faces, primeiramente é necessário realizar a divisão das faces em dois conjuntos: o conjunto de faces de treinamento, que são as faces das pessoas identificadas do banco de face, e o conjunto de faces de teste, que são as faces de pessoas

desconhecidas pelo algoritmo de reconhecimento facial. Para realizar a divisão do banco de face, foi adotado o método estatístico validação cruzada com o método de particionamento *k-folds* (BURMAN, 1989) para criar  $k$  subconjunto de dados mutuamente exclusivo e de mesmo tamanho. Do total do subconjunto  $k$ , um deles será utilizado para realizar testes de predição, enquanto o  $k - 1$  subconjuntos restantes serão usados para treinar o algoritmo. Esse processo é repetido com a permutação do subconjunto de treinamento e teste. Ao final do processo, todos os subconjuntos terão sido utilizados para testar o algoritmo. Um exemplo de particionamento com 5 *folds* é ilustrado na Figura 18, observando que o subconjunto de teste é alternado em cada repetição (coluna). Em cada rodada, um subconjunto diferente será testado.

Figura 18 - Exemplo de particionamento em  $k = 5$  *folds*.

TESTE	TREINO	TREINO	TREINO	TREINO
TREINO	TESTE	TREINO	TREINO	TREINO
TREINO	TREINO	TESTE	TREINO	TREINO
TREINO	TREINO	TREINO	TESTE	TREINO
TREINO	TREINO	TREINO	TREINO	TESTE

Fonte: Própria.

Legenda: Cada quadrado representa um subconjunto, e cada coluna repete a execução do algoritmo de reconhecimento facial.

Para os testes de desempenhos, foram utilizadas todas as classes dos bancos de faces *ORL* e do *Georgia Tech Face*, com a criação *k-folds* aleatórios utilizando a estratégia de validação cruzada. Na Tabela 1, é apresentada a forma de divisão dos bancos de dados para realização dos testes de predição. As métricas de desempenho foram avaliadas nos bancos de faces *ORL* e *Georgia Tech Face*. Os seguintes indicadores foram mensurados: acurácia, tempo de execução do algoritmo e quantidade total de memória utilizada no processamento dos algoritmos de reconhecimento facial.

Tabela 1 - Critério com a divisão dos dados.

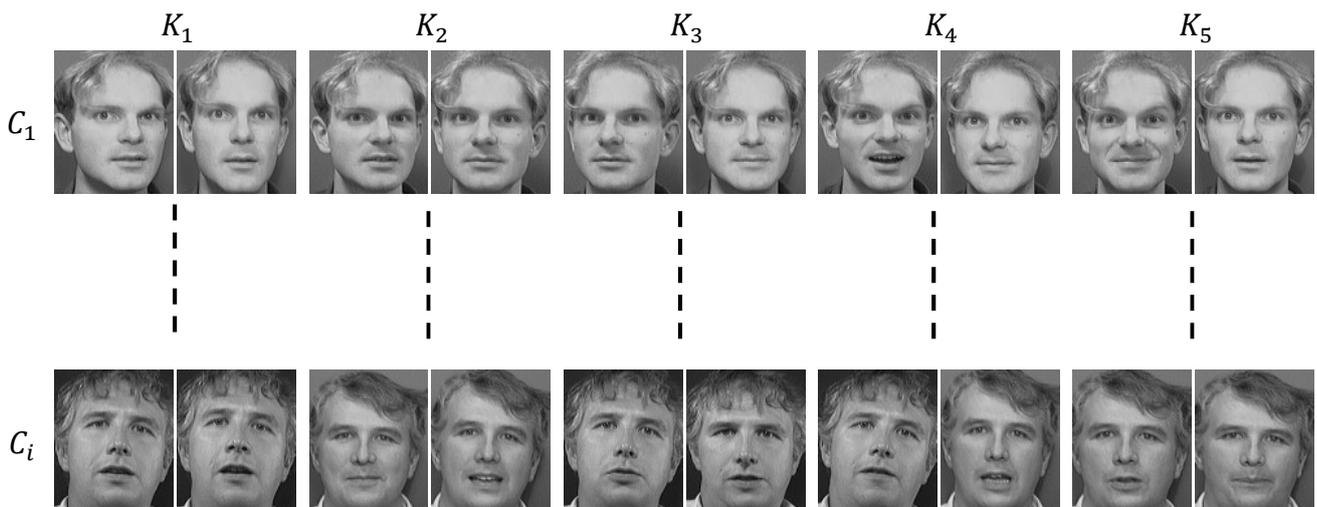
Banco de Face	Classes	Treinamento	Teste	<i>k-folds</i>	( <i>R</i> )
<i>ORL</i>	40	8 Faces	2 faces	5	3
<i>Georgia Tech Face</i>	50	10 faces	5 faces	3	5

Fonte: Própria.

Com objetivo de obter uma amostra maior de dados e obter resultados precisos de avaliação, o processo de validação cruzada foi repetido (*R*) vezes para cada banco de face. Isso resultou na geração de 15 conjuntos amostrais do mesmo conjunto de dados.

No processo de formação de cada *k-folds*, as imagens são selecionadas aleatoriamente para cada classe do banco de face. Com os *k-folds* formado, o treinamento e os testes de reconhecimento serão realizados, seguindo a proporção estabelecida na Tabela 1.

Na Figura 19, é apresentada a forma de divisão do banco de faces. Cada linha representa uma classe do banco de faces, e cada coluna representa a formação de um subconjunto do banco de faces.

Figura 19 - Representação da divisão do banco de dados *ORL*.

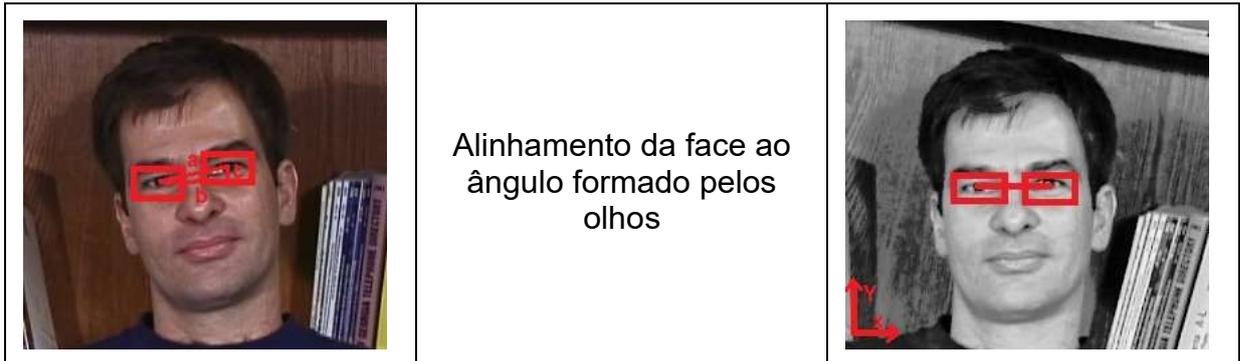
Fonte: Própria.

### 3.3 O ALINHAMENTO DE FACE

A maioria dos bancos de faces disponibiliza imagens das faces com as cabeças alinhadas, ou seja, a linha que separa o ponto central dos dois olhos está

disposta perpendicularmente, sem inclinação. No entanto, existem bancos de faces em que as faces não estão alinhadas, como é o caso do banco de faces *Georgia Tech Face*. Na Figura 20 ilustra uma imagem do banco *Georgia Tech Face* que não está alinhada.

Figura 20 - Processo de alinhamento da face pelo ângulo dos olhos.



Fonte: Própria.

Para realizar o alinhamento dos olhos, é necessário encontrar o ângulo de inclinação formado pelo triângulo  $abc$ . Para calcular o ângulo formado pela hipotenusa  $a$  e o cateto adjacente  $b$ , utiliza-se da seguinte relação:

$$\theta = \arctan \frac{c}{b}. \quad (3.4)$$

O ângulo ( $\theta$ ) encontrado é utilizado negativamente para rotacionar a imagem da face e, com isso, alinhar a imagem da face.

Para realizar o processo de alinhamento da face para o banco *Georgia Tech Face*, utilizou-se da biblioteca *OpenCV* para localizar automaticamente os olhos de cada face e calcular o ângulo (3.4). Com o ângulo obtido, é aplicado o efeito de rotação na imagem da face com ângulo negativo, e assim a face estará alinhada.

Serão utilizadas as métricas: Acurácia, tempo de processamento e carga de processamento na execução dos métodos de reconhecimento facial em ambos os bancos de faces.

Os indicadores de desempenho: acurácia, tempo de processamento para a execução do método do algoritmo de reconhecimento facial e carga de processamento, serão testados nos bancos de faces *ORL* e *Georgia Tech Face*. Com exceção do método de alinhamento de faces, que será aplicado somente no banco *Georgia Tech Face*, pois no banco de faces *ORL* todas as faces já estão normalizadas.

A medida de acuraria é definida na equação 3.5, na qual representa a proporção de acertos em relação ao total de testes realizados.

$$Acurácia = \frac{V_p + V_n}{V_p + V_n + F_p + F_n}. \quad 3.5$$

A unidade de medida para os indicadores de desempenho tempo de processamento e carga de processamento foi respectivamente obtida em segundos e MiB (Mebibyte).

Na Equação 3.5,  $V_p$  e  $V_n$  representam a quantidade de acertos ao prever a classe correta de uma face desconhecida e ao rejeitar uma face que não foi treinada pelo algoritmo de reconhecimento facial, respectivamente. Já  $F_p$  e  $F_n$  representam a quantidade de erros ao prever a classe correta na predição e ao rejeitar uma face alegando que não foi treinada, mas que na verdade foi, respectivamente.

Na Tabela 2, é apresentado o tamanho da característica para cada face após a realização do corte. A resolução representa a matriz de pixel para cada imagem da característica, em que todas as imagens do banco de face devem ter o mesmo tamanho. Na coluna "Vetor" é definido o tamanho resultante da concatenação da matriz de pixels de cada imagem.

Tabela 2 - Resolução adotada para cada característica.

(Continua)

Banco de face	Característica	Resolução (LarguraXAltura)	Vetor
ORL	BOCA	38X15	570
GEORGIA TECH	BOCA	78X36	2808
ORL	OLHOS	29X23	667
GEORGIA TECH	OLHOS	45X56	2520
ORL	NARIZ	29X23	667
GEORGIA TECH	NARIZ	45X56	2520
ORL	QUEIXO	82X25	2050
GEORGIA TECH	QUEIXO	109X36	3924
ORL	TESTA	92x34	3128

Tabela 2- Resolução adotada para cada característica.

(Conclusão)

Banco de face	Característica	Resolução (LarguraXAltura)	Vetor
GEORGIA TECH	TESTA	160X51	4784
ORL	FACE	92X112	10304
GEORGIA TECH	FACE	223X233	51959

Fonte: Própria.

Para avaliar a diferença entre as médias não paramétricas utilizou-se o teste de *Mann-Whitne* (MANN; WHITNEY, 1947) tendo em vista que não houve o atendimento das pressuposições para utilização do teste T de *Student* (1908) paramétrico.

A seguinte hipótese de teste foi formulada para comparar o desempenho entre os algoritmos de reconhecimento facial.

$$\text{Hipóteses: } \begin{cases} H_0: \bar{x}_0 - \bar{x}_1 = 0 \\ H_1: \bar{x}_0 < \bar{x}_1 \end{cases}$$

**Em que:**  $\mu_0$  é a média obtida pelo algoritmo de reconhecimento facial *Fisherface* utilizando a imagem da face original.

$\mu_1$  é a média obtida para cada método testado.

## 4 RESULTADOS E DISCUSSÃO

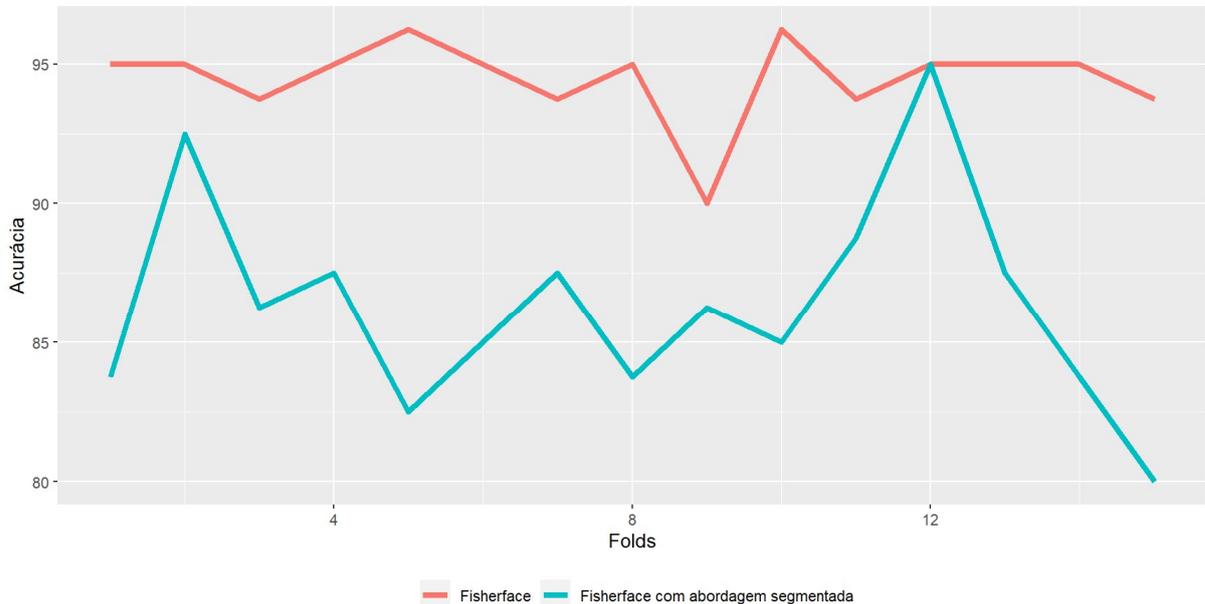
Nesta seção, serão apresentados os resultados obtidos para os métodos de reconhecimento facial: *Fisherface* utilizando as imagens originais dos bancos de faces, *Fisherface* com abordagem segmentada e o método de alinhamento de faces para o banco de faces *Georgia Tech Face*.

### 4.1 ACURÁCIA

Todas as faces separadas para realização de teste deverão pertencer a uma classe treinada pelo algoritmo. Para todos os resultados apresentados, foi definido o valor limiar de 100 para a separação de classes. Portanto, o algoritmo não rejeitará nenhuma face alegando falso negativo.

Na Figura 21, é ilustrado o resultado dos testes realizado no banco de faces *ORL* com a utilização de faces completas em todos os *folds*.

Figura 21 - Comparação da métrica de acurácia nos testes de reconhecimento do banco de face *ORL*.



Fonte: Própria.

O menor valor de acurácia encontrada para o algoritmo *Fisherface*, foi de 90%, o que representa 242 acertos em um total de 250 avaliações. A acurácia média do algoritmo de reconhecimento facial *Fisherface* com abordagem segmentada foi de 86,33%, valor inferior ao mínimo obtido pelo algoritmo *Fisherface*. A acurácia

média do algoritmo *Fisherface* foi de 94,5%, uma taxa próxima ao maior valor encontrado na abordagem segmentada, que foi de 95%.

Foi realizado um teste de comparação entre as médias da métrica de acurácia para os algoritmos *Fisherface* e a abordagem segmentada no banco de faces *ORL*. Com 95% de confiança, rejeita-se a hipótese nula  $H_0: \bar{x}_0 - \bar{x}_1 = 0$ . Com isso, constata-se que a acurácia média da abordagem segmentada apresentou desempenho inferior em relação ao algoritmo *Fisherface*. Os dados são apresentados na Tabela 3.

Tabela 3 - Resultado do teste Mann-Whitney para comparação acurácia média

Banco de faces	$\bar{x}_0$	$\bar{x}_1$	$M_0$	$M_1$	confiança	valor-p (Mann-Whitney)
<i>ORL</i>	94,50	86,33	95,00	86,25	95%	<0.00001
<i>Georgia tech Face</i>	61,84	70,29	62,80	70,40	95%	<0.00001

Fonte: Própria.

Legenda:  $\bar{x}_0$  e  $M_0$  representam, respectivamente, a média e a mediana da acurácia do *Fisherface*.  
 $\bar{x}_1$  e  $M_1$  representam, respectivamente, a média e a mediana da acurácia da abordagem segmentada.

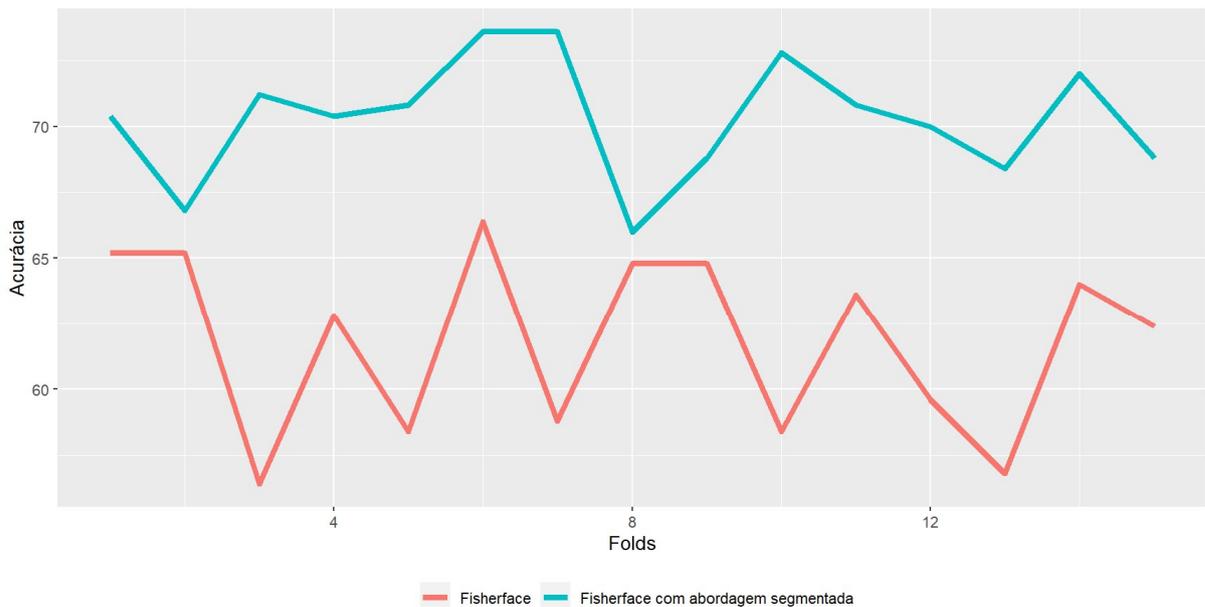
A acurácia média obtida pelo algoritmo *Fisherface* utilizando banco de face *ORL* sem modificações foi 8,17% superior ao utilizar a estratégia de segmentação. O desvio padrão e o coeficiente de variação encontrado na *Fisherface* e na *Fisherface* com abordagem segmentada foram, respectivamente, de  $\pm 1.48$  e 1,56%,  $\pm 3,79$  e 4,39%. Observa-se que a abordagem segmentada apresenta maior variabilidade na média da acurácia em comparação com a *Fisherface* original para o banco de faces *ORL*.

A baixa taxa de acurácia da abordagem segmentada no banco *ORL* pode ser explicado pelo fato de que, ao comparar características individuais, há uma diminuição da distinção entre elas. Por exemplo, pode haver várias imagens de testas calvas para característica “testa”, o que leva o algoritmo a cometer erros e resultar em uma taxa de acerto final degradada.

Na Figura 22 são apresentados os resultados para o banco de faces *Gorgia tech face*, a taxa média da acurácia do algoritmo da *Fisherface* nos testes com

banco *Georgia Tech Face* foi de 61,84%. A menor taxa encontrada para o algoritmo da *Fisherface* foi de 56,40%, que representa 141 acertos na predição para um total de 250 faces de testes.

Figura 22 - Comparação da métrica acurácia nos testes de reconhecimento entre *Fischerface* e *Fisherface* com abordagem segmentada no banco de face *Georgia Tech Face*.



Fonte: Própria.

Realizando a comparação entre as médias da acurácia para os algoritmos *Fisherface* e a Abordagem Segmentada no banco de face *Georgia Tech Face*, foi decidido rejeitar a  $H_0: \bar{x}_0 - \bar{x}_1 = 0$ , com 95% de confiança. A taxa média da acurácia obtida pelo algoritmo *Fisherface* com abordagem segmentada foi superior, com um aumento de 8,45%, em comparação com a predição com a imagem de face sem modificação, conforme apresentado na Tabela 3.

O desvio padrão e o coeficiente de variação encontrados na *Fisherface* e na *Fisherface* com abordagem segmentada foram respectivamente,  $\pm 3,41$  e 5,51%,  $\pm 2,25$  e 3,51%. O algoritmo *Fisherface* original apresentou maior variabilidade na acurácia obtidas para cada *fold* em comparação com a abordagem segmentada.

Analisando as faces dos bancos de face da *ORL* e da *Georgia Tech face*, é possível perceber que todas as faces do banco de face *ORL* estão centralizadas e alinhadas. Diante disso, o algoritmo teve um maior poder de predição em relação à

*Georgia Tech Face*. Para ilustrar essa diferença, na Figura 23 é apresentado um exemplo de faces de uma mesma classe para ambos os bancos de face. Observe-se que as faces do *Georgia Tech Face* estão inclinadas, enquanto as do banco *ORL* estão alinhadas.

Figura 23 - Exemplo de faces dos bancos de Face *Georgia Tech Face* e *ORL*.

(a) Banco de Face *ORL*



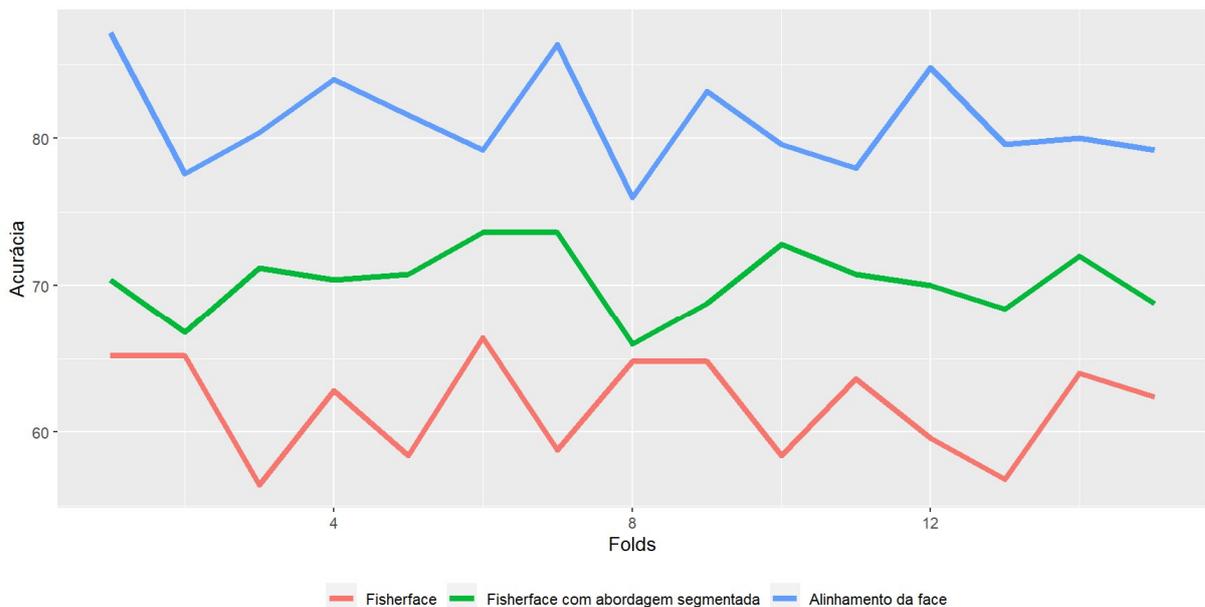
(b) Banco de Face *Georgia Tech Face*



Fonte: Própria.

Com o intuito de obter maiores taxas de acurácia, foi realizado o processo de alinhamento de todas as faces do banco *Georgia Tech Face*. A realização desse processo é simples, uma vez que pode ser feito automaticamente. Com isso, haverá ganhos ao obter maior poder de acurácia para o algoritmo de reconhecimento facial *Fisherface*, quando as faces estiverem desalinhadas.

Figura 24 - Resultado da métrica acurácia nos testes de reconhecimento no banco de face *Georgia Tech Face* com faces alinhadas.



Fonte: Própria.

Na Figura 24, é apresentada a acurácia obtida nos testes de reconhecimento do banco de faces *Georgia Tech Face*, com alinhamento das faces em relação à inclinação dos olhos, para os 15 folds. Com o processo de alinhamento das faces, houve um aumento de 19,28% na média da acurácia em relação à face desalinhada.

Foi realizado um teste de comparação das médias de acurácia obtidas na execução do algoritmo de reconhecimento facial *Fisherface* e da abordagem de alinhamento das faces no banco de faces *Georgia Tech Face*. Conforme apresentado na Tabela 4, com 95% de confiança, rejeita-se a hipótese nula  $H_0: \bar{x}_0 - \bar{x}_1 = 0$ . Com isso, fica evidente que a acurácia média da abordagem de alinhamento das faces é superior à obtida pelo teste de reconhecimento utilizando a face original. Fica claro que o alinhamento das faces em relação aos olhos é um processo fundamental para melhorar o poder de predição do algoritmo de reconhecimento facial *Fisherface*.

Tabela 4 - Resultado do teste Mann-Whitney para comparação da acurácia média entre alinhamento de faces e face original.

Banco de faces	$\bar{x}_0$	$\bar{x}_1$	$M_0$	$M_1$	confiança	valor-p (Mann-Whitney)
<i>Georgia tech Face</i>	61,84	81,12	62,80	80,00	95%	<0.00001

Fonte: Própria.

Legenda:  $\bar{x}_0$  e  $M_0$  representam, respectivamente, a média e a mediana da acurácia do *Fisherface* utilizando a face original.  $\bar{x}_1$  e  $M_1$  representam, respectivamente, a média e a mediana da acurácia utilizando o alinhamento de face antes da execução do algoritmo do *Fisherface*.

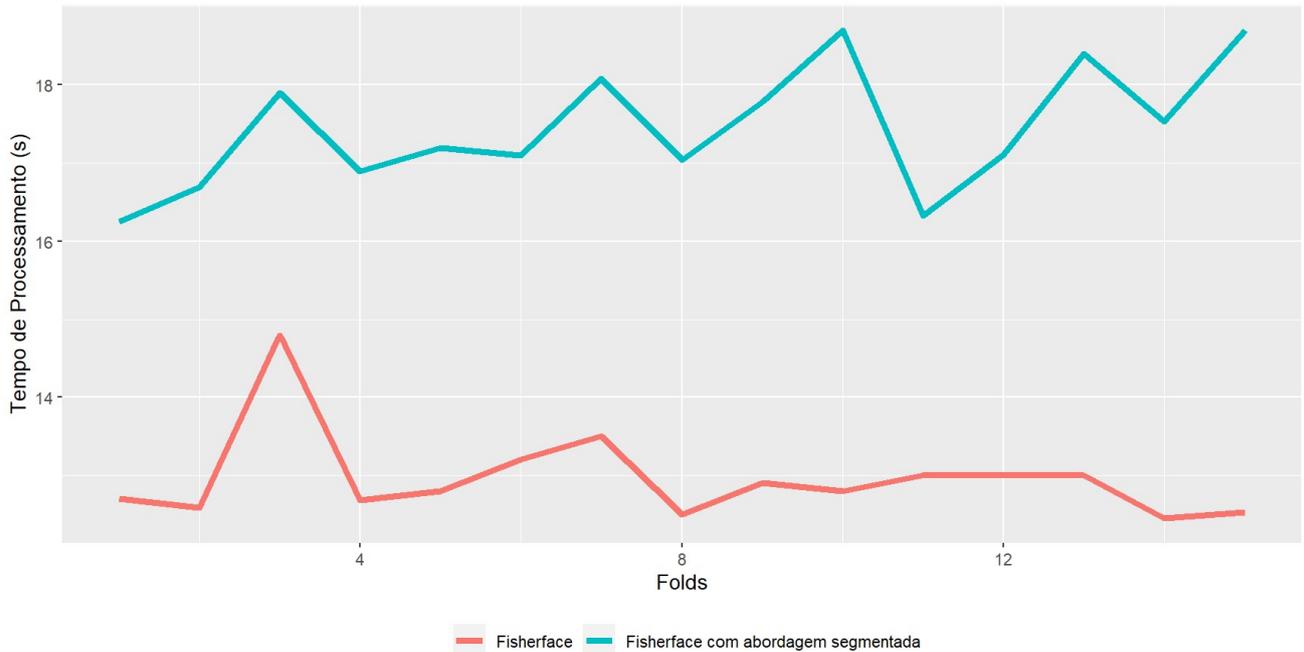
O desvio padrão e o coeficiente de variação encontrados na *Fisherface* utilizando o banco de face *Georgia Tech Face* com ajuste de alinhamento das faces foram  $\pm 3,30$  e 4,08%, respectivamente.

#### 4.2 TEMPO DE PROCESSAMENTO

O tempo médio de processamento do algoritmo de reconhecimento facial *Fisherface* foi de 12,96 segundos, tempo inferior à média do tempo gasto pelo algoritmo de reconhecimento facial *Fisherface* com abordagem segmentada (17.48 segundos).

Na Figura 25, é ilustrado o tempo total gasto nos testes de reconhecimento facial para os algoritmos *Fisherface* e *Fisherface* segmentado, considerando todos os 15 *folds* no banco de faces *ORL*.

Figura 25 - Tempo de processamento do algoritmo de reconhecimento facial *Fisherface* e *Fisherface* com abordagem segmentada no banco *ORL*.



Fonte: Própria.

Foi realizado um teste de comparação entre as médias do tempo de processamento na execução dos algoritmos *Fisherface* e da abordagem segmentada no banco de face *ORL*. Com 95% de confiança, rejeita-se a hipótese nula  $h_0: \bar{x}_0 - \bar{x}_1 = 0$ .

Tabela 5 - Resultado do teste Mann-Whitney para comparação do tempo de processamento.

Banco de faces	$\bar{x}_0$	$\bar{x}_1$	$M_0$	$M_1$	Confiança	valor-p (Mann-Whitney)
ORL	12,96	17,44	12,90	17,20	95%	<0.00001
Georgia tech Face	204,95	136,40	208,60	143,27	95%	<0.00001

Fonte: Própria.

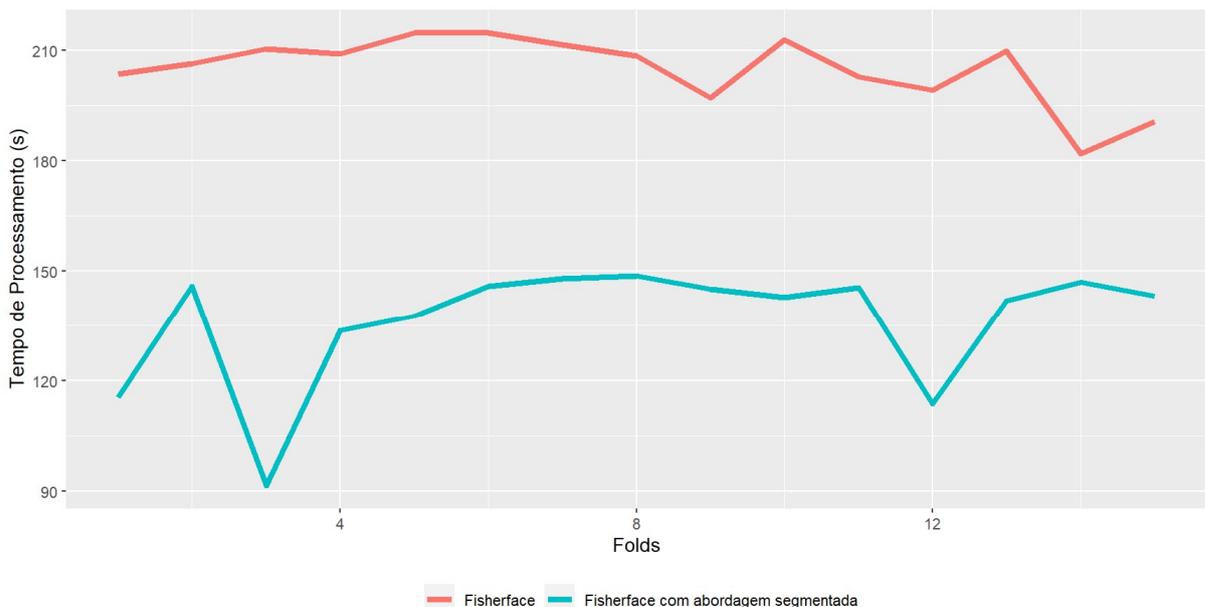
Legenda:  $\bar{x}_0$  e  $M_0$  representam, respectivamente, a média e a mediana do tempo gasto na execução do *Fisherface* com baco de face original.  $\bar{x}_1$  e  $M_1$  representam, respectivamente, a média e a mediana do tempo gasto na execução da abordagem segmentada.

Portanto, ao executar a abordagem segmentada, utilizando a mesma resolução definida na Tabela 2 do banco de dados *ORL* para cada característica, o tempo médio é, em média, 4,48 segundos mais lento do que realizar o reconhecimento facial utilizando a face completa.

O desvio padrão e o coeficiente de variação do tempo de processamento encontrado na *Fisherface* e na *Fisherface* com abordagem segmentada foram, respectivamente,  $\pm 0,58$  e 1,56%,  $\pm 0,79$  e 4,54% para o banco de face *ORL*. A abordagem segmentada apresentou maior variabilidade em comparação com a face completa.

Na Figura 26, é apresentado o tempo de processamento para ambos os algoritmos de reconhecimento facial no banco de face *Georgia Tech Face*. Neste cenário, o algoritmo *Fisherface* com abordagem segmentada obteve menor tempo médio de processamento. O algoritmo da *Fisherface* gastou em média 204,95 segundos, enquanto o algoritmo *Fisherface* com abordagem segmentada gastou em média 136,40 segundos.

Figura 26 - Tempo de processamento do algoritmo de reconhecimento facial *Fisherface* e *Fisherface* com abordagem segmentada no banco *Georgia Tech Face*.



Fonte: Própria.

Foi realizado um teste de comparação do tempo de processamento entre as médias dos algoritmos *Fisherface* e da abordagem segmentada no banco de face *Georgia Tech Face*. Conforme os dados da Tabela 5, com 95% de confiança, rejeita-se a hipótese nula  $H: \bar{x}_0 - \bar{x}_1 = 0$ . Conseqüentemente, a execução da abordagem segmentada requer menos tempo de processamento em comparação ao uso da face inteira. Isso considera que a execução da abordagem segmentada utiliza a resolução definida para cada característica da face, conforme apresentado na Tabela 2.

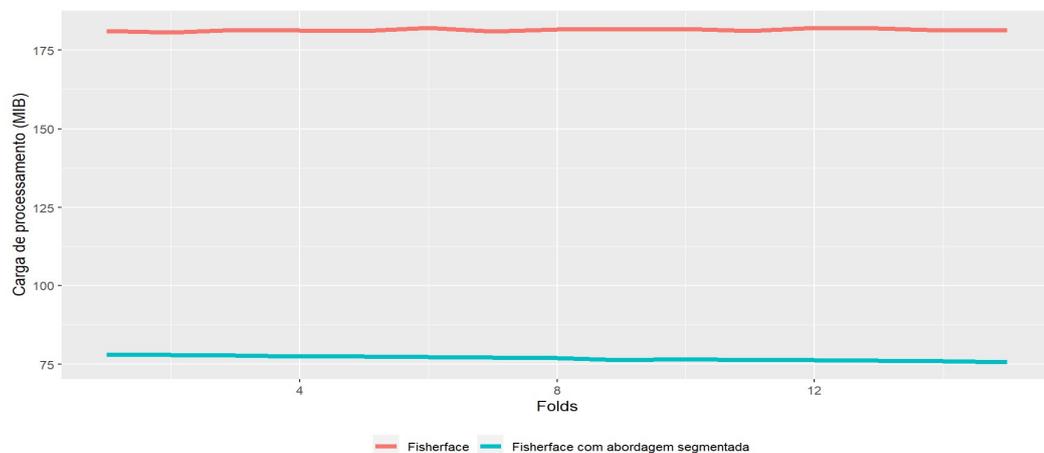
O desvio padrão e o coeficiente de variação do tempo de processamento encontrados na *Fisherface* e na *Fisherface* com abordagem segmentada foram, respectivamente,  $\pm 9,35$  e  $4,56\%$ ,  $\pm 16,52$  e  $12,11\%$ , no banco de faces da *Georgia Tech Face*. A abordagem segmentada apresentou maior variabilidade em comparação com a abordagem da face completa.

#### 4.3 CARGA DE PROCESSAMENTO

Por último, foi avaliado a métrica carga de processamento, que representa a quantidade de memória utilizada pelo computador para realizar a execução completa dos testes de reconhecimento facial. O pacote *peakRAM* apresenta a quantidade de memória utilizada em *MebiByte* (MIB), onde:

$$1 \text{ MIB} = 1048576 \text{ Bytes.} \quad (4.1)$$

Figura 27 - Carga de processamento do algoritmo de reconhecimento facial *Fisherface* e *Fisherface* com abordagem segmentada no banco *ORL*.



Fonte: Própria.

Na Figura 27 é apresentada a quantidade de memória utilizada pelo computador para execução dos dois algoritmos de reconhecimento facial, *Fisherface* e *Fisherface* com abordagem segmentada, no banco de face *ORL*. Foram executados todos os 15 *folds* para ambos os algoritmos.

Apesar de o algoritmo *Fisherface* ter sido executado em menor tempo, o algoritmo de reconhecimento facial *Fisherface* com abordagem segmentada teve uma média menor de carga de processamento (76,95 MIB) no processo de reconhecimento facial. O algoritmo *Fisherface* utilizou (181,39 MIB), o que corresponde ao dobro de memória utilizado para a abordagem segmentada. Consequentemente, para o banco de faces *ORL*, o método *Fisherface* requer um computador com maior capacidade de memória em comparação com a abordagem segmentada para a execução do reconhecimento facial.

Tabela 6 - Resultado do teste Mann-Whitney para comparação da carga de processamento.

Banco de face	$\bar{x}_0$	$\bar{x}_1$	$M_0$	$M_1$	Confiança	valor-p (Mann-Whitney)
<i>ORL</i>	181,39	76,95	181,30	77,00	95%	<0.0001
<i>Georgia Tech Face</i>	1643,91	241,44	1651,80	244,10	95%	<0.0001

Fonte: Própria.

Legenda:  $\bar{x}_0$  e  $M_0$  representam, respectivamente, a média e a mediana para carga de processamento gasta na execução do algoritmo *Fisherface* com banco de face original.  $\bar{x}_1$  e  $M_1$  representam, respectivamente, a média e a mediana da carga de processamento utilizada na execução do algoritmo *Fisherface* com a abordagem segmentada.

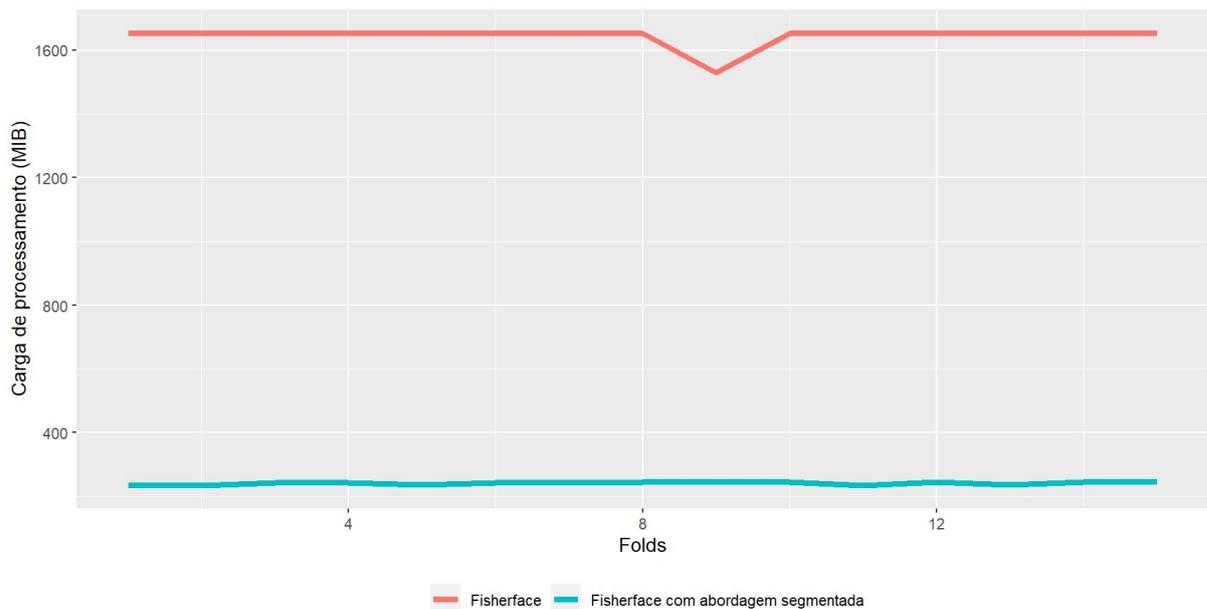
Foi realizado um teste entre a carga média de processamento na execução dos algoritmos de reconhecimento facial *Fisherface* e da abordagem segmentada no banco de faces *ORL*. Conforme os dados da Tabela 6, com 95% de confiança, rejeita-se a hipótese nula  $H_0: \bar{x}_0 - \bar{x}_1 = 0$ . Consequentemente, a abordagem segmentada utiliza menos memória para a execução do algoritmo em comparação à utilização da face completa.

O desvio padrão e o coeficiente de variação da métrica carga de processamento obtidos na execução dos algoritmos de reconhecimento facial *Fisherface* e *Fisherface* com abordagem segmentada, no banco de face *ORL*, foram,

respectivamente,  $\pm 0,43$  e  $0,024\%$ ,  $\pm 0,77$  e  $1,00\%$ . A abordagem segmentada apresentou maior variabilidade na execução de cada *fold*s para carga de processamento em comparação com a face original.

No banco de faces da *Georgia Tech face*, houve também uma redução na carga de processamento com a utilização da abordagem segmentada. Conforme ilustrado na Figura 28, o algoritmo de reconhecimento facial *Fisherface* teve um consumo médio de memória de 1643,91 MIB, o que corresponde a mais de sete vezes o consumo de memória se fosse utilizada a abordagem segmentada, que é de 241,44 MIB.

Figura 28 - Carga de processamento do algoritmo de reconhecimento facial *Fisherface* e *Fisherface* com abordagem segmentada no banco *Georgia Tech Face*.

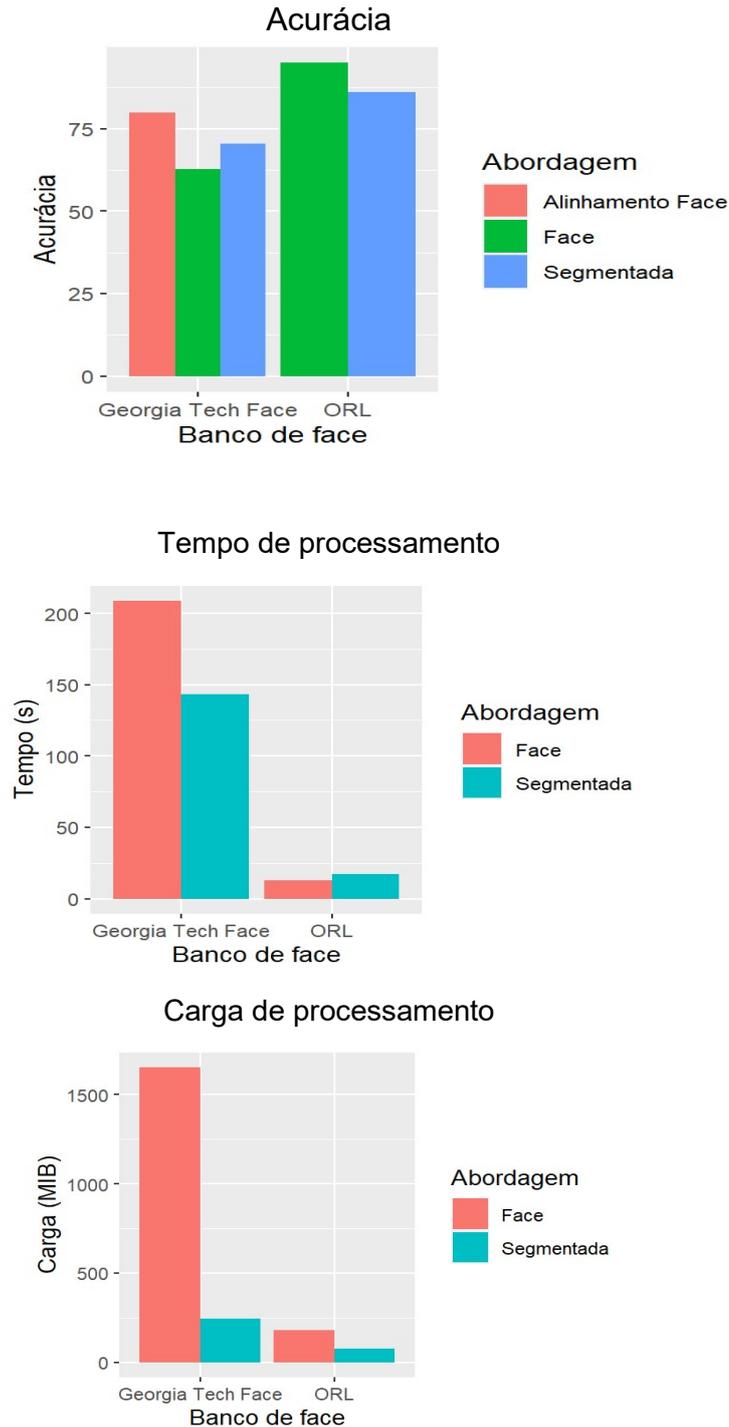


Fonte: Própria.

Foi realizado um teste de carga de processamento na execução dos algoritmos de reconhecimento facial *Fisherface* e da abordagem Segmentada para o banco de face *Georgia Tech Face*. Conforme os dados da tabela 6, com 95% de confiança, rejeita-se a hipótese nula  $H_0: \bar{x}_0 - \bar{x}_1 = 0$ . Com isso, a abordagem segmentada mostrou-se mais leve, pois utiliza matrizes de menor dimensão para realizar os cálculos, resultando em menor utilização de memória em comparação com a utilização da face completa.

O desvio padrão e o coeficiente de variação da métrica carga de processamento obtido na execução dos algoritmos de reconhecimento facial *Fisherface* e *Fisherface* com abordagem segmentada no banco de faces *Georgia Tech Face* foram, respectivamente,  $\pm 0,32$  e  $1,92\%$ ,  $\pm 4,64$  e  $1,92\%$ .

Figura 29 - Comparação das métricas avaliadas.



Fonte: Própria.

A Figura 29 apresenta um resumo comparativo entre as três abordagens testadas. Apesar de a abordagem segmentada exigir menos memória para processamento, ela se mostrou menos eficaz em relação à métrica de acurácia quando comparada ao algoritmo de reconhecimento facial *Fisherface* no banco de faces *ORL*.

A abordagem segmentada é um processo trabalhoso e dificilmente terá uma implementação confiável automatizada. Nos testes realizados com a abordagem segmentada, em que o banco de faces já esteja normalizado, ou seja, todas as faces estejam alinhadas e o tamanho das faces reduzido, demonstraram que essa abordagem piorava o desempenho desse indicador. Embora tenha alcançado uma taxa de acurácia maior no banco *Georgia Tech Face*, por não estar normalizado, em comparação ao *Fisherface*, a abordagem de alinhamento das faces demonstrou ser mais precisa nas predições do que a abordagem segmentada para o banco *Georgia Tech Face*.

A abordagem de alinhamento de face é um processo simples que pode ser automatizado. Conforme os resultados dos testes de reconhecimento com o alinhamento de face, demonstraram um aumento de quase 20% no poder de predição em relação à imagem original. Por isso, é recomendável sempre utilizar o alinhamento de face antes de executar o algoritmo de reconhecimento facial *Fisherface*. No estudo de Li *et al.* (2010), que utilizou o processo de alinhamento da face no reconhecimento facial, foram obtidas melhores taxas de acerto, proporcionalmente de 2 a 10 vezes maiores.

A abordagem de segmentação utiliza características fragmentadas da face, o que torna o processamento dessas características mais rápido. No entanto, é importante observar que, dependendo do tamanho escolhido para cada fragmento (característica), o tempo total acumulado pode ser maior do que o tempo de processamento da face completa. Esse fato foi observado no banco de faces *ORL*, onde o tempo de processamento da abordagem segmentada foi superior ao da abordagem da face completa.

O único cenário em que realmente seja vantajoso utilizar a abordagem segmentada é a carga de processamento. Apesar de executar o algoritmo de

reconhecimento facial várias vezes, cada execução é de uma matriz de dados muito menor do que a face completa. Com isso, o computador necessita de menor espaço para armazenar as operações do cálculo das projeções das faces e, assim, obter as previsões para cada característica.

## 5 CONCLUSÕES

Ao avaliar a carga de processamento da abordagem segmentada, fica claro que ela reduz a quantidade de memória necessária para a execução do algoritmo de reconhecimento facial, mesmo em cenários em que o tempo de processamento seja maior.

Ao analisar a métrica de tempo, foi possível observar que, quanto maior o tamanho da face completa, melhor é o tempo de processamento da abordagem segmentada. No entanto, em casos em que a imagem da face possui baixa resolução, como no banco *ORL*, em que a imagem tem dimensões de  $92 \times 112$ , o tempo gasto para a face completa foi menor do que o da abordagem segmentada.

A acurácia obtida pela abordagem segmentada foi inferior à da abordagem tradicional do algoritmo *Fisherface* no banco de faces *ORL*. A acurácia é uma métrica fundamental para avaliar a qualidade de um algoritmo de reconhecimento facial, e, portanto, a abordagem segmentada não apresentou um desempenho superior em relação a abordagem da face original. Por outro lado, a estratégia de alinhamento da face melhorou consideravelmente a acurácia do algoritmo *Fisherface* no banco de faces *Georgia Tech Face*, superando a abordagem segmentada.

## REFERÊNCIAS

ANTON, Howard. **Álgebra linear com aplicações**. 8. ed. Porto Alegre: Bookman, 2001.

ASHBOURN, J. **Biometrics in the new world: the cloud, mobile technology and pervasive identity**. Switzerland: Springer International, 2014. v. 1.

BAEZA ARGÜELLO, Saúl; WAKKARY, Ron; ANDERSEN, Kristina; TOMICO, Oscar. Exploring the potential of apple face ID as a drag, queer and trans technology design tool. *In: DESIGNING INTERACTIVE SYSTEMS CONFERENCE*, 1., 2021. Virtual Event USA. **Anais** [...], Virtual Event USA, 2021. p. 1654–1667. Disponível em: <https://dl.acm.org/doi/10.1145/3461778.3461999>. Acesso em: 15 ago. 2021.

BAQEEL, Hind; SAEED, Saqib. Face detection authentication on Smartphones: End Users Usability Assessment Experiences. *In: INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCES (ICCIS)*, 2019. **ANAIS** [...]. Sakaka, Saudi Arabia: IEEE, abr. 2019. p. 1–6. Disponível em: <https://ieeexplore.ieee.org/document/8716452/>. Acesso em: 11 ago. 2023.

BELHUMEUR, Peter N; HESPANHA, Joao P; KRIEGMAN, David J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. **IEEE Transactions on pattern analysis and machine intelligence**, [s. l.], v. 19, n. 7, p. 10, 1997.

BERTILLON, Alphonse. Identification anthropométrique: Instructions signalétiques. **Imprimerie Administrative**, Melun, 1893.

BHUIYAN, Md Al Amin; AMPORNARAMVETH, Vuthichai; MUTO, Shin Yo; UENO, Haruki. Face detection and facial feature localization for human-machine interface. **NII Journal**, [s. l.], p. 25–39, 2003.

BISSI, Theury David. **Reconhecimento facial com os algoritmos**. 2018. 41 f. Trabalho de Conclusão de Curso (Graduação em Bacharel em Ciências das Computação) - Universidade Federal de Uberlândia, Uberlândia, 2018.

BLEDSON, WW. The model method in facial recognition. **Technical Report**, Panoramic Research, [s. l.], v. 15, 1964.

BORADE, Sushma Niket; DESHMUKH, Ratnadeep R.; SHRISHRIMAL, Pukhraj. Effect of distance measures on the performance of face recognition using principal component analysis. *In: BERRETTI, Stefano; THAMPI, Sabu M.; SRIVASTAVA, Praveen Ranjan (Org.). Intelligent systems technologies and applications*. Advances in intelligent systems and computing. Cham: Springer International, 2016. v. 384, p. 569–577. Disponível em: [http://link.springer.com/10.1007/978-3-319-23036-8\\_50](http://link.springer.com/10.1007/978-3-319-23036-8_50). Acesso em: 24 maio 2022.

BRAGA, Luiz Filipe Zenicola. **Sistemas de reconhecimento facial**. 2013. 83 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica com Ênfase em Eletrônica) – Universidade de São Paulo, São Carlos, SP, 2013.

BURMAN, Prabir. A comparative study of ordinary cross-validation,  $v$ -fold cross-validation and the repeated learning-testing methods. **Biometrika**, [s. l.], v. 76, n. 3, p. 503–514, 1989. <https://doi.org/10.1093/biomet/76.3.503>.

CARLOS JUNIOR, Luis Fernando Martins. **Reconhecimento de faces utilizando um modelo conexcionista baseado em populações de neurônios**. 2015. 59 f. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) – Universidade de São Paulo, São Carlos, SP, 2015. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-04082015-104017/>. Acesso em: 16 ago. 2021.

CHEN, Lien-Wu; HO, Yu-Fan; TSAI, Ming-Fong. Instant social networking with startup time minimization based on mobile cloud computing. **Sustainability**, [s. l.], v. 10, n. 4, p. 1195, 2018.

COSTA, Lucas José da *et al.* Análise de métodos de detecção e reconhecimento de faces utilizando visão computacional e algoritmos de aprendizado de máquina. **Colloquium Exactarum**, [s. l.], v. 13, n. 2, p. 01–11, 2021.

CROWLEY, James L.; CHRISTENSEN, Henrik I. **Vision as process: basic research on Computer Vision**. 1. ed. [S. l.]: Springer, 1995.

CUNHA, André Luiz Barbosa Nunes da. **Sistema automático para obtenção de parâmetros do tráfego veicular a partir de imagens de vídeo usando OpenCV**. 2013. 119 f. Tese (Doutorado em Planejamento e Operação de Sistemas de Transporte) – Universidade de São Paulo, São Carlos, 2013. Disponível em: <http://www.teses.usp.br/teses/disponiveis/18/18144/tde-19112013-165611/>. Acesso em: 16 ago. 2021.

DA SILVA, JOSÉ IVSON SOARES. **Reconhecimento facial em imagens, de baixa resolução**. 2015. 73 f. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal de Pernambuco. Recife. 2015.

DE SOUZA, Flavio Lima. **Classificador fisherface fuzzy para o reconhecimento de faces**. 2014. 106 f. Dissertação (Mestrado em Matemática) - Universidade Estadual Paulista. São José do Rio Preto. 2014.

DUAN, Hong; YAN, Ruohe; LIN, Kunhui. Research on face recognition based on PCA. *in*: INTERNATIONAL SEMINAR ON FUTURE INFORMATION TECHNOLOGY AND MANAGEMENT ENGINEERING, 20., 2008, Leicestershire, **Anais [...]**, Leicestershire: IEEE, 2008. p. 29-32.

FERREIRA, Dr Danton Diego. **Método de reconhecimento de face baseado em estatísticas de ordem superior**. 2016. 73 f. Dissertação (Mestrado em Engenharia de Sistemas e Automação) - Universidade Federal de Lavras, Lavras, MG, 2016.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of Eugenics**, [s. l.], v. 7, n. 2, p. 179–188, sept. 1936. <https://doi.org/10.1111/j.1469-1809.1936.tb02137.x>.

GEORGIA TECH FACE DATABASE. [S. l.: s. n.], [2000]. Disponível em: [http://www.anefian.com/research/face\\_reco.htm](http://www.anefian.com/research/face_reco.htm). Acesso em: 12 maio 2022.

GIRON-SIERRA, Jose Maria. **Digital signal processing with Matlab examples**. Singapore: Springer, 2017. v 2, 944 p.

GUPTA, Sheifali *et al.* A new optimized approach to face recognition using EigenFaces. **Global Journal of Computer Science and Technology**, [s. l.], v. 10, p. 15–17, 2010.

GEIGEN. **Calculate generalized eigenvalues, the generalized schur decomposition and the generalized singular value decomposition of a matrix pair with Lapack**. [S. l.: s. n.], [2009]. Disponível em: <https://CRAN.R-project.org/package=geigen>. Acesso em: 20 jan 2022.

HONGYU, Kuang; SANDANIELO, Vera Lúcia Martins; OLIVEIRA JUNIOR, Gilmar Jorge de. Análise de componentes principais: resumo teórico, aplicação e interpretação. **E&S Engineering and Science**, [s. l.], v. 5, n. 1, p. 83–90, Jun 2016. <https://doi.org/10.18607/ES201653398>.

JOHNSON, Richard A.; WICHERN, Dean W. **Applied multivariate statistical analysis**. 6th ed. Upper Saddle River, N.J: Pearson Prentice Hall, 2007.

KANADE, Takeo. **Computer recognition of human faces**. Basel: Birkhäuser Basel, 1977. 97 f.

KITANI, Edson C.; THOMAZ, Carlos E. Análise de discriminantes lineares para modelagem e reconstrução de imagens de face. *In*: XXVII CONGRESSO DA SBC, ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL, VI., 2007, Rio de Janeiro. **Anais [...]**. Rio de Janeiro: SBC, 2007. p. 962-971.

LOPES, Eduardo Costa. **Deteção de faces e características faciais**. Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul, 2005. Relatório Técnico N. 45.

MACHADO, André Wilson; SOUKI, Bernardo Quiroga. Simplificando a obtenção e a utilização de imagens digitais: scanners e câmeras digitais. **Revista Dental Press de Ortodontia e Ortopedia Facial**, [s. l.], v. 9, n. 4, p. 133–156, ago. 2004. <https://doi.org/10.1590/S1415-54192004000400012>.

MANN, H. B.; WHITNEY, D. R. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. **The Annals of Mathematical Statistics**, v. 18, n. 1, p. 50–60, mar. 1947. <https://doi.org/10.1214/aoms/1177730491>.

MARQUÉS, Ion. **Face recognition algorithms**. 2010. 78 f. Dissertação (Mestrado em Ciências da Computação) – Universidade do País Basco, Lejona, 2010.

MEDEIROS, L X; CARRIJO, G A; FLÔRES, E L; VEIGA, A C P. Genetic Algorithms Applied in Face Recognition. **IEEE Latin America Transactions**, [s. l.], v. 10, n. 6, p. 6, 2012.

MENDES, Kreisler Brenner. **Comparativo de algoritmos clássicos de aprendizado de máquina em um problema de reconhecimento de faces**. 2019. 71 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Eletrônica e de Telecomunicações) – Universidade Federal de Uberlândia, Patos de minas, 2019.

MINGOTI, Sueli Aparecida. **Análise de dados através de métodos de estatística multivariada: uma abordagem aplicada**. Belo Horizonte: UFMG, 2005.

OPENIMAGER: **An image processing toolkit**. [S. l.: s. n.], [2021]. Disponível em: <https://CRAN.R-project.org/package=OpenImageR>. Acesso em: 20 jan 2022.

OJALA, Timo; PIETIKÄINEN, Matti; HARWOOD, David. A Comparative study of texture measures with classification based on featured distribution. **Pattern Recognition**. [s. l.], v. 29, p. 51–59, 1996.

PERLIBAKAS, Vytautas. Distance measures for PCA-based face recognition. **Pattern Recognition**, [s. l.], v. 25, n. 6, p. 711–724, abr. 2004.

PRADO, Kelvin Salton do. **Comparação de técnicas de reconhecimento facial para identificação de presença em um ambiente real e semicontrolado**. 2018. Dissertação (Mestrado em Sistemas de Informação) – Universidade de São Paulo, São Paulo, 2018. Disponível em: <http://www.teses.usp.br/teses/disponiveis/100/100131/tde-07012018-222531/>. Acesso em: 13 ago. 2021.

PEAKRAM: **Tracking Peak RAM use and minimizing overhead**. [S. l.: s. n.], [2017], Disponível em: <https://cran.rproject.org/web/packages/peakRAM/index.html>. Acesso em: 20 jan 2022.

R DEVELOPMENT CORE TEAM. **R: A Language and Environment for Statistical Computing**. Vienna: R Foundation for Statistical Computing, [2009]. Disponível em: <http://www.R-project.org>.

ROSA, Simone Marques. **A invisibilidade da papiloscopia na persecução penal em Goiás**. 2016. 112 p. Dissertação (Mestrado em História) - Pontifícia Universidade Católica de Goiás, Goiás, 2016.

RSTUDIO TEAM. **RStudio: integrated development for R**. Boston, MA: Posit PBC, 2023. Disponível em: <http://www.rstudio.com/>. Acesso em: 20 Jan 2022.

SAMARIA, F.S.; HARTER, A.C. Parameterisation of a stochastic model for human face identification. *In: IEEE WORKSHOP ON APPLICATIONS OF COMPUTER VISION*, 1994. **Proceedings** [...]. Sarasota, FL: IEEE Comput. Soc. Press, 1994. p. 138–142. DOI 10.1109/ACV.1994.341300. Disponível em: <http://ieeexplore.ieee.org/document/341300/>. Acesso em: 27 set. 2021.

SCORSATO, Helen. O uso da fotografia em processos de identificação e o método Bertillon – século XIX. **Estudios Historicos: CDHRPyB**, Uruguai, Ano 4, N.9, Dez 2012 - N° 9. dez. 2012. Disponível em: <https://estudioshistoricos.org/edicion9/eh0911.pdf>. Acesso em: 27 set 2022.

SILVA, Alex Lima. **Redução de características para classificação de imagens de faces**. 2016. Dissertação (Mestrado em Ciências da Computação) - Universidade Federal Rural do Semi-Árido, Mossoró, 2016.

SILVA, Alexandre Fieno Da. **Reconhecimento de faces via PCA: análise de desempenho**. 2006. Dissertação (Mestrado em Ciências da Computação) – Universidade de Uberlândia, Uberlândia, 2006.

SIROVICH, L.; KIRBY, M. Low-dimensional procedure for the characterization of human faces. **Journal of the Optical Society of America A**, [s. l.], v. 4, n. 3, p. 519, March. 1987. <https://doi.org/10.1364/JOSAA.4.000519>.

SPANHOL, Fernando José; MENDONÇA LUNARDI, Giovani; VIEIRA DE SOUZA, Márcio. **Tecnologias da informação e comunicação na segurança pública e direitos humanos**. São Paulo: Editora Edgard Blücher, 2016. Disponível em: <https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5809618>. Acesso em: 10 ago. 2021.

ORL: **Database of faces**. [S. l.: s. n.], [1994]. Disponível em: <https://cam-orkl.co.uk/facedatabase.html>. Acesso em: 1 ago. 2022.

STUDENT. The probable error of a mean. **Biometrika**, [s. l.], v. 6, n. 1, p. 1, march. 1908.

TURK, Matthew; PENTLAND, Alex. Eigenfaces for recognition. **Journal of Cognitive Neuroscience**, [s. l.], v. 3, n. 1, p. 71–86, Jan. 1991.

VIOLA, PAUL; JONES, MICHAEL. Rapid object detection using a boosted cascade of simple features. *In: IEEE COMPUTER SOCIETY CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION*. 1., 2001, Kauai. **Proceedings [...]**. Kauai: IEEE, 2001. v. 1, p. I–I. <https://doi.org/10.1109/CVPR.2001.990517>.

WAHYU MULYONO, Ibnu Utomo ,*et al.* Performance analysis of face recognition using Eigenface approach. *In: INTERNATIONAL SEMINAR ON APPLICATION FOR TECHNOLOGY OF INFORMATION AND COMMUNICATION (ISEMANTIC)*. **Anais [...]**. Semarang: IEEE, 2019. p. 1–5. Disponível em: <https://ieeexplore.ieee.org/document/8884225/>. Acesso em: 16 nov. 2022.

YANG, Guangzheng; HUANG, Thomas S. Human face detection in complex background. **Pattern Recognition**, [s. l.], v. 27, n. 1, p. 53–63, 1994.

YANG, Ming-Hsuan; KRIEGMAN, D. J; AHUJA, N. Detecting faces in images: a survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [s. l.], v. 24, p. 34–58, 2002. .

## APÊNDICE A – Implementação linguagem R

```

library('geigen')
library('OpenImageR')
library('peakRAM')

#Recebe uma lista de todas as bases de face, cada uma
#contendo um vetor com o caminhos das faces de treinamento e
#de testes
fisherface <- function (dados) {
  predicao <- function (bteste, btreinamento) {
    tlabel <- list()

    tamostrat <- list()

    linha <- 0

    coluna <- 0

    classes <-
      list.dirs(btreinamento,
                full.names = T,
                recursive = F)

    for (cl in classes) {
      #Procurar pela face completa
      #\\d+\\w*\\.jpg$

      x <- list.files(
        cl,
        recursive = T,
        include.dirs = F,
        ignore.case = T,
        pattern = '\\d+\\w*\\.jpg'
      )

      px <- list()

      for (xi in x) {
        fl <- paste(cl, '/', xi, sep = '')

        im <- readImage(fl)

        if (length(dim(im)) > 2) {
          im <- rgb_2gray(im)
        }
      }
    }
  }
}

```

```

    }
    posNome <- regexpr('/\\w+$', cl, TRUE)
    nomear <- substr(cl, posNome + 1, nchar(cl))
    tlabel[[length(tlabel) + 1]] <- paste(nomear, xi)

    px[[xi]] <- c(im)

    if (linha == 0 & coluna == 0) {
      linha <- NCOL(im)

      coluna <- NROW(im)

    }
  }
  tamostral[[cl]] <- px
}

x <-
list.files(
  bteste,
  recursive = T,
  include.dirs = F,
  ignore.case = T,
  pattern = '\\d+\\w*\\.jpg'
)

qteste <- list()

for (r in x) {
  im <- readImage(paste(bteste, '/', r, sep = ''))

  if (length(dim(im)) > 2) {
    im <- rgb_2gray(im)

  }

  qteste[[r]] <- c(im)
}

totalponto <- linha * coluna

qfotosPessoa <- length(tamostral[[1]])

```

```

nomr_mat <- function(m) {
  return(m / norm(m, '2'))
}

contadores <- function (tamostral) {
  totalN <- 0

  soma <- c()

  for (t in tamostal) {
    for (s in t) {
      if (totalN == 1) {
        soma <- c(1:length(s) * 0)
      }
      soma <- soma + s

      totalN <- totalN + 1
    }
  }
  dados = list(totald = totalN, mediaGeral = soma /
totalN)

  return (dados)
}
cont <- contadores(tamostral)

engenface <- function (tamostral, cont) {
  tamostalmedio <- list()

  tamostaldesv <- list()

  tamostalCov <- list()

  tmatrixGeral <- list()

  ct <- 1

  for (i in tamostal) {
    for (t in 1:length(i)) {
      item <- i[[t]]

      dif <- item - cont[['mediaGeral']]

```

```

    dif2 <- dif / norm(cbind(dif), '2')
    tmatrixGeral[[ct]] <- matrix(dif2, ncol = 1, byrow =
F)
    ct <- ct + 1

  }

}
res = matrix(unlist(tmatrixGeral),
            ncol = cont[['totald']],
            byrow = F)

eigen <- eigen(t(res) %*% res)

transf <- list()

#total de face menos numero de classes
dif <- cont[['totald']] - length(tamostral)

cont <- 1

for (c in order(eigen[['values']], decreasing = T)) {
  transf[[cont]] <- eigen[['vectors']][, c]

  cont <- cont + 1
}

eigenfaces <-
  matrix(unlist(transf),
        ncol = length(transf),
        byrow = F)

eigenfaces <- nomr_mat(res %*% eigenfaces)

res <-
  list(covdif = res,
       autov = transf,
       eigenfac = eigenfaces[, 1:dif])

return (res)

}
eigenfaces <- engenface(tamostral, cont)

reconstruir <- function(eigenfaces) {

```

```

#total de face menos numero de classes
dif <- cont[['totald']] - length(tamostral)

fs = matrix(rep(0, dif * cont[['totald']]), ncol =
cont[['totald']])
for (ct in 1:cont[['totald']]) {
  colun <-
    t(eigefaces[['eigenfac']]) %*%
matrix(eigefaces[['covdif']][, ct], ncol =
1)

  fs[, ct] <- colun
}

return (fs)
}
fs <- reconstruir(eigefaces)

#calcular A LDA
lda <- function(fs, tamostal, eigenf, qfotosPessoa) {
  dif <- cont[['totald']] - length(tamostral)

  mediaTotal <- (rowMeans(fs))

  mediaPorClasse <-
    matrix(rep(0, dif * length(tamostral)), ncol =
length(tamostral))

  sw <- matrix(rep(0, dif * dif), ncol = dif)

  sb <- matrix(rep(0, dif * dif), ncol = dif)

  for (c in 1:length(tamostral)) {
    ix <- ((c - 1) * qfotosPessoa) + 1):(c *
qfotosPessoa)

    mediaPorClasse[, c] <- (rowMeans(fs[, ix]))

    #calcula da covariancia da intra classe
    for (d in ix) {
      aux <- matrix(fs[, d] - mediaPorClasse[, c], ncol =
1)

      sw <- sw + (aux %*% t(aux))
    }
}

```

```

#calculo da inter-classe
aux <- matrix(mediaPorClasse[, c] - mediaTotal, ncol =
1)

sb <- sb + (aux %*% t(aux))

}
#t <- solve(sw)%*%sb;
autovetor <- geigen(sb, sw, symmetric = F)

#ordenar os autovetores
autov <-
  autovetor[['vectors']][, order(autovetor[['values']],
decreasing = T)]

autov <- autov[, 1:length(tamostral) - 1]

fisherface <- eigenf[['eigenfac']] %*% autov

res = list(sw = sw,
          sb = sb,
          fisherface = fisherface)

return (res)
}

matrisesLda <- lda(fs, tamostral, eigefaces, qfotosPessoa)

fisherfs <- function(wf,
                    tdif,
                    tamostral,
                    totalponto,
                    totalamostra) {
  pf <-
    matrix(rep(0, (length(tamostral) - 1) * totalamostra),
ncol = totalamostra)

  for (c in 1:totalamostra) {
    tdifc <- matrix(tdif[, c], ncol = 1)
    pf[, c] <- t(wf) %*% tdifc
  }
}

```

```

#reconstruir
rf <-
  matrix(rep(0, totalponto * totalamostra), ncol =
totalamostra)
  for (c in 1:totalamostra) {
    rf[, c] <- nomr_mat(wf %*% pf[, c])
  }
result <- list(pf = pf, rf = rf)
return(result)
}

plotarGrafico <- function(ffs, totalamostra) {
  comp1 <- list()

  comp2 <- list()

  for (c in 1:totalamostra) {
    aux <- abs(ffs[, c])

    aux1 <- order(aux, decreasing = T)

    aux3 <- ffs[aux1, c]

    comp1[c] <- aux3[1]
    comp2[c] <- aux3[2]
  }

  plot(
    data.frame(c1 = unlist(comp1), c2 = unlist(comp2)),
    col = c('blue', 'blue', 'red', 'red', 'green',
'green')
  )
}

ffs <-
  fisherfs(matrisesLda[['fisherface']],
eigefaces[['covdif']], tamostrat, totalponto,
cont[["totald"]])

#plotarGrafico(ffs[['pf']],cont[["totald"]]);
realizarteste <- function(testes, med, wf, ffs) {
  tvlTeste = list()

```

```

for (ind in names(testes)) {
  teste <- testes[[ind]]

  vdif <- teste - med

  to <- vdif / norm(vdif, '2')

  tfs <- t(wf) %*% to

  tvl = list()

  for (r in 1:NCOL(ffs[['pf']])) {
    aux <- norm(unlist(tfs - ffs[['pf']][, r]), '2')

    label <- tlabel[[r]]

    tvl[[label]] <- aux
  }
  tvlTeste[[ind]] <- tvl
}

return(tvlTeste)
}

dist <- realizarteste(qtteste, cont[['mediaGeral']],
                    matrisesLda[['fisherface']],
                    ffs['pf'])

showface <- function(ff, posPes) {
  imgsh = matrix(ff[, posPes], nrow = coluna)

  imageShow(imgsh)
}
#showface(ffs[['rf']],3);
return (dist)
}
listPred <- list()

for (caract in names(dados)) {
  base <- unlist(dados[caract])

  listPred[[caract]] <- predicao(base[[1]], base[[2]])
}

```

```

}

tv1Medio <- list()

erros <- list()

for (nomeImgTeste in names(listPred[[1]])) {
  min <- 100

  ImgTreinaMenor <- 0

  tv1Teste <- list()

  for (imgTreina in names(listPred[[1]][[nomeImgTeste]])) {
    vl <- 0

    for (carect in names(listPred)) {
      vl <- vl +
listPred[[carect]][[nomeImgTeste]][[imgTreina]]
    }
    vlMed <- vl / length(listPred)
    if(!length(vlMed)){
      print(nomeImgTeste);
      print(imgTreina)
    }

    if (vlMed < min) {
      min <- vlMed
      ImgTreinaMenor <- imgTreina
    }
    tv1Teste[[imgTreina]] <- vlMed

  }
  classNomeImgTeste <- unlist(strsplit(nomeImgTeste, '/'))
  classnomeImgtreina <- unlist(strsplit(ImgTreinaMenor, '
'))

  if (classNomeImgTeste[1] != classnomeImgtreina[1]) {
    comparacaoErro <- list();
    for (carect in names(listPred)) {
      for(imgTreina in
names(listPred[[1]][[nomeImgTeste]])){
        classImgTreina <- unlist(strsplit(imgTreina, '
'))[1];
        if(classImgTreina == classNomeImgTeste[1]){

```

```

        dados <-
append(comparacaoErro[[imgTreina]],listPred[[carect]][[nomeImg
Teste]][[imgTreina]])
        comparacaoErro[[imgTreina]] <- dados
    }

    if(imgTreina == ImgTreinaMenor){
        dados <-
append(comparacaoErro[[imgTreina]],listPred[[carect]][[nomeImg
Teste]][[imgTreina]])
        comparacaoErro[[imgTreina]] = dados;
    }
}

erros[[nomeImgTeste]] <-
list(
    imgteste = nomeImgTeste,
    imgIdent = ImgTreinaMenor,
    qtest = comparacaoErro
)

}

    tvlMedio[[nomeImgTeste]] <- list(dados = tvlTeste)
}
res <- list(dados=tvlMedio,erros=erros)

return (res)
}

georgia_seg0 <-
list(
    FACE = c(
        'PASTA/Testar',
        'PASTA/TREINAR'
    )
)

men <- peakRAM({
    result <- fisherface(georgia_seg0)
})

```