

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Thelry David Bissi

**Reconhecimento Facial com os algoritmos**  
***Eigenfaces* e *Fisherfaces***

**Uberlândia, Brasil**

**2018**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Thelry David Bissi

**Reconhecimento Facial com os algoritmos *Eigenfaces* e  
*Fisherfaces***

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, Minas Gerais, como  
requisito exigido parcial à obtenção do grau  
de Bacharel em Ciência da Computação.

Orientador: Mauricio Cunha Escarpinati

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2018

Thelry David Bissi

## **Reconhecimento Facial com os algoritmos *Eigenfaces* e *Fisherfaces***

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 14 de Julho de 2018:

---

**Mauricio Cunha Escarpinati**  
Mauricio Cunha Escarpinati

---

**Murillo Guimarães Carneiro**

---

**Jocival Dantas Dias Junior**

Uberlândia, Brasil  
2018

*Dedico principalmente a meu querido pai e falecida mãe que em todos os momentos dessa conquista me apoiaram de forma incondicional sempre objetivando meu sucesso.*

*Dedico também a todos familiares e amigos que ao longo dessa jornada foram fundamentais para que eu mantivesse minha perseverança e força de vontade para superar os desafios da graduação.*

# Agradecimentos

Agradeço principalmente a todo corpo docente da Faculdade de Computação da Universidade Federal de Uberlândia. Com destaque ao orientador Mauricio Cunha Escarpinati que ao longo da orientação me motivou e fez com que meu conhecimento fosse expandido de forma produtiva, ao professor Luís F. Faina que ao longo da graduação foi sempre apoiador e motivador nos seus conselhos. Por último e não menos importante quero agradecer a banca avaliadora desse trabalho que representam para mim a superação e conquista de um dos objetivos mais importantes de minha vida.

*“A melhor maneira de prever o futuro é inventá-lo. Alan Kay”*

# Resumo

O objetivo deste trabalho é realizar uma investigação sobre o comportamento de algoritmos de reconhecimento facial descritos pela literatura. Foram utilizados dois dos principais algoritmos descritos pela literatura e proposto um conjunto de testes de modo a avaliar o comportamento destes algoritmos a fim de obter resultados que sejam compatíveis com uma possível aplicação de controle de acesso a espaços físicos. As técnicas utilizadas foram o PCA e LDA, que são a base do funcionamento dos algoritmos *Eigenfaces* e *Fisherfaces*, respectivamente. Para os treinamentos e testes dos algoritmos, foram utilizados 3 diferentes bases de dados, a citar, *YALE Database A*, ORL(conhecido também como AT&T) e *Merge*(constituído da junção dos dois bancos anteriores). Para análise dos resultados foram construídas matrizes de confusão que auxiliaram na avaliação dos algoritmos diante o número de falsos positivos, dado crítico para ferramentas de controle de acesso. Dos dois algoritmos estudados o que apresentou melhores resultados foi o *Fisherfaces*, que apresentou uma melhor taxa de acerto e com mínimo de falsos positivos.

**Palavras-chave:** *Eigenfaces*, *Fisherfaces*, Reconhecimento Facial, Componentes Principais, *threshold*.

# Lista de ilustrações

Figura 1 – Reconhecimento das partes - A esquerda a imagem original e a direita as partes reconhecidas. . . . .	17
Figura 2 – Demonstração do algoritmo Camshift - (MARENGONI; STRINGHINI, 2009) . . . . .	18
Figura 3 – Etapas do Reconhecimento Facial . . . . .	19
Figura 4 – Exemplos de faces do banco Yale - (DAVE; CHAO; SRIADIBHATLA, 2010) . . . . .	23
Figura 5 – Face Média e <i>eigenfaces</i> - (DAVE; CHAO; SRIADIBHATLA, 2010) . .	24
Figura 6 – Espaço de Faces <i>Eigenfaces</i> - (SILVA, 2016) . . . . .	24
Figura 7 – Espaço de Faces <i>Fisherfaces</i> - (SILVA, 2016) . . . . .	25
Figura 8 – Sistema Híbrido proposto em (SILVA, 2016) . . . . .	29
Figura 9 – Estrutura da imagens com seus identificadores e variações de poses . .	32



# Lista de tabelas

Tabela 1 – Comparação de Traços Biométricos - Silva 2016. . . . .	14
Tabela 2 – Resultados das métricas e classificadores usados em (SILVA, 2016) . . .	30
Tabela 3 – Banco <i>Yale</i> e <i>Eigenfaces</i> . . . . .	34
Tabela 4 – Banco <i>ORL</i> e <i>Eigenfaces</i> . . . . .	35
Tabela 5 – Banco <i>Merge</i> e <i>Eigenfaces</i> . . . . .	35
Tabela 6 – Banco <i>Yale</i> e <i>Fisherfaces</i> . . . . .	36
Tabela 7 – Banco <i>ORL</i> e <i>Fisherfaces</i> . . . . .	36
Tabela 8 – Banco <i>Merge</i> e <i>Fisherfaces</i> . . . . .	36

# Lista de abreviaturas e siglas

PCA	Principal Component Analysis
LDA	Linear Discriminant Analysis
Camshift	Continuously Adaptive Mean-SHIFT
PDI	Processamento Digital de Imagens
PGM	Portable Gray Map
K-NN	K-Nearest Neighbor

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Objetivos</b>	<b>12</b>
<b>1.2</b>	<b>Organização do Trabalho</b>	<b>12</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
<b>2.1</b>	<b>Biometria</b>	<b>13</b>
<b>2.2</b>	<b>Processamento de Imagens</b>	<b>14</b>
<b>2.3</b>	<b>Visão Computacional</b>	<b>15</b>
2.3.1	Segmentação	16
2.3.2	Reconhecimento De Padrões	16
2.3.3	Rastreamento	17
<b>2.4</b>	<b>Reconhecimento Facial</b>	<b>18</b>
2.4.1	Localização e Detecção Facial	19
2.4.2	Extração de características faciais	21
2.4.2.1	PCA - <i>Principal Component Analysis</i>	21
2.4.2.2	LDA - <i>Linear Discriminant Analysis</i>	22
2.4.2.3	<i>Eigenfaces</i>	22
2.4.2.4	<i>Fisherfaces</i>	24
2.4.3	Reconhecimento da Face	26
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>27</b>
<b>4</b>	<b>MÉTODO</b>	<b>31</b>
<b>4.1</b>	<b>Ferramentas e Fases do Processo</b>	<b>31</b>
4.1.1	Base de Dados de Imagens	32
4.1.2	Detecção Facial	33
4.1.3	Treinamento	33
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>34</b>
<b>5.1</b>	<b><i>Eigenfaces</i></b>	<b>34</b>
<b>5.2</b>	<b><i>Fisherfaces</i></b>	<b>35</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>38</b>
	<b>REFERÊNCIAS</b>	<b>39</b>

# 1 Introdução

Ano após ano observa-se a migração de produtos e serviços para plataformas digitais e acompanhando esse processo surgiu-se a necessidade do desenvolvimento de serviços digitais ágeis para controle de acessos e que preservem a identidade dos indivíduos. Cada indivíduo apresenta características únicas e é através da biometria que tais características podem ser extraídas e trabalhadas para que a identidade do usuário seja comprovada e assim garantir maior segurança no acesso de sistemas e espaços de modo a aumentar a comodidade aos usuários.

O reconhecimento facial se baseia em técnicas biométricas que consistem identificar padrões faciais tais como formato da boca, rosto, distância entre olhos, entre outros(SILVA; CINTRA, 2015). O reconhecimento por faces oferece vantagens, como não precisar de equipamentos especiais para leitura biométrica, além de não precisar de contato direto do indivíduo com o sistema, basta uma câmera ou *webcam* capaz de capturar a face, sendo assim o custo de implementação se torna baixo para sistemas virtuais. No reconhecimento facial contamos com uma alta acessibilidade, considerando que imagens e fotos são fáceis de ser capturadas nos dias atuais, contamos também com uma mensurabilidade, unicidade e permanência médias, que são respectivamente a capacidade de medir os traços dos indivíduos, características únicas dos indivíduos e o tempo que essas características são preservadas com o passar do tempo.

A pretensão deste trabalho é estudar e avaliar o comportamento de algoritmos de reconhecimento facial. Para tanto serão estudados algoritmos que fazem uso de duas técnicas bem conhecidas e difundidas pela literatura, o PCA(*Principal Component Analysis*) ou do português, Análise de componentes principais e o LDA(*Linear Discriminant Analysis*) em português Análise Discriminante Linear. Tais técnicas consistem reduzir a redundância ou mesmo para detecção de padrões(faces, objetos, etc) em um conjunto de dados utilizando procedimentos da álgebra linear e estatística aplicada.

Com base na técnica do PCA foi avaliado um dos algoritmos mais antigos quando o assunto é reconhecimento facial o *Eigenfaces*, que representa uma implementação mais sofisticada da PCA e por isso com resultados melhores do que a técnica PCA sendo usada de maneira “pura”. No algoritmo *Eigenfaces* é aplicado a técnica PCA a fim de extrair os *eigenvectors* que são o conjunto de componentes de maior relevância para a base de dados de treinamento passados e assim gerar o espaço de fases com as *eigenfaces* que melhor representa os dados de treinamento. À premissa básica do algoritmo é gerar uma face média e a partir das *eigenfaces* gerar inúmeras faces similares aos banco de dados usados.

O mesmo foi feito para o algoritmo *Fisherfaces* porém com uso da técnica LDA

também conhecida por reduzir a dimensionalidade(eliminar redundância) dos dados de um conjunto de dados, portanto temos aqui a extração das *fisherfaces* que irão caracterizar a base de dados de acordo com os dados usados no treinamento.

Para validação dos algoritmos implementados neste trabalho foram utilizados dois bancos de dados de imagens de faces, um conhecido como *Yale Face Database A*, que pode ser baixado no sítio (<http://vision.ucsd.edu/content/yale-face-database>), tal banco é constituído por 165 imagens de faces de pessoas em escala de cinza. O segundo banco de dados de imagens é conhecido como ORL(*Our Database of Faces*) ou também como AT&T e é constituído por 400 imagens de pessoas e pode ser obtido através do sítio (<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>).

## 1.1 Objetivos

O objetivo do trabalho é usar ferramentas otimizadas de visão computacional e aprendizado de máquina supervisionado a fim de verificar o desempenho de algoritmos computacionais de reconhecimento facial a fim de indicar sua viabilidade no desenvolvimento de ferramentas de controle de acesso de indivíduos. Mais especificamente serão investigados dois dos principais algoritmos de reconhecimento facial descritos pela literatura, a saber: o *Eigenfaces* e *Fisherfaces* visando avaliar a robustez de cada algoritmo usando um conjunto de 3 bancos de dados, a citar, *Yale*, ORL e Merge(criado com a junção dos dois bancos anteriormente citados).

## 1.2 Organização do Trabalho

No capítulo 2 temos a fundamentação teórica que é responsável por demonstrar os conceitos sobre tipos de biometrias e fases do reconhecimento facial. O capítulo 3 é responsável por citar alguns trabalhos relacionados a reconhecimento facial e os métodos usados pelos autores. No capítulo 4 temos a metodologia usada neste trabalho, que demonstra as ferramentas usadas e todos os dados dos bancos de imagens e como os mesmos foram usados para os treinamentos e testes. No capítulo 5 é responsável pelos resultados e discussões mostrando os dados estatísticos observados durante o trabalho e também pela discussão sobre os resultados obtidos. No capítulo 6 temos a conclusão com relação aos algoritmos e bancos usados no trabalho.

## 2 Fundamentação Teórica

### 2.1 Biometria

A biometria é a ciência que consiste em mensurar as características dos indivíduos, tal ciência demonstra que cada indivíduo é singular, (JAIN; FLYNN; ROSS, 2007) explica que a biometria é a ciência que estabelece a identidade do indivíduo baseado nos seus atributos físicos, químicos e comportamentais, que é exclusivo de cada pessoa.

Seguindo essa linha de raciocínio temos que (SRIVASTAVA, 2013) complementa que o corpo de uma pessoa ou os perfis comportamentais da mesma podem ser usados para identificá-la de maneira única e exclusiva através da biometria, tais dados biométricos podem ser fisiológicos ou comportamentais.

(SRIVASTAVA, 2013) divide a biometria em duas categorias, a citar, características fisiológicas e comportamentais onde cada categoria de biometria pode ser medida usando diferentes métricas.

**Biometria Fisiológica:** Está atrelada a cada corpo humano, sendo difícil ser falsificada e com poucas chances de perder suas características ao longo do tempo. As mais comuns são os DNA's, impressões digitais, características faciais, geometria da mão, palma da mão, retina e íris.

**Biometria Comportamental:** Dependente da condição psicológica ou mesmo do ambiente que a pessoa está inserida, podendo sofrer variações dependendo do estado sentimental do indivíduo ou doenças, que afetam seu jeito de falar, andar ou se expressar de uma maneira geral, como exemplo temos a voz, assinaturas e ritmo de digitação.

Cada gênero biométrico possui vantagens e desvantagens, portanto a definição desses atributos como medida são essenciais para mensurar a eficiência da biometria escolhida. Os atributos são:

1. **Acessibilidade:** O traço métrico é acessível á leitura, como por exemplo a leitura das impressões digitais que possuem equipamentos capazes de capturar.
2. **Desempenho:** Se possui um bom índice de reconhecimento ou seja se a biometria realmente identifica o individuo de maneira correta.
3. **Evasão:** O traço biométrico pode ser fraudado, como exemplo considerar dois gêmeos idênticos onde ambos possuem traços faciais muito semelhantes.

4. **Mensurabilidade:** Se a biometria captura pode ser medida.
5. **Permanência:** Manter as principais características com o tempo, por exemplo a íris dos olhos que mantem sua representação constante durante toda a vida do indivíduo.
6. **Unicidade:** Característica que discrimina unicamente o indivíduo, ou seja aquela representação é exclusiva de um só indivíduo, por exemplo o DNA.
7. **Universalidade:** O traço biométrico se encontra em todos os indivíduos, temos como exemplo, todos os seres humanos tem faces, dedos, olhos entre outras características.

A Tabela 1 (SILVA, 2016) mostra um comparativo entre os vários tipos biométricos encontrados atualmente, entre eles deve-se destacar a biometria por face que será o objeto de estudo deste trabalho.

Tabela 1 – Comparação de Traços Biométricos - Silva 2016.

Biometria	Acessibilidade	Desempenho	Evasão	Mensurabilidade	Permanência	Unicidade	Universalidade
Arcada dentária	Baixo	Médio	Alto	Baixo	Baixo	Baixo	Baixo
Assinatura	Alto	Médio	Alto	Médio	Baixo	Alto	Baixo
DNA	Alto	Alto	Baixo	Baixo	Alto	Alto	Alto
Face	Alto	Baixo	Alto	Médio	Médio	Médio	Alto
Geometria da mão	Médio	Médio	Médio	Alto	Baixo	Médio	Alto
Íris	Médio	Alto	Baixo	Médio	Alto	Alto	Alto
Impressão digital	Alto	Médio	Médio	Alto	Alto	Alto	Médio
Impressão da Palma da mão	Alto	Médio	Médio	Alto	Alto	Alto	Médio
Ouvido	Médio	Alto	Baixo	Médio	Médio	Médio	Baixo
Retina	Alto	Alto	Baixo	Baixo	Alto	Baixo	Alto
Voz	Baixo	Médio	Alto	Médio	Baixo	Alto	Baixo
Ritmo de digitação	Baixo	Baixo	Médio	Baixo	Baixo	Baixo	Baixo

Como é possível ver na tabela, cada atributo para cada tipo de biometria é classificado como baixo, médio e alto, facilitando a escolha do gênero biométrico para cada tipo de aplicação.

## 2.2 Processamento de Imagens

Processamento digital de imagens(PDI) é a manipulação de imagens de maneira que tanto a entrada quanto a saída do processo seja uma imagem. A ideia em processar uma imagem é que ela seja representada com mais efetividade no contexto computacional, uma vez que ela é convertida em dados. Os dados capturados após o processamento são submetidos a técnicas afim de obter variações favoráveis da imagem para que essas variações sejam utilizadas em outros processamentos (CÂMARA et al., 1996).

De acordo com (GONZALEZ; WOODS, 2012) o espectro que vai do PDI até a visão computacional, passa por 3 níveis, sendo eles baixo-nível, nível-médio e alto-nível. O processo de baixo-nível consiste em operações primitivas como redução de ruídos ou melhorias no contraste da imagem, nível-médio que são processos como segmentação ou

classificação, já os de alto-nível são relacionados a tarefas de cognição associadas a visão humana. São tarefas relativas ao baixo-nível (SANTOS, 2011):

1. **Escala de Cinza:** Escala de cinza é uma escala em tons de cinza em que os valores *pixels* da imagem variam de acordo com sua intensidade (luminosidade), numa escala entre preto e branco. É calculada a partir da conversão RGB para YIQ que só normaliza os níveis de brilho da imagem (FILHO; NETO, 1999).
2. **Histograma:** O histograma de uma imagem é um conjunto de números indicando o percentual de *pixels* naquela imagem que apresentam um determinado nível de cinza. Estes valores são normalmente representados por um gráfico de barras que fornece para cada nível de cinza o número (ou o percentual) de *pixels* correspondentes na imagem. Através da visualização do histograma de uma imagem obtemos uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura) (FILHO; NETO, 1999).
3. **Binarização:** O algoritmo de binarização é normalmente utilizado quando é preciso fazer a separação entre o fundo da imagem com os objetos que representam os caracteres. A binarização ou limiarização (do inglês, *thresholding*) é o método mais simples para segmentação de imagens que consiste em separar regiões de não interesse através da escolha de um ponto limiar. Em alguns casos não é possível dividir a imagem em apenas um limiar que resulte em resultados satisfatórios, nesses casos são definidos mais de um ponto de corte (*threshold*) da imagem (FILHO; NETO, 1999).
4. **Filtros:** As técnicas de filtragem são transformações da imagem *pixel* a *pixel*, que não dependem apenas do nível de cinza de um determinado *pixel*, mas também do valor dos níveis de cinza dos *pixels* vizinhos. O processo de filtragem é feito utilizando matrizes denominadas máscaras, as quais são aplicadas sobre a imagem (FILHO; NETO, 1999).

## 2.3 Visão Computacional

O objetivo da visão computacional é tomar decisões úteis sobre objetos físicos reais (SHAPIRO; STOCKMAN, 2001) ou ter como objetivo a construção de descrições de cenas a partir de imagens. A grosso modo, a visão computacional tenta simular a visão humana.

No tópico de PDI, ficou caracterizado os níveis de tratamento das imagens e demonstrado que na fase PDI os processos de baixo-nível consiste em eliminar ruídos e melhorar o contraste das imagens. Na visão computacional entramos no nível-médio (segmentação e reconhecimento de padrões) e posteriormente complementar com o nível-alto



responsável pelos processos cognitivos como o rastreamento de um objeto numa sequência de imagens(SANTOS, 2011).

### 2.3.1 Segmentação

O processo visa segmentar a imagem em regiões ou em objetos diferentes e a segmentação é guiada pelas características do objeto ou região, como a cor ou a proximidade (MARENGONI; STRINGHINI, 2009). A qualidade do detalhamento da segmentação depende da qualidade da imagem analisada, como exemplo, imagine que voce saiu na rua da sua casa e tirou uma foto dos carros estacionados em frente sua casa, se procurarmos por um carro nessa foto, provavelmente a figura do carro ocupou grande parte dessa imagem(regiões grandes), agora imagine que um drone passou pela sua rua e tirou fotos a uma altura de 400 metros, note que na foto tirada pelo drone, os carros ocuparam apenas alguns pontos menores e em relação ao tamanho total da foto esses pontos representam regiões pequenas.

Embora a tarefa seja segmentar, a resolução das imagens são diferentes e o tratamento utilizado no processo de segmentação podem ser diferentes também. Alguns métodos de segmentação são segmentação por detecção de borda, segmentação por corte e segmentação por crescimento de região(MARENGONI; STRINGHINI, 2009).

### 2.3.2 Reconhecimento De Padrões

Reconhecer significa conhecer de novo, e isto implica num processo onde existe algum conhecimento prévio e algum tipo de armazenamento do conhecimento sobre o objeto a ser reconhecido. Esta é a parte onde os sistemas de visão possuem uma intersecção com a área de inteligência artificial. Para fazer o reconhecimento um sistema de visão necessita de uma base de conhecimento dos objetos a serem reconhecidos, esta base de conhecimento pode ser implementada diretamente no código, através, por exemplo, de um sistema baseado em regras, ou esta base de conhecimento pode ser aprendida a partir de um conjunto de amostras dos objetos a serem reconhecidos utilizando técnicas de aprendizado de máquina (MARENGONI; STRINGHINI, 2009).

Reconhecimento de objetos é uma área fundamental na visão computacional e está profundamente relacionada com o reconhecimento de padrões. Objetos em uma imagem apresentam padrões de textura, formas, cores, dimensões entre outros e com o reconhecimento individual desses padrões é possível reconhecer o objeto como um todo, pense que o conjunto das partes pequenas quando juntos formam uma parte grande que contém todas as outras.

As técnicas de reconhecimento de padrões podem ser divididas em dois grandes grupos: estruturais, onde os padrões são descritos de forma simbólica e a estrutura é

a forma como estes padrões se relacionam; o outro grupo é baseado em técnicas que utilizam teoria de decisão, neste grupo os padrões são descritos por propriedades quantitativas e deve-se decidir se o objeto possui ou não estas propriedades(MARENGONI; STRINGHINI, 2009).

Os processos de reconhecimento de padrões podem ainda ser uma mistura das técnicas utilizadas nestes dois grupos, por exemplo, no processo de reconhecimento de faces apresentado em (CÂNDIDO; MARENGONI, 2009), é utilizado um modelo estrutural para determinar o local mais provável para se encontrar partes de uma face (boca, olhos e pele). A figura 1 mostra o reconhecimento das regiões da boca, olhos e testa.

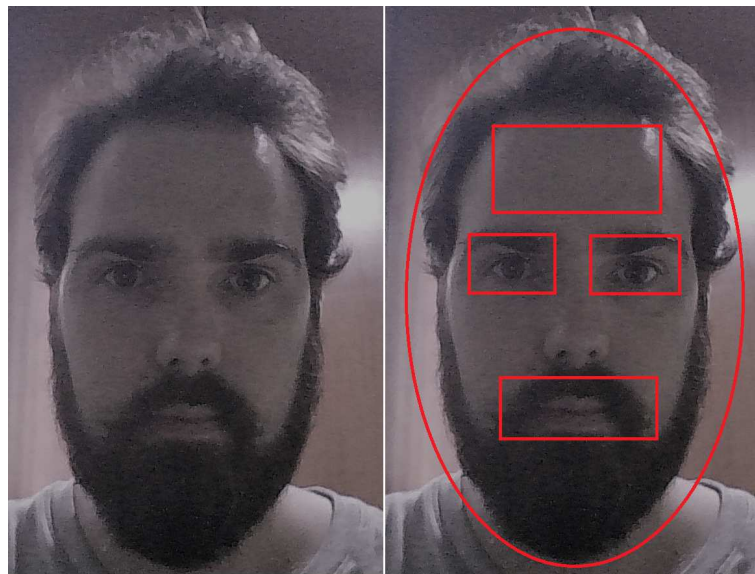


Figura 1 – Reconhecimento das partes - A esquerda a imagem original e a direita as partes reconhecidas.

Cada uma destas partes podem agora ser reconhecida utilizando outro tipo de técnica, por exemplo, os olhos podem ser reconhecidos utilizando uma rede neural, a pele pode ser reconhecida por uma análise estatística e a boca pode ser reconhecida por um critério de distância mínima, todas são técnicas de teoria de decisão.

### 2.3.3 Rastreamento

É um processo de reconhecer um padrão em uma sequência de imagens. O rastreamento poderia ser feito desta forma, porém, a busca em cada imagem de uma sequência sem o uso de qualquer conhecimento específico é relativamente lenta. Os processos de rastreamento atrelam um conhecimento sobre o movimento do objeto que está sendo rastreado para minimizar a busca entre as imagens em uma sequência. Os processos de rastreamento podem ser aplicados em diversas áreas, indo de sistemas de segurança/vigilância até o uso em sistemas de interface humano – computador, por exemplo (SANTOS,

2011). Através de funções é possível aplicar a técnica de rastreamento, tais funções recorrem a dois algoritmos conhecidos como *meanshift* e *camshift*.

O algoritmo *Meanshift* trabalha em cima de distribuições de probabilidade e para rastrear objetos coloridos em sequências de quadros os dados da cor da imagem tem que ser representado como uma distribuição probabilística, então temos o histograma de cores citado na seção de PDI. Distribuições de cores derivadas de sequências de imagens mudam ao longo do tempo, assim o algoritmo *Meanshift* tem de ser modificado para adaptar-se dinamicamente à distribuição de probabilidade de rastreamento. Como uma melhoria ao *Meanshift*, temos o *Camshift*(BRADSKI, 1998).

*Camshift*(*Continuously Adaptive Mean-SHIFT*) é um algoritmo desenvolvido para o rastreamento de cor, possibilitando também o rastreamento de faces. É baseado numa técnica estatística onde se busca o pico entre distribuições de probabilidade em gradientes de densidade. Esta técnica é chamada de “média por deslocamento” (*meanshift*) e foi adaptada no *Camshift* para tratar a mudança dinâmica das distribuições de probabilidade das cores numa sequência de vídeo. Pode ser usada no rastreamento de objetos e no rastreamento de faces, como descrito a seguir. Para cada frame, a imagem é convertida para outra de distribuição de probabilidade de cor através de um modelo de histograma da cor da pele. O centro e o tamanho da face que se quer rastrear são encontrados através do *Camshift* operando na imagem de probabilidade de cores. O tamanho e a localização corrente da face são informados e usados para definir o tamanho e a localização da janela de busca da próxima imagem de vídeo (MARENGONI; STRINGHINI, 2009).

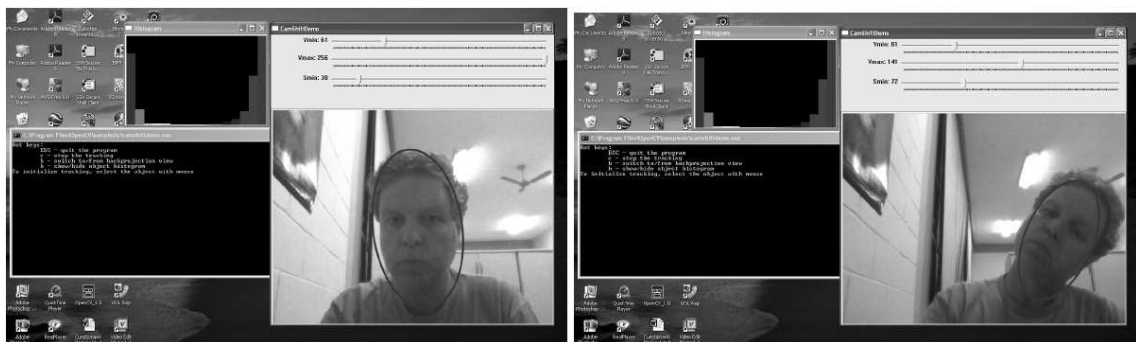


Figura 2 – Demonstração do algoritmo Camshift - (MARENGONI; STRINGHINI, 2009)

## 2.4 Reconhecimento Facial

O reconhecimento facial do ponto de vista do ser humano é uma tarefa trivial e cotidiana, entretanto quando essa tarefa é transportada para o mundo digital usando visão computacional e reconhecimento de padrões a complexidade aumenta criando grandes desafios aos pesquisadores da área(MORAES, 2010).

Ao tentar reconhecer faces em um sistema computacional fazemos uso de um caso particular do reconhecimento de padrões que ano após ano tem ganhado destaque devido a grande aplicabilidade em vários campos, a citar, investigação forense, controle de acesso, reconhecimento de foragidos, aplicação de leis entre outros. As etapas de um sistema de reconhecimento facial é mostrado na Figura 3.

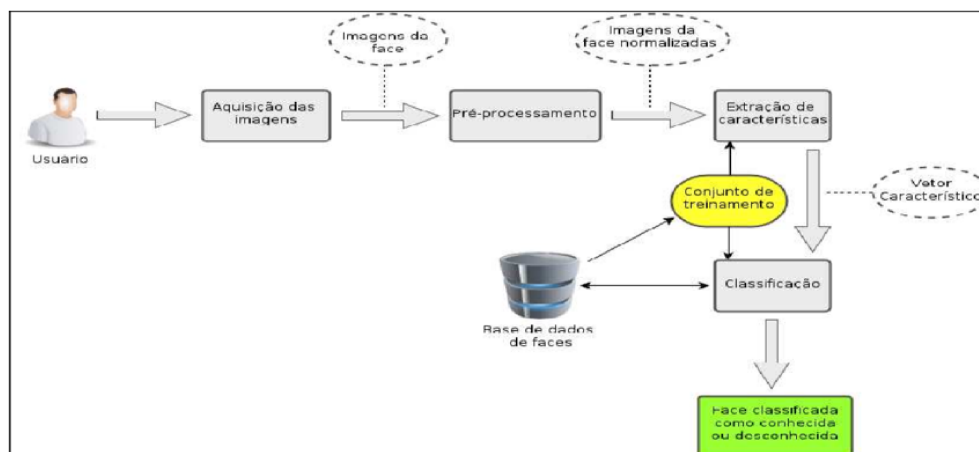


Figura 3 – Etapas do Reconhecimento Facial

Por se tratar de um subproblema de reconhecimento de padrões temos que o reconhecimento facial segue três tarefas fundamentais, sendo elas, localização e detecção da face na imagem obtida, extração de características da face e o reconhecimento da face propriamente dito(MORAES, 2010). As três etapas são detalhadas nas seções a seguir.

### 2.4.1 Localização e Detecção Facial

Um dos primeiros trabalhos relacionados com detecção facial foi proposto por (SAKAI; NAGAO; KANADE, 1972), onde a ideia era criar uma imagem binária que representa-se os contornos da figura e posteriormente extrair características relacionadas a face humana. A técnica apresentou uma taxa de detecção de aproximadamente 91%, porém ela apresentava problemas quando a imagem sofria variações como, barbas e óculos onde ocorreu da técnica não conseguir detectar face nenhuma.

A pesquisa em detecção de faces, se estagnou por longos anos quando na década de 90 (HONG; WAN; JAIN, 1998) foi retomada devido ao surgimento de conferências sobre o tema como *International Conference on Automatic Face and Gesture Recognition* (AFGR) e *International Conference on áudio and Video-Based Authentication* (AVBPA), somado a isso as técnicas de reconhecimento também evoluíram (RIZVI; PHILLIPS; MOON, 1998).

De acordo com (ZHAO et al., 2003) dois fatores contribuíram para o crescimento na área que foram a larga aplicabilidade dessas técnicas em sistemas comerciais e de segurança pública e grande disponibilidade de recursos e tecnologias que se tornaram viáveis nos últimos 30 anos.

Atualmente podemos contar com 4 métodos bem difundidos(LOPES; FILHO; NO, 2005) que são:

1. **Baseado em conhecimento:** Os métodos baseados em conhecimento representam as técnicas de detecção de faces que utilizam alguma base de regras estabelecida a partir do conhecimento prévio sobre o problema, ou seja métodos que possuem regras que definem o que é uma face, de acordo com o conhecimento do pesquisador. Por exemplo, sabemos que faces humanas possuem determinadas características válidas para a maioria de suas instâncias, como: todas as faces teoricamente possuem dois olhos, um nariz e uma boca. Além disso, esses elementos se encontram distribuídos de maneira específica sobre a face. Com este conhecimento é possível estabelecer regras que identificam uma face humana. Este método sofre de algumas desvantagens inerentes a construção do conjunto de regras. Se as regras são muito gerais, corre-se o risco de que o sistema que as utiliza apresente uma alta taxa de falsos positivos, ou seja, elementos erroneamente identificados como face.
2. **Baseado em características invariâtes:** Em contraste com o método visto na no item anterior, os métodos que serão vistos neste item incorporam técnicas que tem por objetivo encontrar características invariantes da face. Particularmente, estes métodos são inspirados na capacidade que os seres humanos possuem de identificar objetos independentes do ponto de vista. A principal desvantagem de tal abordagem é que tais características podem ser corrompidas devido as condições de iluminação ou algum tipo de ruído, comprometendo assim, a eficiência de tal abordagem (YANG; KRIEGMAN; AHUJA, 2002). A cor da pele e a textura da face são as principais características invariantes que podem ser utilizadas para separar a face de outros objetos presentes em uma cena (YANG; KRIEGMAN; AHUJA, 2002). No caso da cor, a informação contida somente na escala de cinza se consitui numa representação básica do objeto, enquanto que a utilização de todo o espectro de cores possível dispõe de informações muito mais poderosas quanto a aparência do objeto.
3. **Baseado em Templates:** Uma técnica clássica de detectar objetos é procurar pelo mesmo dentro da imagem e testar se ele corresponde a um modelo prévio da sua forma. Uma das formas mais comuns de modelar a forma de um objeto é descrevê-lo através de seus componentes geométricos básicos, como círculos, quadrados ou triângulos, esta técnica é denominada *template*. A detecção do objeto, portanto, consistirá em achar a melhor correspondência, definida através de uma função de energia, entre o objeto presente na imagem e o seu molde *template*.
4. **Baseado em Aparência:** Os métodos que compõe a abordagem baseada na aparência *appearance based methods*, recebem este nome devido ao fato de não utiliza-

rem nenhum conhecimento a priori sobre o objeto ou características a ser detectada. Nesta classe de algoritmos surge os conceitos de aprendizado e treinamento, uma vez que as informações necessárias para realizar a tarefa de detecção são retiradas do próprio conjunto de imagens sem intervenção externa.

## 2.4.2 Extração de características faciais

Em uma imagem original que contenha rostos humanos temos um ambiente que pode ser mais ou menos complexo, facilitando ou dificultando a detecção da face no contexto total da imagem. Para extrair as características a face é isolada da melhor maneira possível “recortando” a imagem nos limites da face e posteriormente extraindo as principais características faciais que serão usados no reconhecimento facial (MORAES, 2010). A extração das características faciais gera um conjunto de atributos que representa os traços do indivíduo (PENTEADO, 2009). Neste trabalho dois métodos serão utilizados, sendo eles o PCA aplicado ao *Eigenfaces* e o LDA com *Fisherfaces*.

### 2.4.2.1 PCA - *Principal Component Analysis*

Um método muito difundido no tratamento de características é o de Análise das Componentes Principais(PCA - *Principal Component Analysis*), por ser o primeiro método de reconhecimento facial a obter sucesso e ainda hoje é muito usado para comparação a novas propostas de reconhecimento facial (PENTEADO, 2009).

A PCA é uma técnica matemática que descreve um conjunto de dados usando as principais componentes que representam da melhor maneira o conjunto de dados, usados de maneira a reduzir a dimensionabilidade dos dados ou então detecção de padrões. Quando vista do ponto de vista da transformada o resultado é a mudança de base, uma projeção em um novo espaço onde cada componente seja expressa em ordem de variância ou em contribuição ao conjunto de dados(PENHARBEL et al., 2005).

Na detecção de padrões pode-se empregar a distância euclidiana como critério de classificação. Na redução do conjunto utiliza-se as componentes que mais contribuem nessa variação do espaço, ou seja, as componentes cujos auto-vetores estejam relacionados com os maiores auto-valores da matriz de covariância do conjunto sendo analisado, desprezando os auto-vetores com baixos auto-valores associados.

A técnica PCA é descrita no algoritmo a seguir:

1. **Matriz D:** Organize os dados em uma matriz D, onde a primeira linha será formada pelas componentes da primeira amostra, a segunda linha pelas componentes da segunda amostra e proceda até a N-ésima linha que será formada pela N-ésima amostra com suas componentes.



2. **Vetor Média:** Crie um vetor média  $E$ , formado pelas médias de cada coluna  $M$  da matriz  $D$ .
3. **Subtração da média:** Subtrair de cada item de cada coluna  $M$  da matriz  $D$  a média calculada em  $E$  correspondente a qual o item pertença.
4. **Matriz de Covariância:** Calcule a matriz de covariância da matriz  $D$ .
5. **Auto-Vetores e Auto-Valores:** Da matriz de covariância  $D$ , calcule os auto-vetores e auto-valores gerando uma matriz de auto-vetores e um vetor de auto-valores.
6. **Ordenação dos valores:** Ordene a matriz de auto-vetores pela ordem crescente de auto-valores do vetor de auto-valores correspondentes.

#### 2.4.2.2 LDA - *Linear Discriminant Analysis*

Com mesmo objetivo da técnica PCA, temos que o LDA consiste em reduzir a dimensionabilidade dos dados visando a classificação. A partir de um conjunto de dados multidimensionais rotulados, o LDA gera um conjunto de dados de menor dimensionabilidade que representa as classes dos dados originais.

Tal técnica realiza a redução por meio de um mapeamento dos objetos de alta dimensionabilidade para um espaço de características de menor dimensionabilidade. Para realizar o mapeamento, a LDA cria um vetor  $E$  para construir um espaço de características que busca minimizar a dispersão intra-classe e maximizar a dispersão interclasse, ou seja, aproximar os objetos de uma mesma classe e distanciar objetos de classes distintas (SILVA, 2016).

Os objetos do espaço de dados, ao serem mapeados para o espaço de características gerado pela LDA, são alocados próximos de outros objetos da mesma classe, facilitando o processo de classificação. Por meio da LDA, foi gerado um espaço de características cuja distância entre as projeções de um mesmo indivíduo fosse menor que a distância entre projeções de indivíduos distintos. Desse modo, a representação obtida pela LDA possibilita um meio para a discriminação dos dados pela classe (SILVA, 2016).

#### 2.4.2.3 *Eigenfaces*

O método *Eigenfaces* busca um conjunto de características que não depende das formas geométricas da face (olhos, nariz, orelhas e boca), utilizando toda a informação da representação facial (KSHIRSAGAR; BAVISKAR; GAIKWAD, 2011). Seu funcionamento é similar ao funcionamento do PCA, entretanto é utilizada uma leve otimização para reduzir a matriz de covariância, reduzindo o processamento necessário para fazer o cálculo de seus auto-vetores e auto-valores.

Vale lembrar que no *Eigenfaces* a iluminação é um aspecto importante a ser considerado, uma vez que ele cria os *eigenvectors* baseado nas diferentes variações das faces, bem com as variações de luminosidade, portanto se tivermos muitas faces de uma mesma pessoa com variações faciais consideráveis seu resultado será melhor, porém com a variação de iluminação para cada face, ele usará essas variações para determinar as *eigenfaces* (faces fantasmas) de forma que essas variações prejudiquem a eficiência do algoritmo.

Baseadas na Teoria da Informação, as *eigenfaces* buscam identificar um pequeno número de características que são relevantes para diferenciar uma face de outras faces. Essas características podem ser analisadas apenas com a variação dos valores assumidos pelos *pixels*, em um conjunto de imagens de faces (DINIZ et al., 2013).

O nome *eigenfaces* é atribuído aos auto-vetores (*eigenvectors*) da matriz de covariância das imagens das faces do banco de faces de treinamento por possuírem aspectos de faces. No *eigenfaces* o conjunto de imagens de treinamento irão extrair as componentes mais relevantes da face humana e criar uma face média com base no conjunto de dados de imagens e com a variação dos valores dessas componentes é possível representar uma imensa gama de faces apenas fazendo multiplicações escalares utilizando os auto-valores, na verdade cada face poderá ser representada como combinação linear das diversas *eigenfaces*. Na Figura 4 temos um exemplo das faces do banco de dados Yale.



Figura 4 – Exemplos de faces do banco Yale - (DAVE; CHAO; SRIADIBHATLA, 2010)

Na Figura 5 é possível ver as *eigenfaces* extraídas das faces do banco de dados Yale, bem como a face média que será a base para a reconstrução das diversas faces.



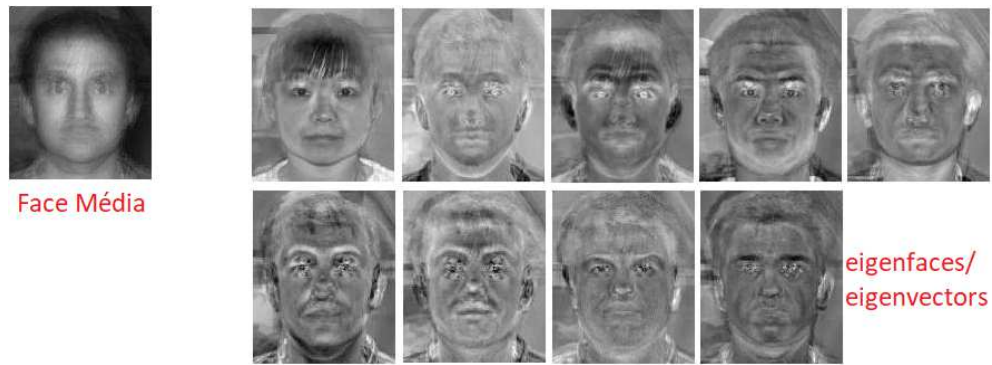


Figura 5 – Face Média e *eigenfaces* - (DAVE; CHAO; SRIADIBHATLA, 2010)

Este método possibilita a classificação de faces a partir do cálculo da distância entre a imagem sendo analisada e a imagem sendo analisada projetada no novo espaço. Se o valor da distância estiver dentro de uma distância limite (*threshold*), a imagem sendo analisada é considerada face, caso contrário é considerada como não face. O método *eigenfaces* é interessante por possibilitar, além da classificação, a reconstrução e a compactação de imagens de faces. Para o cálculo da distância é escolhido as técnicas que mais se adaptem ao banco de imagens usado, sendo as mais usadas a distância euclidiana aplicado ao algoritmo KNN. Na Figura 6 temos a representação de um espaço de faces no plano cartesiano.

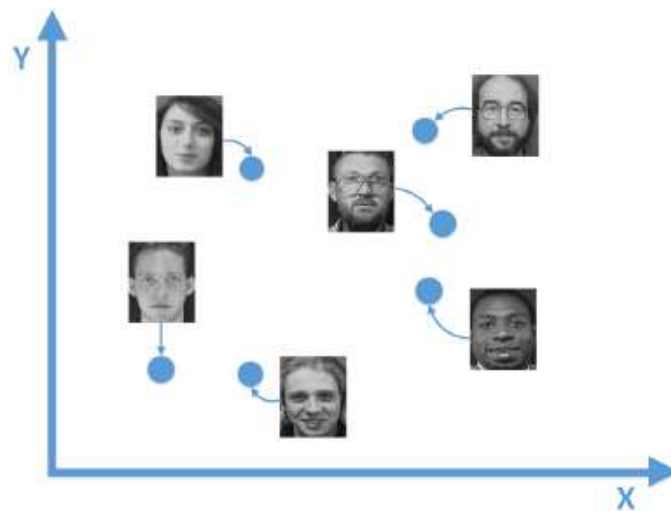


Figura 6 – Espaço de Faces Eigenfaces - (SILVA, 2016)

#### 2.4.2.4 Fisherfaces

Similar *eigenfaces*, as *fisherfaces* podem ser visualizadas como imagens de características onde as características das *fisherfaces* são variações de aparência presentes nas imagens de cada indivíduo, tais como variações de luminosidade, poses e expressões faci-

ais. Assim como as imagens no espaço de dados possuem um valor para cada atributo, os vetores e características possuem um valor para cada *fisherface*(faces fisher). Na Figura 7 é mostrado a projeção das faces relativo ao *Fisherfaces*.

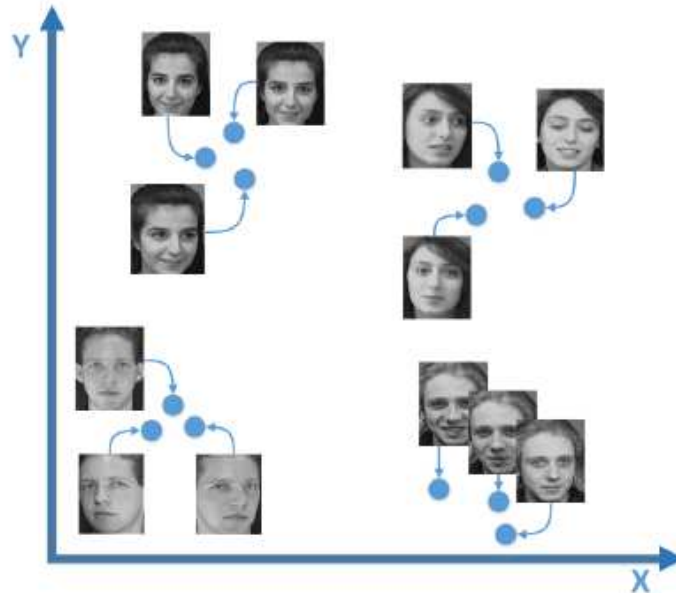


Figura 7 – Espaço de Faces *Fisherfaces* - (SILVA, 2016)

Temos então que quanto mais diferente for uma pessoa da outra, mais distantes deverão estar suas projeções, e quanto mais parecidas forem as imagens de uma pessoa mais próximas deverão estar suas projeções.

O algoritmo *Fisherface* é composto por 8 etapas, sendo elas:

1. **Cálculo da face média por classe:** A face média por classe é o resultado da soma pixel a pixel de todas as imagens de uma mesma classe do conjunto de treinamento dividido pelo total de imagens da classe.
2. **Cálculo de face média geral:** A face média é o resultado da soma pixel a pixel de todas as imagens do conjunto de treinamento dividido pelo total de imagens.
3. **Transformação das imagens em vetores:** A transformação de uma imagem em um vetor consiste em concatenar as linhas da imagem, unindo o último pixel de cada linha com o primeiro pixel da linha seguinte, formando um vetor de *pixels*.
4. **Construção da matriz de dispersão intra-classe:** A matriz de dispersão intra-classe calcula o quanto os dados estão dispersos dentro da própria classe, isto é, o quanto as imagens de um mesmo indivíduo diferem umas das outras.

5. **Construção da matriz de dispersão inter-classe:** A matriz de dispersão inter-classe  $S_b$  calcula o quanto os dados estão dispersos entre as classes, isto é, o quanto as imagens de indivíduos distintos diferem umas das outras.
6. **Cálculo das *fisherfaces*:** As *fisherfaces* do LDA e seus valores associados são, respectivamente, os autovetores e autovalores.
7. **Cálculo dos vetores de características:** As imagens de face são representadas pela soma ponderada das *fisherfaces*. Os pesos dessa combinação foram os vetores de características das imagens de face. Para calcular o peso de cada *fisherfaces* na representação de uma imagem de face, multiplica-se a imagem de face, subtraída da média, por cada uma *fisherfaces*.
8. **Cálculo da Similaridade:** O reconhecimento pode ser realizado por meio do uso de um classificador ou de uma métrica de similaridade. Uma das métricas de similaridade mais comum é a distância euclidiana

### 2.4.3 Reconhecimento da Face

Uma vez que todas as etapas de localização, detecção de faces e a extração de características foram feitas, temos em “mãos” todas as características relevantes representadas pelas componentes principais que mais retratam o banco de imagens usado no treinamento, com isso temos os espaços de faces composto por inúmeras *eigenfaces* ou *fisherfaces* que sendo compostas ou variadas temos a reconstituição de várias faces similares a base de dados do treinamento.

As imagens das faces do banco de imagens são projetadas neste espaço, gerando o vetor de características que representa a face média de uma pessoa e estas são armazenadas na base de dados. Ao se fazer uma consulta usando uma face de teste, esta mesma é projetada no mesmo espaço de faces, seu vetor de características é gerado e deste modo ele é comparado com os vetores armazenados, por meio de algum classificador (PENTEADO, 2009).

### 3 Trabalhos Relacionados

Uma abordagem interessante voltada ao reconhecimento facial com intenção de verificar a identidade da pessoa é a verificação contínua da face, essa proposta é discutida no estudo de (SILVA; SEGUNDO, 2015). A idéia consiste em avaliar a pessoa em frente ao dispositivo de captura de imagem de tempos em tempos, assim garantindo que uma pessoa logar em um sistema, o reconhecimento facial contínuo garantirá que durante o uso do sistema outros individuos não o usem. No artigo citado foi usado um Microsoft Kinect que captura imagens em alta resolução e posteriormente á biblioteca *FaceTracker* é utilizada para detectar as faces em tempo real.

A face uma vez detectada passa por um processo de normalização e posteriormente o reconhecimento usando o algoritmo Padrões Binários Locais(LBP - *Local Binary Patterns*), tal algoritmo foi escolhido pois nos experimentos alcançou uma uma alta taxa de reconhecimento com uma elevada variação na iluminação, sendo assim descartados os algoritmos como *eigenfaces* e *fisherfaces*, comuns nesses tipos de aplicações.

É constatado no artigo que a técnica utilizada na normalização não foi robusta, sendo assim apresentou uma modificação no reconhecimento quando a face efetuava mudanças de ângulo, como olhar para baixo, cima ou para os lados, descaracterizando em partes o reconhecimento contínuo, pois apesar do uso do LBP ter obtido melhores resultados no quesito iluminação, se a normalização não for robusta ele apresenta deficiência em outros quesitos.

Apresentando um abordagem diferente de (SILVA; SEGUNDO, 2015), no trabalho de (PENTEADO, 2009) é priorizado a otimização do reconhecimento facial nos sistemas *web*, ou seja acesso remoto de usuários em sistemas *e-learning*, onde a eficiência e rapidez são pontos extremamente relevantes, pois vale lembrar que algoritmos de processamento de imagens e visão computacional apresentam em geral grande consumo de recursos computacionais.

Na dissertação de (PENTEADO, 2009) a captura é feita por uma *webcam* que coleta as amostras do vídeo, posteriormente para cada frame do vídeo capturado é detectada a face que passará pelos processos de segmentação, pré-processamento para a normalização da iluminação e escala. Na fase seguinte os vetores de características das faces são extraídos, e posteriormente usados na cassificação de faces que já estão previamente cadastradas em alguma base de dados. Baseado na pontuação do casamento dos vetores de características da face de consulta e da face na base de dados o sistema reconhece ou não a face em questão.

No trabalho (PENTEADO, 2009) o algoritmo usado para a detecção de faces no

vídeo é o Viola-Jones, pois o conjunto de passos do algoritmo faz com que as faces sejam detectadas rapidamente e a taxa de detecção é tão boa quanto a outras técnicas apresentadas na literatura. Além da rápida detecção das faces ele apresenta um processamento consideravelmente menor, sendo ideal para aplicações *web* que usam sistemas de autenticação em tempo real.

No âmbito da extração de características o método de PCA foi utilizado, por ter um sucesso em sistemas de reconhecimento facial, além de estar bem difundido no mercado, lembrando que também diminui a dimensionabilidade e mantendo apenas as características de maior importância e descartando as de menor relevância. A técnica PCA gera um conjunto de imagens do domínio que é selecionado para treinamento, na etapa seguinte é criado um espaço de faces e posteriormente uma base de dados dos usuários é criada.

No trabalho foi orientado que os participantes variassem suas poses e expressões faciais para que amostras variadas pudessem ser coletadas, usou-se em média 45 amostras. O uso do Viola-Jones e do PCA foram usados por estar presente na biblioteca *OpenCV* (ITSEEZ, 2015), biblioteca de código livre e multiplataforma com algoritmos já otimizados.

Utilizando a técnica PCA, o algoritmo *Viola-Jones* e a biblioteca *OpenCV* os trabalhos de (PENTEADO, 2009) demonstra um bom desempenho para o sistema, com taxa de reconhecimento de pouco mais 97% em uma base de dados de 43 pessoas, em condições ideais. A análise dos resultados experimentais mostrou que o algoritmo *Viola-Jones* apresentou bom desempenho, com alta taxa de acertos, baixo tempo de processamento e com taxas aceitáveis de falsos negativos e falsos positivos, considerando o tamanho das subjanelas respectivas à distância das faces para a câmera. Conclui-se que é possível com as tecnologias de hoje, aplicar biometria por reconhecimento facial em sistemas EAD ou acessos remotos que possuem uma alta demanda de usuários de forma eficiente.

Uma proposta mais atual de reconhecimento facial é descrito no trabalho de (SILVA, 2016), onde o autor faz uso de um sistema híbrido usando as técnicas PCA e LDA, ao utilizar essas duas técnicas o autor conseguiu minimizar a dimensão das imagens e ao mesmo tempo manter características com grande capacidade de discriminação de faces. No trabalho de (SILVA, 2016) ele faz várias comparações com outros estudos da área de reconhecimento facial e conclui que o extrator de características baseados em PCA e LDA traz avanços na área, pois na literatura o LDA apresenta um desempenho substancial mesmo quando é variado iluminação, expressões diversas das faces e usando diferentes bancos de dados de imagens e o PCA usado para minimizar o número de atributos captados, mas mantendo uma acurácia elevada e consequentemente redução no processamento. A Figura 9 ilustra bem o sistema híbrido proposto.

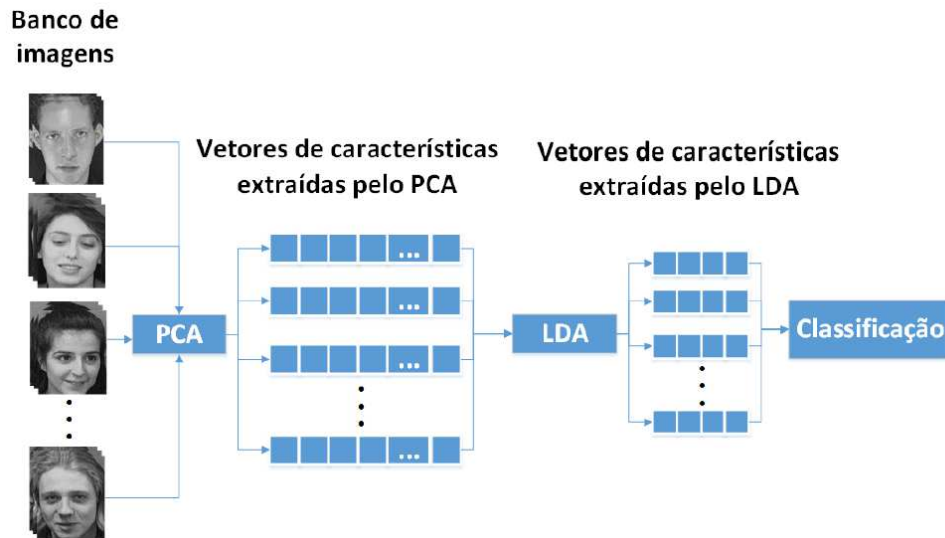


Figura 8 – Sistema Híbrido proposto em (SILVA, 2016)

Em (SILVA, 2016) foi usado os bancos de dados de imagens ORL(AT&T) e JAFFE, pois esses bancos apresentaram características comuns a diversos bancos de imagens de face, tais como faces com diferentes iluminações, expressões faciais e poses. Após o uso das técnicas PCA e LDA para extração de características a classificação das faces foi avaliada experimentalmente por métricas de similaridades e classificadores. As métricas de similaridade usadas foram distância euclidiana, distância Cosseno, distância de Mahalanobis e distância de Manhattan; e nos classificadores: k-NN, SVM e redes neurais MLP para realizar a classificação de faces. Na Tabela 2 é apresentado os resultados gerais.

Os resultados dos experimentos mostram que as métricas Cosseno e Mahalanobis obtiveram a maior taxa de acurácia, 99.25% nos experimentos realizados com o banco ORL, superando trabalhos encontrados na literatura. Com o banco de imagens JAFFE, os classificadores KNN e SVM apresentaram o melhor desempenho com 98.50% de acurácia, apresentando desempenho superior aos métodos encontrados na literatura.

Vale ressaltar que tanto nos trabalhos de (PENTEADO, 2009) e (SILVA, 2016) evidencia-se a preocupação de minimizar o processamento das imagens sem perder a qualidade na acurácia dos reconhecimentos. Considerando que sistemas *web* são limitados computacionalmente e que a demanda por sistemas de controle de acesso via *web* tem aumentado, os dois trabalhos se complementam apresentando melhoras significativas no sistema de biometria de reconhecimento facial.

Tabela 2 – Resultados das métricas e classificadores usados em (SILVA, 2016)

Taxa de acurácia (%)		
Bancos de imagens	ORL	JAFFE
<b>Métricas</b>		
Euclidiana	99.00	96.50
Cosseno	<b>99.25</b>	96.50
Mahalanobis	<b>99.25</b>	97.00
Manhattan	99.00	97.00
<b><i>k</i>-NN</b>		
3	98.75	<b>98.50</b>
5	98.75	<b>98.50</b>
7	98.75	<b>98.50</b>
<b>SVM</b>		
PolyKernel	98.25	<b>98.50</b>
RBFKernel	95.75	98.00
<i>Normalized</i> Polykernel	97.75	96.50
<b>Rede Neural</b>		
MLP	99.00	97.50

## 4 Método

### 4.1 Ferramentas e Fases do Processo

Antes de mais nada é preciso definir quais ferramentas foram usadas para o desenvolvimento do projeto. A linguagem utilizada durante toda a implementação foi Python na versão 3.6.4 em conjunto com as bibliotecas *numpy* (biblioteca completa do python para tratamento com números, vetores e outras operações matemáticas), biblioteca *PIL* (do python para trabalhar com imagens) e a biblioteca de processamento de imagens *OpenCV* na versão 3.4, que possui os algoritmos necessários para esse trabalho.

Python foi escolhido por ser uma linguagem de programação bem difundida, possuir muitas comunidades ativas, apresentar facilidade no seu aprendizado e ser compatível com a biblioteca *OpenCV*. Por sua vez *OpenCV* foi escolhido por apresentar inúmeras ferramentas na área de processamento de imagens, com algoritmos (*Eigenfaces* e *Fisherfaces*) e funções otimizadas, fazendo um “casamento” bem sucedido entre a linguagem e a biblioteca.

Para que um processo de reconhecimento facial seja completo, devemos passar por 4 etapas básicas, a citar, coleta de imagens (em muitos casos usa-se um banco de dados de imagens já padronizado), detecção facial, treinamento com as imagens coletadas ou de um banco já existente e o reconhecimento facial (podendo ser em tempo real ou usando um banco de imagens de teste), tais etapas serão detalhadas nas subseções seguintes, mostrando como foi definida cada uma das etapas e parte de códigos implementados.

Os algoritmos utilizados para treinamento e reconhecimento facial em todo o projeto foi o *Eigenfaces* e *Fisherfaces* sendo que o primeiro usa a técnica PCA para reduzir e eliminar redundância de dados nas imagens, enquanto que o segundo utiliza a técnica LDA para reduzir a dimensionabilidade dos dados. Suas implementações na biblioteca *OpenCV* é bem completa e de fácil uso na linguagem python.

O processo usado foi o de aprendizado supervisionado, onde nas estruturas das imagens dos bancos de dados temos um identificador(id) para cada sujeito, seguido das várias poses feitas por cada um dos sujeitos. Sendo assim foi possível medir o nível de acerto do algoritmo com relação as faces usadas para teste e também saber quais faces foram detectadas erroneamente. A estrutura de cada imagem pode ser visualizada na Figura 9.



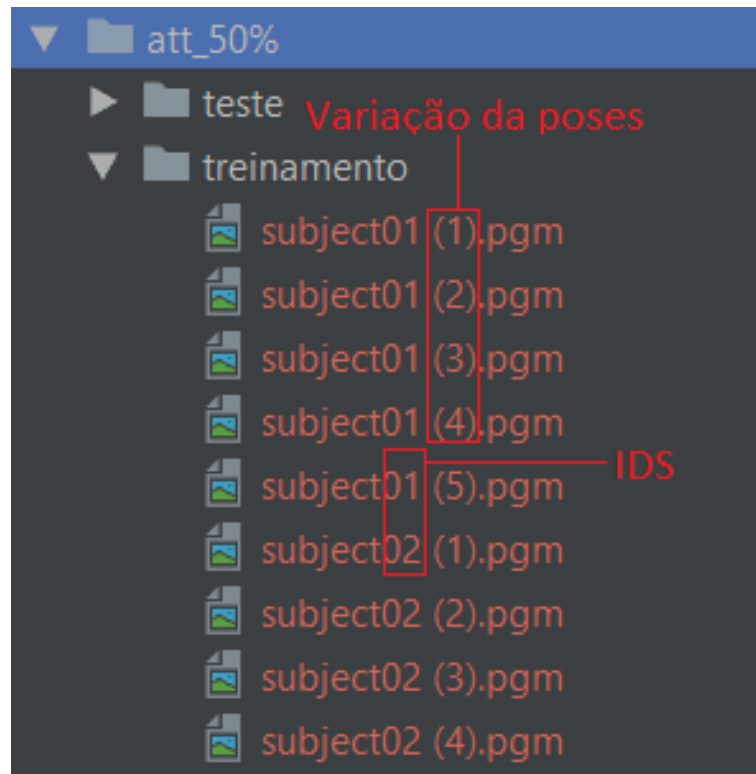


Figura 9 – Estrutura da imagens com seus identificadores e variações de poses

#### 4.1.1 Base de Dados de Imagens

Os bancos de dados usados neste trabalho foi o [Yale Face A](#) e [AT&T](#), sendo que o Yale possui um total de 165 imagens em escala de cinza com dimensões de 320x243 e com extensão da imagem .gif. Foram coletadas fotos de 15 pessoas sendo 10 fotos por pessoa, onde houve variações das faces e configurações diversas como, luz no centro da face, na esquerda da face, a direita da face, uso de óculos, sem óculos, expressões de feliz, triste, sonolento, com piscada.

Foram usados para treinamento 70 imagens, sendo 7 imagens de cada indivíduo, onde do total de 15 pessoas, 5 foram retiradas da fase de treinamento e utilizadas nos testes para avaliação dos falsos negativos e verdadeiros negativos.

O banco de dados AT&T é constituído de 400 imagens com as faces de 40 indivíduos nas dimensão 92x112 e formato PGM(*Portable Gray Map*). Deste banco foram usadas 5 imagens de cada indivíduo para treinamento, sendo que dos 40 indivíduos foram retirados 10 deles com intuito de usá-los na fase de teste, e similar ao banco anterior, ver como se comporta os algoritmos para faces que não constam na base de dados gerada. Do total tivemos 150 imagens para treino e 200 imagens para teste, onde 5 imagens de cada indivíduo foi testado, somando um total de 30 indivíduos que constam na base e os 10 indivíduos retirados da fase de treino.

Um terceiro banco de dados foi construído fazendo a junção dos bancos *Yale A* e

AT&T que foi chamado de *Merge*, onde constitui-se um total de 559 imagens, todas com as dimensões de 92x112 pixels e formato PGM. Para o banco *Merge* obtivemos um total de 55 indivíduos, onde desses 55 foram retirados 15 da fase de treinamento, ficando então apenas 5 imagens de cada um dos 40 indivíduos totalizando 280 imagens para treino. Na fase de teste foram usadas 5 imagens de cada indivíduo do total de 55 indivíduos, incluso os 15 indivíduos retirados na criação da base de dados. Similar aos outros dois bancos citados essa divisão foi feita a fim de avaliar os algoritmos em situações em que na base de dados não possuía alguns indivíduos

#### 4.1.2 Detecção Facial

Na detecção facial foram usados os arquivos com extensão “.xml” conhecido como *Haar* que é responsável por codificar informações sobre a existência de contrastes orientados entre regiões da imagem. As imagens foram amostradas quadro a quadro e as faces foram detectadas e extraídas usando o algoritmo *Viola-Jones* na implementação da biblioteca *OpenCV*, esse arquivo é gerado fazendo o treinamento com algumas centenas de imagens de amostra onde possui um objeto específico que se deseja captar (faces de pessoas por exemplo), esses são os exemplos positivos da amostra, enquanto uma amostra com imagens aleatórias com diversos objetos ou formas diferentes de faces são considerados exemplos negativos, por não conterem o que se deseja.

Os bancos de imagens usados não apresentam amostras negativas uma vez que todas as imagens de treinamento possuem faces de pessoas. Para este trabalho foi escolhido o arquivo *haarcascade\_frontalface\_alt2.xml* disponibilizado no Github<sup>1</sup>, apesar de existir outros esse *Haar* foi selecionado por apresentar maior índice de detecção de faces com os conjuntos de banco de dados utilizados.

#### 4.1.3 Treinamento

Na fase de treinamento foram utilizados 3 limitantes (*thresholds*) diferentes, variando para o algoritmo *Eigenface* o *thresholds* em 2000, 2500 e 3000, enquanto que para o algoritmo *Fisherfaces* os *thresholds* usados foram de 1000, 1200 e 1500. Todas essas variações de *thresholds* foram feitas com todos os bancos.

Conforme recomendado pela literatura (JAIN, 2013) o número de componentes utilizados na implementação do *Eigenfaces* foi de 100. Ainda de acordo com recomendações da literatura (BELHUMEUR; HESPANHA; KRIEGMAN, 1997), para o *Fisherfaces* foi utilizado o número de 39 componentes, uma vez que esse algoritmo necessita de muito menos componentes para representar 100% do conjunto de dados.

---

<sup>1</sup> <https://github.com/opencv/opencv/tree/master/data/haarcascades>

## 5 Resultados e Discussões

Durante os testes o objetivo foi encontrar uma configuração que minimiza ao máximo os valores de falso positivos (caso onde o indivíduo não está na base e é reconhecido de maneira errônea), e maximizar a especificidade (a capacidade do sistema prever a ausência do indivíduo quando ele realmente não está na base de dados).

Dois conceitos importantes usados foram o de sensibilidade e especificidade, é através deles que é possível observar o rendimento real quanto aos resultados. As fórmulas que representam os dois conceitos são:

$$SENSIBILIDADE = VP / (VP + FN)$$

Temos que VP corresponde aos verdadeiros positivos e o FN corresponde aos falsos negativos na fórmula da sensibilidade.

$$ESPECIFICIDADE = VN / (VN + FP)$$

Já para a fórmula da especificidade temos que VN são os verdadeiros negativos e os FP corresponde aos falsos positivos.

### 5.1 Eigenfaces

Nesta seção serão apresentados os resultados obtidos com o algoritmo *Eigenfaces*, quando submetido aos testes com os 3 bancos de dados usados no trabalho. Na tabela 3 são apresentados dos valores de *thresholds* avaliados para cada conjunto de dados *Yale*.

Tabela 3 – Banco *Yale* e *Eigenfaces*

Threshold = 2000				Threshold 2 = 2500				Threshold 3 = 3000			
VP	VN	FP	FN	VP	VN	FP	FN	VP	VN	FP	FN
37,77778	31,11111	2,222222	28,88889	40	26,66667	6,666667	26,66667	46,66667	13,33333	13,33333	26,66667
Acurácia		0,688889		Acurácia		0,666667		Acurácia		0,6	
Sensibilidade		0,566667		Sensibilidade		0,6		Sensibilidade		0,636364	
Especificidade		0,93333		Especificidade		0,80000		Especificidade		0,50000	
Eficiência		0,75		Eficiência		0,7		Eficiência		0,568182	
Taxa Acerto		68,88889		Taxa de Acerto		66,66667		Taxa de Acerto		60	

Observa-se pela Tabela 3 que conforme os limites (*thresholds*) são variados de forma a aumentar seu valor temos um aumento na taxa de falsos positivos e consequentemente uma diminuição na taxa de especificidade, tal resultado é desfavorável uma vez que o valor mais baixo de *threshold* é apresentado falsos positivos e uma especificidade não satisfatória, considerando que o valor da especificidade sai de 0.93 e piora até chegar em 0.50.

Na Tabela 4 é apresentado os valores obtidos com o mesmo algoritmo, porém para o banco de dados AT&T(ORL).

Tabela 4 – Banco ORL e *Eigenfaces*

Threshold = 2000				Threshold 2 = 2500				Threshold 3 = 3000			
VP	VN	FP	FN	VP	VN	FP	FN	VP	VN	FP	FN
29,5	24,5	0,5	45,5	49	23,5	2	26	62,5	15,5	9,5	13
Acurácia	0,54			Acurácia	0,725			Acurácia	0,78		
Sensibilidade	0,393333			Sensibilidade	0,653333			Sensibilidade	0,827815		
Especificidade	0,98000			Especificidade	0,92157			Especificidade	0,62000		
Eficiência	0,686667			Eficiência	0,787451			Eficiência	0,723907		
Taxa de Acerto	54			Taxa de Acerto	72,5			Taxa de Acerto	78		

Apesar de uma melhora na especificidade e uma diminuição dos falsos positivos ainda é evidenciado que o algoritmo apresenta valores não satisfatórios, mesmo quando a quantidade de imagens de treino aumenta com a intenção de melhorar a taxa de reconhecimento facial. Para o banco *Merge* temos os dados da Tabela 5.

Tabela 5 – Banco *Merge* e *Eigenfaces*

Threshold = 2000				Threshold 2 = 2500				Threshold 3 = 3000			
VP	VN	FP	FN	VP	VN	FP	FN	VP	VN	FP	FN
26,1818	29,0909	0	44,3636	41,4545	27,6364	1,45455	28,7273	51,6364	25,4545	3,63636	18,9091
Acurácia	0,55273			Acurácia	0,69091			Acurácia	0,77091		
Sensibilidade	0,37113			Sensibilidade	0,59067			Sensibilidade	0,73196		
Especificidade	1,00000			Especificidade	0,95000			Especificidade	0,87500		
Eficiência	0,68557			Eficiência	0,77034			Eficiência	0,80348		
Taxa de Acerto	55,2727			Taxa de Acerto	69,0909			Taxa de Acerto	77,0909		

Para o banco *Merge*, conjunto maior de imagens, observa-se que para o primeiro *threshold* temos uma especificidade 1(ideal) e com zero falsos positivos, porém fica evidente que a taxa de acerto do algoritmo é baixa, chegando apenas a 55.27%. Com o aumento do *threshold* essa taxa de acerto aumenta para 69.09%, mas as especificidades e os falsos positivos começam a aparecer tornando o algoritmo novamente não satisfatório.

Para os 3 bancos fica claro que os falsos positivos e que a especificidade procurada não é atingido de maneira válida pelo *Eigenfaces*.

## 5.2 *Fisherfaces*

Nesta seção os mesmos testes foram efetuados, mas agora com o algoritmo *Fisherfaces*, que por usar uma técnica mais robusta em termos de classificação tende a apresentar melhores resultados com uma maior confiança com relação as classificações. A Tabela 6, mostra os teste para o bando de dados *Yale*.

Diferentemente do *Eigenfaces*, o *Fisherfaces*, mesmo com valores de *thresholds* mais baixos, consegue obter uma especificidade de 1 e zero falsos positivos. Mesmo com o aumento do *threshold* os falsos positivos só aparecem com o *threshold* igual a 1500 e a especificidade tem uma queda de 1.00 para 0.80, mas não tão drástica como o *Eigenfaces*.

Tabela 6 – Banco Yale e Fisherfaces

Threshold = 1000				Threshold 2 = 1200				Threshold 3 = 1500			
VP	VN	FP	FN	VP	VN	FP	FN	VP	VN	FP	FN
26,66667	33,33333	0	40	33,33333	33,33333	0	33,33333	42,22222	26,66667	6,666667	24,44444
Acurácia	0,6			Acurácia	0,666667			Acurácia	0,688889		
Sensibilidade	0,4			Sensibilidade	0,5			Sensibilidade	0,633333		
Especificidade	1,00000			Especificidade	1,00000			Especificidade	0,80000		
Eficiência	0,7			Eficiência	0,75			Eficiência	0,716667		
Taxa Acerto	60			Taxa Acerto	66,66667			Taxa Acerto	68,88889		

Vale salientar que o valor do maior *threshold* do *Fisherfaces* não chega a ser próximo do menor valor de *threshold* do *Fisherfaces*, mostrando uma maior certeza nos valores obtidos, porém a taxa de acerto não ultrapassa 66.66%, entretanto sem falsos positivos e com a especificidade ainda em 1.00.

A Tabela 7 mostra resultados com o Banco ORL que constitui um número maior de imagens para treino.

Tabela 7 – Banco ORL e Fisherfaces

Threshold = 1000				Threshold 2 = 1200				Threshold 3 = 1500			
VP	VN	FP	FN	VP	VN	FP	FN	VP	VN	FP	FN
37	24,5	0,5	38	49	23,5	3	24,5	60	10	20	9,5
Acurácia	0,615			Acurácia	0,725			Acurácia	0,7		
Sensibilidade	0,493333			Sensibilidade	0,666667			Sensibilidade	0,863309		
Especificidade	0,98000			Especificidade	0,88679			Especificidade	0,33333		
Eficiência	0,736667			Eficiência	0,77673			Eficiência	0,598321		
Taxa de Acerto	61,5			Taxa de Acerto	72,5			Taxa de Acerto	70		

Na contra-mão dos resultados obtidos anteriormente, o *Fisherfaces* já começa a apresentar falsos positivos com primeiro valor de *threshold* e uma especificidade menor que 1.00, sendo então não muito eficiente para essa base de dados, inclusive perdendo em especificidade e falsos positivos pro *Eigenfaces*, entretanto cabe salientar que o valor de *threshold* no *Fisherfaces* é menor. A soma da taxa de acertos também não passa de 72.50% que ainda constitui uma valor baixo para uma sistema de reconhecimento facial.

A Tabela 8 mostra resultados com o Banco *Merge* criado pela junção dos bancos Yale e ORL.

Tabela 8 – Banco Merge e Fisherfaces

Threshold = 1000				Threshold 2 = 1200				Threshold 3 = 1500			
VP	VN	FP	FN	VP	VN	FP	FN	VP	VN	FP	FN
27,6364	29,0909	0	43,2727	42,1818	29,0909	0	28,7273	54,5455	18,9091	10,1818	16
Acurácia	0,56727			Acurácia	0,71273			Acurácia	0,73455		
Sensibilidade	0,38974			Sensibilidade	0,59487			Sensibilidade	0,7732		
Especificidade	1,00000			Especificidade	1,00000			Especificidade	0,65000		
Eficiência	0,69487			Eficiência	0,79744			Eficiência	0,7116		
Taxa de Acerto	56,7273			Taxa de Acerto	71,2727			Taxa de Acerto	73,4545		

Aqui o cenário se apresenta mais generoso, sendo que conseguimos atingir uma taxa de acerto de 71.27% com uma especificidade de 1 e uma contagem de falsos positivos igual a zero com *threshold* em 1200, ou seja temos um cenário onde obtivemos a maior

taxa de acerto mantendo as necessidades de um sistema em não reconhecer um indivíduo que não esteja na base. Para valores de *threshold* maiores a tendência é o algoritmo perder muito em valor de especificidade e consequentemente uma alta em falsos positivos.

## 6 Conclusão

Tendo em vista os teste efetuados é evidente que as imagens das bases de dados são fundamentais para que os algoritmos testados sejam bem sucedidos, pois com o mesmo algoritmo e variando as bases de dados os resultados foram diversificados. Outro ponto importante está no fato dos algoritmos utilizados usarem técnicas interessantes porém antigas com relação a técnicas da atualidade na questão de reconhecimento facial, sendo assim tanto o *Eigenfaces* quanto o *Fisherfaces* são limitados se for pensar a escopo de aplicações reais. No trabalho de (SILVA, 2016) é possível visualizar melhores resultados usando as técnicas PCA e LDA de forma complementar, demonstrando que apesar dos algoritmos não serem eficientes, as técnicas podem gerar bons resultados.

Com relação ao *Eigenfaces*, é nele que o nível de confiança dos classificadores são os piores, e o algoritmo apresenta muitos falsos positivos e uma baixa especificidade, divergindo do que tentou-se buscar nos testes e se pensarmos em uma aplicação prática a probabilidade de o sistema ser falho é grande considerando que a taxa de acerto dele é baixa e ainda com falsos positivos inaceitáveis para realidade de uma aplicação prática como por exemplo, reconhecimento de face para obter acesso a uma sala ou celular.

Já no algoritmo *Fisherfaces* vimos uma robustez maior quando se trata na certeza que o algoritmo tem nas suas respostas, seus valores mais baixos de *thresholds* demonstra isso mantendo uma taxa de acerto máxima de 71.27%, porém sem apresentar falsos positivos e sem ter perda de especificidade. Vale considerar que essa taxa de acerto obtida pelo *Fisherfaces* foi no banco *Merge* que apresenta maiores variações com relação ao conjunto de imagens usadas, uma vez que temos a junção de imagens com diferenças de iluminação ou mesmo tamanho das faces no conjunto.

Apesar da maior confiança nos resultados, não apresentar falsos positivos com *thresholds* até 1200 e ainda manter a especificidade no valor ideal sua taxa de acertos ainda se constituiu baixa para aplicações reais, portanto similar ao *Eigenfaces*, temos que o uso dos dois algoritmos mundo atual são insatisfatórios.

Algoritmos mais recentes como LBPH e uso de redes neurais com *deep learning* são mais eficientes e por isso são propostas mais condizentes com a realidade atual, portanto em trabalhos futuros inevitavelmente tais caminhos sejam importantes no campo do reconhecimento facial.

# Referências

- BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. *Eigenfaces vs. fisherfaces: Recognition using class specific linear projection*. [S.l.], 1997. Citado na página 33.
- BRADSKI, G. R. Computer vision face tracking for use in a perceptual user interface. Citeseer, 1998. Citado na página 18.
- CÂMARA, G. et al. Spring: Integrating remote sensing and gis by object-oriented data modelling. *Computers & graphics*, Elsevier, v. 20, n. 3, p. 395–403, 1996. Citado na página 14.
- CÂNDIDO, J.; MARENGONI, M. Combining information in a bayesian network for face detection. *Brazilian Journal of Probability and Statistics*, JSTOR, p. 179–195, 2009. Citado na página 17.
- DAVE, G.; CHAO, X.; SRIADIBHATLA, K. Face recognition in mobile phones. *Department of Electrical Engineering Stanford University, USA*, 2010. Citado 3 vezes nas páginas 7, 23 e 24.
- DINIZ, F. A. et al. Redface: um sistema de reconhecimento facial baseado em técnicas de análise de componentes principais e autofaces. *Revista Brasileira de Computação Aplicada*, v. 5, n. 1, p. 42–54, 2013. Citado na página 23.
- FILHO, O. M.; NETO, H. V. *Processamento digital de imagens*. [S.l.]: Brasport, 1999. Citado na página 15.
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. [S.l.]: Upper Saddle River, NJ: Prentice Hall, 2012. Citado na página 14.
- HONG, L.; WAN, Y.; JAIN, A. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 20, n. 8, p. 777–789, 1998. Citado na página 19.
- ITSEEZ. *Open Source Computer Vision Library*. 2015. <<https://github.com/itseez/opencv>>. Citado na página 28.
- JAIN, A. K.; FLYNN, P.; ROSS, A. A. *Handbook of biometrics*. [S.l.]: Springer Science & Business Media, 2007. Citado na página 13.
- JAIN, T. S. S. Face recognition. 2013. Citado na página 33.
- KSHIRSAGAR, V.; BAVISKAR, M.; GAIKWAD, M. Face recognition using eigenfaces. In: IEEE. *Computer Research and Development (ICCRD), 2011 3rd International Conference on*. [S.l.], 2011. v. 2, p. 302–306. Citado na página 22.
- LOPES, E. C.; FILHO, J. C. B.; NO, R. T. Detecção de faces e características faciais. *Porto Alegre: PUCRS*, 2005. Citado na página 20.
- MARENGONI, M.; STRINGHINI, S. Tutorial: Introdução à visão computacional usando opencv. *Revista de Informática Teórica e Aplicada*, v. 16, n. 1, p. 125–160, 2009. Citado 4 vezes nas páginas 7, 16, 17 e 18.



- MORAES, J. *Controle de Acesso Baseado em Biometria Facial*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, 2010. Citado 3 vezes nas páginas 18, 19 e 21.
- PENHARBEL, E. A. et al. E-faces-um classificador capaz de analisar imagens e classifica-las como faces ou não faces utilizando o método eigenfaces. *IW orkshop de Visão Computacional*, p. 13, 2005. Citado na página 21.
- PENTEADO, B. E. Autenticação biométrica de usuários em sistemas de e-learning baseada em reconhecimento de faces a partir de vídeo. Universidade Estadual Paulista (UNESP), 2009. Citado 5 vezes nas páginas 21, 26, 27, 28 e 29.
- RIZVI, S. A.; PHILLIPS, P. J.; MOON, H. The feret verification testing protocol for face recognition algorithms. In: IEEE. *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. [S.l.], 1998. p. 48–53. Citado na página 19.
- SAKAI, T.; NAGAO, M.; KANADE, T. *Computer analysis and classification of photographs of human faces*. [S.l.]: Kyoto University, 1972. Citado na página 19.
- SANTOS, R. M. dos. Um estudo de processamento de imagens com opencv. 2011. Citado 3 vezes nas páginas 15, 16 e 18.
- SHAPIRO, L.; STOCKMAN, G. C. *Computer vision*. 2001. Ed: Prentice Hall, 2001. Citado na página 15.
- SILVA, A. A. da; SEGUNDO, M. P. Reconhecimento facial 2d para autenticação continua. 2015. Citado na página 27.
- SILVA, A. L. Redução de características para classificação de imagens de faces. 2016. Citado 10 vezes nas páginas 7, 8, 14, 22, 24, 25, 28, 29, 30 e 38.
- SILVA, A. L.; CINTRA, M. E. Reconhecimento de padrões faciais: Um estudo. In: *Encontro Nacional de Inteligência Artificial e Computacional, 2015, Proceedings ENIAC*. [S.l.: s.n.], 2015. p. 224–231. Citado na página 11.
- SRIVASTAVA, H. A comparison based study on biometrics for human recognition. *IOSR Journal of Computer Engineering (IOSR-JCE) Volume*, v. 15, p. 22–29, 2013. Citado na página 13.
- YANG, M.-H.; KRIEGMAN, D. J.; AHUJA, N. Detecting faces in images: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 24, n. 1, p. 34–58, 2002. Citado na página 20.
- ZHAO, W. et al. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, ACM, v. 35, n. 4, p. 399–458, 2003. Citado na página 19.