UNIVERSIDADE DO VALE DO TAQUARI - UNIVATES
CURSO DE ENGENHARIA DA COMPUTAÇÃO

ESTUDO COMPARATIVO DE ALGORITMOS DE BIOMETRIA FACIAL DISPONIBILIZADOS PELA BIBLIOTECA OPENCV PARA CONTROLE DE ACESSO

Luiz Henrique de Oliveira Galimberti

Lajeado, dezembro de 2018

Luiz Henrique de Oliveira Galimberti

ESTUDO COMPARATIVO DE ALGORITMOS DE BIOMETRIA FACIAL DISPONIBILIZADOS PELA BIBLIOTECA OPENCV PARA CONTROLE DE ACESSO

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Tecnológicas da Universidade do Vale do Taquari UNIVATES, como parte dos requisitos para a obtenção do título de bacharel em Engenharia da Computação.

Orientador: Prof. Alexandre Wolf

Lajeado, dezembro de 2018

AGRADECIMENTOS

À empresa Interact Solutions, pela ideia, tempo, equipamento e espaço cedidos.

Aos colaboradores da empresa Interact Solutions, pela disponibilidade para os testes.

À minha namorada Francine, pelo apoio e compreensão.

A todos os professores, colegas e coordenadores que participaram de minha jornada acadêmica.

RESUMO

O presente trabalho apresenta um comparativo dos algoritmos de reconhecimento facial disponibilizados pela biblioteca OpenCV. Para isso, foi desenvolvido um protótipo de um sistema de controle de acesso baseado em biometria facial. A comparação dos algoritmos é feita considerando situações reais de uso de um sistema de controle de acesso, onde condições ambientais, luminosidade e tempo de exposição do indivíduo são fatores decisivos na viabilidade de utilização da biblioteca. A aplicação responsável pela detecção facial é desenvolvida na linguagem Java em uma plataforma computacional Raspberry Pi, o qual faz chamadas via *webservice* para outra aplicação, também desenvolvida em Java, responsável por realizar o reconhecimento facial, além de armazenar o banco de imagens e informações dos usuários.

Palavras chave: Visão Computacional, Reconhecimento Facial, OpenCV.

ABSTRACT

The present work shows a comparison of the facial recognition algorithms available on OpenCV library. To acomplish it, a access control system prototype based on facial recognition was developed. The comparison between the algorithms was done considering real use situtations of a access control system, where environmental conditions, luminosity and time of exposure of the individual are decisive factors in the feasibility of using the library. The application responsible for facial detection is developed in the Java language on a Raspberry Pi computer platform, which makes calls via *webservice* to another application, also developed in Java, responsible for performing facial recognition, as well as storing the database of images and information of the users.

Key words: Computer Vision, Facial Recognition, OpenCV.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS

API	–	Application Program Interface
FAR	–	False Acceptance Rate
FN	–	False Negative
FP	–	False Positive
FRR	–	False Rejection Rate
GPIO	–	General Purpose Input/Output
JVM	–	Java Virtual Machine
LBP	–	Local Binary Patterns
LBPH	–	Local Binary Patterns Histograms
LDA	–	Linear Discriminant Analysis
LED	–	Light Emitter Diode
OpenCV	–	Open Source Computer Vision Library
PCA	–	Principal Components Analysis
RAM	–	Random Access Memory
SOAP	–	Simple Object Access Protocol
TN	–	True Negative
TP	–	True Positive
UDDI	–	Universal, Description, Discovery and Integration
WSDL	–	Web Service Description Language
XML	–	Extensible Markup Language

SUMÁRIO

1 INTRODUÇÃO

O processo de autenticação de indivíduos tem desempenhado um papel importante na nossa sociedade. Seja para controle de acesso a edificações ou acesso a contas bancárias. A cada dia novas soluções são criadas a fim de proteger bens, sejam eles físicos ou digitais.

Para Miller (1994), os métodos de autenticação podem ser classificados em três categorias: baseados em algo físico que o indivíduo possui, como por exemplo um cartão, ou baseados em uma informação que o indivíduo possui como por exemplo uma senha, podendo ainda ser baseado em características físicas ou comportamentais do próprio indivíduo, como por exemplo o seu rosto.

Segundo Hong e Jain (1998) as abordagens tradicionais de autenticação são baseadas em algo que a pessoa sabe ou possui, porém estes métodos possuem algumas vulnerabilidades, pois as chaves de acesso podem ser transferidas a outra pessoa. Além disso, senhas podem ser esquecidas e cartões podem ser perdidos.

Devido a sua abrangente área de aplicabilidade, o controle de autenticação biométrico, que se resume a um dispositivo de captura de informações biométricas, como íris, impressão digital e face (VETTER, 2010), tem ganho cada vez mais atenção pela indústria e pelos pesquisadores acadêmicos (YANG; KRIEGMAN; AHUJA, 2002) . No controle de autenticação biométrico, a chave de acesso deixa de ser algo que pode ser perdido, esquecido, transferido ou roubado, e passa a ser o próprio indivíduo.

O reconhecimento biométrico por meio da face, dentre todas as alternativas de reconhecimento biométrico, é o que mais tem-se destacado (ZHAO et al., 2003). Isso dá-se pelo fato de que além de exigir menos interação entre o indivíduo e o dispositivo receptor das informações biométricas em comparação às demais características biométricas (íris,

impressões digitais, etc), o reconhecimento de face possui um baixo custo de instalação e manutenção, sendo necessário basicamente um dispositivo de captura de imagens e um servidor rodando uma *Application Programming Interface* (API) de reconhecimento facial.

Dentre as diversas APIs disponíveis no mercado destaca-se a *Open Source Computer Vision Library* (OpenCV), uma biblioteca gratuita e de código aberto que inclui uma grande coleção de ferramentas para o desenvolvimento de aplicações de visão computacional (OPENCV, 2018).

Dado o avanço de ferramentas de reconhecimento facial, este projeto visa desenvolver um protótipo de um sistema de autenticação biométrica baseado em reconhecimento facial utilizando a biblioteca OpenCV, a fim de comparar os algoritmos de reconhecimento de faces disponibilizados pela biblioteca.

1.1 Motivação

A motivação para o presente trabalho surgiu de uma necessidade da empresa Interact Solutions em implantar um sistema de biometria facial para controle de acesso, combinada à falta de estudos comparativos de algoritmos de biometria facial voltados a casos reais de uso, onde a luminosidade, condições ambientais e tempo de exposição do indivíduo são fatores que apresentam impacto direto na viabilidade de utilização do algoritmo.

1.2 Metodologia

O presente trabalho foi desenvolvido na empresa Interact Solutions, sendo esta uma empresa de pesquisa e desenvolvimento de *softwares* onde o autor deste trabalho exerce a função de Arquiteto de Software.

Utilizou-se como amostra para coleta de dados os mais de 60 colaboradores da empresa Interact Solutions. Os dados foram coletados através de um cenário simulado de um sistema de controle de acesso, no qual os colaboradores eram expostos ao protótipo do sistema de controle de acesso baseado em biometria facial por um breve período. Aplicaram se também técnicas e métricas utilizadas em trabalhos relacionados ao tema do presente trabalho,

obtidos através de uma pesquisa bibliográfica.

1.3 Objetivo geral

Este trabalho objetiva realizar um estudo comparativo entre os algoritmos de reconhecimento de faces disponibilizados pela biblioteca OpenCV em um caso de uso real para controle de acesso em uma empresa.

1.4 Objetivos específicos

- Revisão bibliográfica sobre os métodos e algoritmos de detecção e reconhecimento facial fornecidos pela biblioteca OpenCV;
- Desenvolver um protótipo de um sistema de controle de acesso por biometria facial utilizando a biblioteca OpenCV;
- Realizar testes comparativos de desempenho entre os algoritmos utilizados;
- Analisar os dados coletados através dos testes, a fim de verificar a viabilidade da utilização da biblioteca OpenCV em um sistema de controle de acesso baseado em biometria facial;

1.5 Estrutura do trabalho

O presente trabalho está dividido em capítulos, com a finalidade de facilitar o entendimento do mesmo.

O primeiro capítulo oferece uma contextualização ao tema tratado neste trabalho, além de esclarecer os objetivos e motivação do mesmo.

O segundo capítulo apresenta um referencial teórico relacionado ao tema do trabalho. Os assuntos abordados são a Biometria, Sistemas Biométricos, Biometria Facial e alguns métodos de detecção e reconhecimento de faces utilizados no trabalho.

O terceiro capítulo apresenta um referencial teórico quanto às tecnologias utilizadas no desenvolvimento do trabalho.

O quarto capítulo apresenta trabalhos relacionados ao tema deste trabalho.

O quinto capítulo apresenta o desenvolvimento do protótipo e das aplicações relacionadas ao mesmo.

No sexto capítulo são demonstrados os resultados obtidos na comparação dos algoritmos.

O sétimo capítulo apresenta a conclusão deste trabalho.

2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentadas as referências bibliográficas pesquisadas para o desenvolvimento deste trabalho, onde serão descritos conceitos gerais e também mais específicos, contemplando as necessidades de conhecimento.

2.1 Biometria

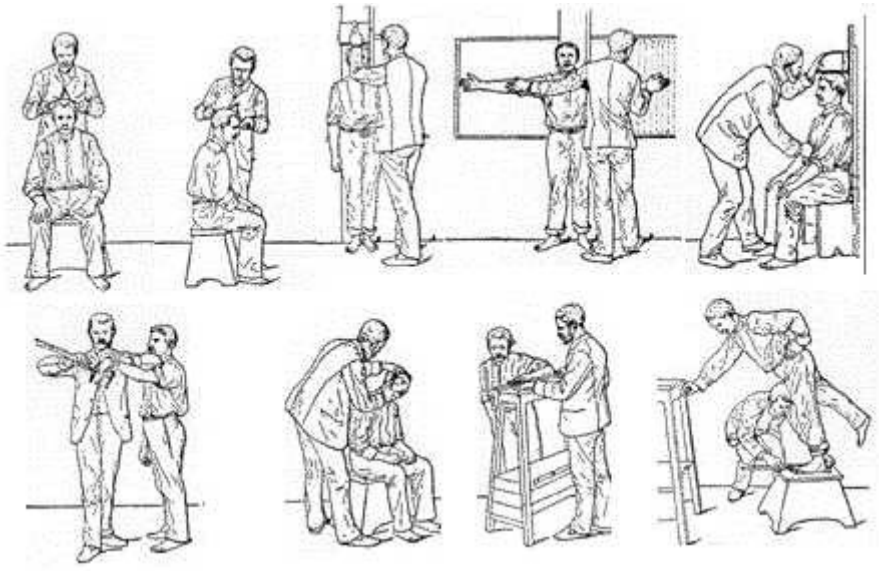
A biometria pode ser caracterizada como o estudo das medidas dos seres vivos, e seu termo é originado da união das palavras gregas *bios* e *metron* que significam vida e medidas, respectivamente (MAGALHÃES; SANTOS, 2003).

Para Jain, Ross e Prabhakr (2007) a biometria é o reconhecimento baseado nas características físicas e comportamentais de um indivíduo. É uma técnica que foi utilizada até mesmo pelos egípcios para o processo de identificação, baseando-se em características da aparência dos indivíduos, como cor dos olhos e cicatrizes (SANTOS, 2007).

Segundo relatos do explorador João de Barros, o povo chinês utilizava a biometria desde o século XIV, carimbando os dedos e mãos de crianças em papel após seu nascimento, com a finalidade de distingui-las umas das outras (MORAES, 2006).

A biometria foi transformada em uma nova área de estudo pelo antropologista francês Alphonse Bertillon, em 1890, quando a utilizou para a identificação de criminosos (MORAES, 2006). O antropologista registrava em cartões medidas do corpo, como comprimento dos pés, estatura, diâmetro da cabeça entre outras (PASQUALI; ARAUJO, 2006). A Figura 1 ilustra algumas das técnicas de medição utilizadas por Bertillon em seu sistema.

Figura 1: Sistema de Bertillon



Fonte: Papiloscopia (2018)

Embora as técnicas utilizadas por Bertillon terem sido desacreditadas após a ocorrência de diversos erros de julgamento, devido às medidas utilizadas por ele não serem únicas, a ideia de usar as medidas e características físicas dos indivíduos perdurou e o sistema continuou sendo utilizado por certo tempo por forças policiais em todo o mundo (BOECHAT, 2008), recebendo o nome de Bertillonage em homenagem ao seu criador.

2.2 Tipos de biometria

Os tipos de biometria podem ser classificados em duas categorias distintas: a primeira categoria se baseia em características físicas do indivíduo, enquanto a segunda tem como base as características comportamentais (COSTA; OBELHEIRO; FRAGA, 2006).

Segundo Costa, Obelheiro e Fraga (2006), as características físicas de um indivíduo são medidas anatômicas relativamente estáveis de uma parte imutável do corpo. Dentre elas podemos citar:

- Impressão Digital: É a biometria mais utilizada atualmente, devido ao seu desempenho satisfatório e custo relativamente baixo de seus aparelhos coletores. Embora a impressão digital seja imutável naturalmente ao longos dos anos, ainda é passível de deterioração. Segundo Jain, Boole e Pankanti (2005), indivíduos praticantes de

atividades braçais podem sofrer alterações em suas impressões digitais devido à cortes e outros ferimentos nos dedos;

- Íris: É uma característica que ganha estabilidade já nos primeiros anos de vida. Sua análise é feita no círculo colorido que envolve a pupila e pode ser feita mesmo com o uso de óculos (LIU; SILVERMAN, 2001);

- Geometria da Mão: Utiliza para análise as medidas da mão como comprimento e largura dos dedos, formato e suas linhas características. Seu ponto fraco ocorre pelo fato de que as mãos podem sofrer alterações em suas características, provocadas por cicatrizes, inchaços e até mesmo ganho de peso (JAIN; ROSS; PRABHAKAR, 2004);

- Retina: Baseia-se na análise de artérias e veias situadas na parte de trás do olho, sendo necessário uma fonte de luz de intensidade baixa para verificar seus padrões únicos. Possui um nível de precisão extremamente alto, sendo que seus padrões, além de serem impossíveis de produzir, são únicos até mesmo em gêmeos idênticos. Sua extração é pouco conveniente pois necessita que o indivíduo foque o olhar em determinado ponto, sendo bastante invasiva e dependente de total cooperação do analisado (JAIN; BOOLE; PANKANTI, 2005);

As características comportamentais baseiam-se nas ações únicas dos indivíduos, aprendidas e adquiridas ao longo de sua vida, analisadas em um determinado intervalo de tempo (NSTC, 2007). Pode-se citar:

- Marcha: Baseada na forma de caminhar do indivíduo, pode ser coletada de forma fácil e sem o conhecimento do indivíduo. Pode sofrer constantes mudanças devido a mudança de peso, fraturas e estado de embriaguez;

- Assinatura: A análise da assinatura pode ser feita de forma estática ou dinâmica (BOECHAT, 2008). A análise estática baseia-se apenas nas características geométricas da assinatura. Já a análise dinâmica utiliza-se da velocidade e pressão nos movimentos do indivíduo ao aplicar a assinatura. Pode sofrer alterações em função do estado físico e emocional do indivíduo (JAIN; BOOLE; PANKANTI, 2005);

- Voz: Embora seja considerada uma característica física, pode ser analisada também de forma comportamental, utilizando-se dos padrões de fala, velocidade e forma de vocalização;

2.3 Sistemas biométricos

Sistemas Biométricos constituem-se da utilização de técnicas avançadas para a identificação de indivíduos utilizando-se de suas características físicas e comportamentais. Pode ser definido como um sistema de reconhecimento de padrões de propósito específico (BOLLE, 2002).

Segundo Jain, Boole e Pankanti (2005), para que uma biometria possa ser utilizada em um sistema biométrico a mesma deve atender determinados requisitos. São eles:

- Universalidade: Deve ser uma característica que todos os indivíduos possuem;
- Unicidade: Deve ser única, ou seja, diferente em cada indivíduo;
- Coletabilidade: Deve ser possível coletá-la e analisá-la;
- Permanência: Deve ser imutável durante determinado período de tempo;

A viabilidade do sistema biométrico em determinado ambiente depende ainda de outros fatores que devem ser levados em consideração (PERRONNIN; JUNQUA; DUGELAY, 2005), como:

- Impostura: Deve ser difícil de imitar;
- Performance: Deve ser possível processar em determinado período de tempo e com uma precisão satisfatória;
- Aceitabilidade: Deve ser não invasiva ou de um modo que o indivíduo aceite fornecer as características ao sistema;

Os sistemas biométricos podem ser separados em dois grupos, os invasivos, que exigem a interação e colaboração do indivíduo para a coleta das características biométricas, e os não invasivos, que podem ser aplicados sem mesmo que o indivíduo saiba que suas características estão sendo coletadas.

Existem duas finalidades na utilização de sistemas biométricos, a verificação, que consiste em comprovar a veracidade da identidade alegada pelo indivíduo com base em suas características biométricas, e a identificação, na qual o indivíduo fornece apenas suas

características biométricas, ficando de responsabilidade do sistema informar a identidade do indivíduo (COSTA; OBELHEIRO; FRAGA, 2006).

2.3.1 Sistema biométrico de controle de acesso

Sistemas biométricos de controle de acesso objetivam impedir o acesso a determinada área ou recurso, sejam eles físicos ou digitais. Os sistemas de controle de acesso físico encarregam-se de permitir ou não a entrada de um indivíduo em determinados locais, em determinados horários, através de sua identificação (SOUZA, 2010).

2.3.2 Métricas de resultado de sistemas biométricos de controle de acesso

Um sistema de controle de acesso baseado em biometria utiliza como resultado uma resposta binária: o indivíduo é quem diz ser ou é um impostor. Porém, mesmo havendo apenas duas respostas possíveis, o resultado pode se enquadrar em quatro cenários (VIEIRALVES; FILHO, 2013):

1. Verdadeiro Positivo – *True Positive* (TP): O indivíduo é legítimo e o sistema o classifica como legítimo;
2. Verdadeiro Negativo – *True Negative* (TN): O indivíduo é um impostor e o sistema o classifica como impostor;
3. Falso Positivo – *False Positive* (FP): O indivíduo é um impostor, porém o sistema o classifica como legítimo;
4. Falso Negativo – *False Negative* (FN): O indivíduo é legítimo, porém o sistema o classifica como impostor;

Com base nos cenários acima, podemos definir as métricas comumente utilizadas para classificar o desempenho de sistemas biométricos (DO VAL; MARCELINO; NETO, 2015). São elas:

- Taxa de Falsa Rejeição – *False Rejection Rate* (FRR): É a chance de o sistema

considerar um indivíduo legítimo como sendo um impostor. É representada pela divisão da quantidade de falsos negativos pela quantidade total de positivos, conforme Equação (1);

$$FRR = \frac{FN}{FN + TP} \quad (1)$$

- Taxa de Falsa Aceitação – *False Acceptance Rate* (FAR): É a chance de o sistema considerar um indivíduo impostor como sendo legítimo. É representada pela divisão da quantidade de falsos positivos pela quantidade total de negativos, conforme Equação (2);

$$FAR = \frac{FP}{FP + TN} \quad (2)$$

2.4 Biometria facial para controle de acesso

A face é a característica biométrica mais utilizada para identificação pessoal (BOECHAT, 2008), sendo frequentemente utilizada em empresas para controle de acesso a áreas restritas (PRODOSSIMO; CHIDAMBARAM; LOPES, 2011).

O processo de reconhecimento de faces é descrito por Jain, Hong e Pankanti (2000) como a análise de características faciais, como boca, nariz, olhos, etc, ou em imagens capturadas por uma câmera digital.

Segundo Jain, Boole e Pankanti (2005), o reconhecimento de faces é muito eficiente no processo de identificação de indivíduos pois, além de ser uma técnica não invasiva e bastante aceita pela sociedade, não demanda envolvimento do indivíduo a ser identificado nem exige longos períodos de espera.

Embora usar a face para identificar conhecidos seja uma tarefa trivial para o ser humano, é uma tarefa bastante complexa para computadores. Mesmo tendo um desempenho razoável em sistemas comerciais, um sistema biométrico facial impõem algumas restrições no

fundo, iluminação e ângulo das imagens utilizadas (PHILLIPS et al., 2000). Devido a isso, Tolba, El-Baz e El-Harby (2006) defendem a ideia de que sistemas biométricos baseados em faces somente tem resultados satisfatórios se utilizados em conjunto de outra característica biométrica.

Segundo Cavalcanti (2005), alterações estéticas, como cabelo e barba, uso de acessórios, como óculos e bonés, são fatores que aumentam as chances de falha no processo de reconhecimento facial.

Sistemas biométricos que utilizam a face como característica biométrica seguem três etapas fundamentais. São elas:

1. Detecção Facial: Fase responsável por detectar e definir a localização de uma ou mais faces em uma imagem estática ou vídeo;

2. Extração de Características: Esta fase é responsável por remover o excesso de informações que rodeiam as faces detectadas, assim como selecionar as melhores características da mesma para serem utilizadas na próxima etapa;

3. Reconhecimento Facial: Esta fase compara as características selecionadas pela fase anterior com outras previamente cadastradas em um banco de dados, sendo responsável por encontrar um registro que se assemelhe ao que precisa ser identificado;

2.5 Detecção facial

Detecção facial é uma técnica que consiste em identificar e localizar faces em imagens digitais através de vários atributos como: aparência da face, formato do rosto ou cabeça, ou a combinação destes (YANG; KRIEGMAN; AHUJA, 2002).

Uma das primeiras técnicas de detecção facial foi apresentada em 1972 por Sakai, Nagao e Kaned. A mesma consistia na criação de uma imagem binária para representar os contornos da face, e então, extrair as características da mesma (SAKAI; NAGAO; KANADE, 1972).

A área em questão ficou sem avanços significativos até os anos 90, quando começou

a receber grande atenção dos pesquisadores (HONG; JAIN, 1998). Segundo Zhao, Chellapa, Phillips e Rosenfeld (2003), este crescimento de interesse se deve basicamente a dois fatores: o aumento na viabilidade de tais sistemas devido ao avanço das tecnologias e recursos, e a larga aplicabilidade destes sistemas na área de segurança.

Até a atualidade, diversas técnicas de diferentes autores foram propostas para a detecção facial. Dentre elas:

- PENG; CHEN; RUAN; KUKHAREV (2005): Busca determinar o centro dos olhos através de múltiplas técnicas, sendo o uso de óculos um fator limitante;
- ROWLEY; BALUJA; KANADE (1998): Utiliza redes neurais de múltiplas camadas para detectar as faces, limitando-se à parte frontal da face;
- VIOLA; JONES (2001): Faz a detecção através de camadas de classificadores treinados. Este algoritmo apresenta uma taxa de detecção de 93,7% em uma base com 130 imagens;

Na escolha do algoritmo, deve-se levar em conta a quantidade de objetos identificados incorretamente como face (falso positivo) e a quantidade de faces não identificadas (falso negativo) (VIOLA; JONES, 2001).

A técnicas utilizada neste trabalho é a de classificadores em cascata. Foram avaliados os classificadores Haar e *Local Binary Patterns* (LBP), fornecidos pela biblioteca OpenCV. Ambos trabalham com imagens em escalas de cinza.

2.5.1 Classificadores em cascata

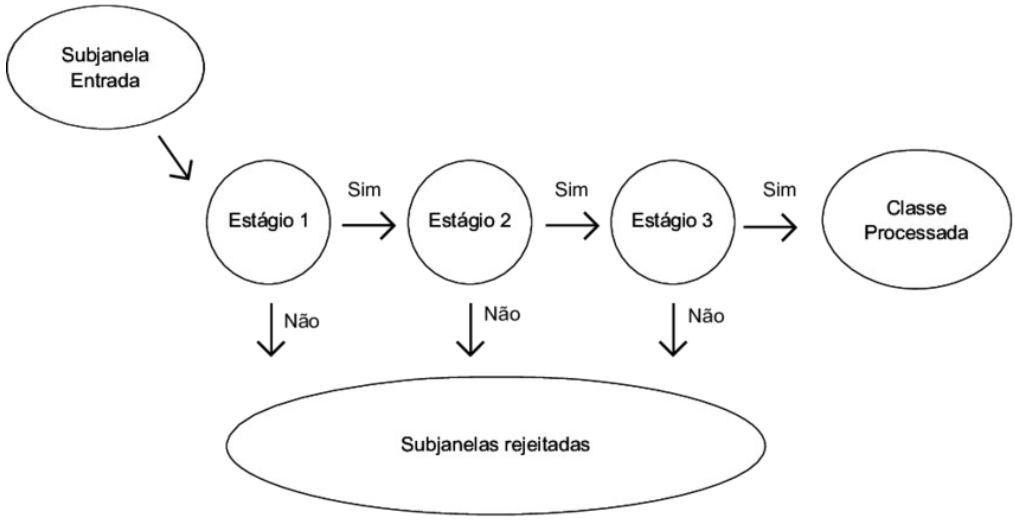
Os classificadores são construídos através de um algoritmo de *machine learning* (aprendizado de máquina) e definem as principais *features* (características) de um objeto. Estas *features* ficam responsáveis por diferenciar os tipos de objetos um dos outros, pois um conjunto de características extraídas de um objeto é altamente diferente de um conjunto de características extraído de outro objeto de tipo diferente.

A busca pela face ou faces na imagem utilizando esta abordagem é feita através de várias varreduras, utilizando janelas e sub-janelas, formadas por retângulos da imagem, com

tamanhos e posições arbitrárias a cada iteração. Nestas varreduras, cada janela ou sub-janela passa por uma cascata de classificadores fracos, que são classificadores simples que utilizam poucas *features* (VIOLA; JONES, 2001), sendo que as camadas dos níveis inicias da cascata objetivam eliminar rapidamente as janelas ou sub-janelas que não contenham o objeto a ser detectado. Para que a face seja detectada, é preciso que passe por todos os níveis da cascata. Através dessa cascata de classificadores fracos é criado um classificador forte.

Uma vez que a face seja detectada em uma janela, é feita uma nova varredura utilizando uma sub-janela menor, com o objetivo de delimitar a localização da face. A varredura é representada pela Figura 2.

Figura 2: Cascata de classificadores



Fonte: De Oliveira (2014)

O número de características utilizadas em cada camada e o número total de camadas da cascata deve ser definido de acordo com a performance que se espera do detector, buscando sempre equacionar baixa taxa de falsos positivos, alta taxa de detecção e baixo tempo de processamento.

2.5.2 Classificador Haar

Embora a motivação inicial ter sido resolver o problema de detecção facial, este classificador pode ser treinado para detecção de uma variedade de objetos (VIOLA; JONES, 2001).

Em seu classificador, Viola e Jones (2001), propõem a utilização de três formatos de *features* retangulares, baseadas nos estudos realizados por Papageorgiou, Poggio e Oren (1998), onde é proposto o uso de "características Haar" no treinamento de classificadores, que podem formar quatro padrões possíveis. Os formatos são:

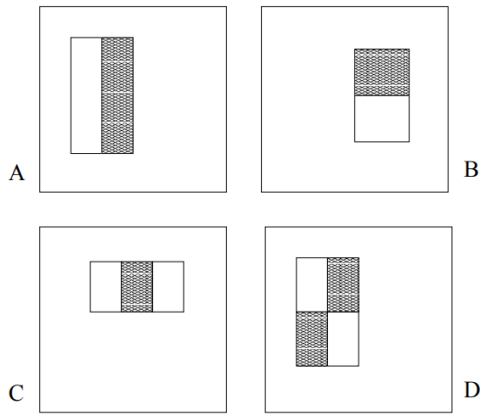
1. Dois retângulos: O valor da *feature* é a diferença entre a soma dos *pixels* de cada uma das duas regiões retangulares;

2. Três retângulos: O valor da *feature* é a soma dos *pixels* de um retângulo central menos a soma dos *pixels* de dois retângulos externos;

3. Quatro retângulos: O valor da *feature* é a diferença entre a soma dos *pixels* de cada par diagonal de retângulos;

Os quatro padrões possíveis podem ser observados na Figura 3, sendo que as *features* A e B são do tipo "dois retângulos", a *feature* C é do tipo "três retângulos" e a *feature* D é do tipo "quatro retângulos".

Figura 3: Padrões possíveis



Fonte: Viola e Jones (2001)

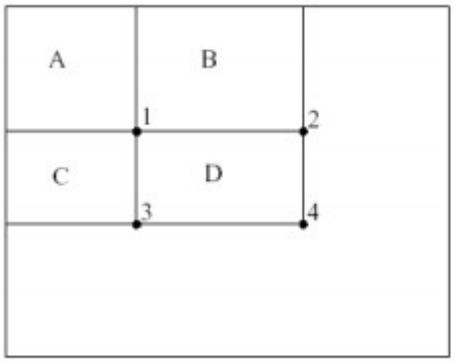
Com a finalidade de otimizar os cálculos dos valores destas *features*, Viola e Jones (2001) propõem a utilização de uma matriz chamada de Imagem Integral, sendo esta uma representação intermediária da imagem original. Na Imagem Integral, cada ponto *x,y* contém o somatório de *pixels* acima e a esquerda de *x,y*, estes inclusos, conforme a Equação (3).

$$ii(x, y) = \sum_{x' \leq x,\, y' \leq y} i(x', y') \tag{3}$$

Utilizando a Imagem Integral, é possível obter de forma ágil o somatório de qualquer retângulo D, ilustrado na Figura 4, através da Equação (4).

$$\sum^{D} pixels = ii(4) + ii(1) - (ii(2) + ii(3)) \tag{4}$$

Figura 4: Retângulo D de posição arbitrária



Fonte: Viola e Jones (2001)

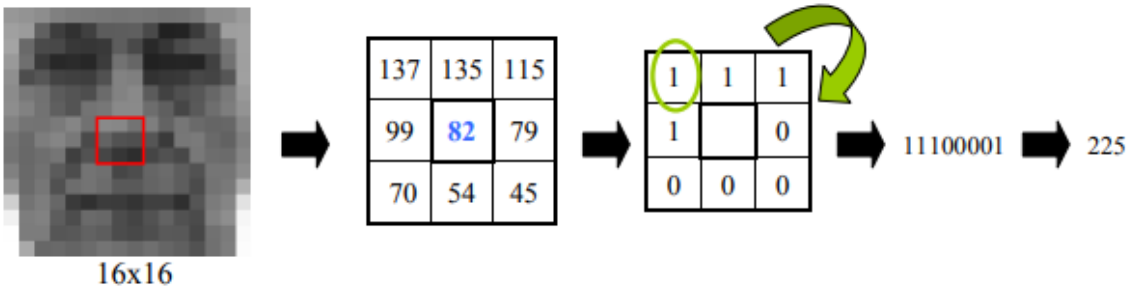
2.5.3 Classificador LBP

LBP foi inicialmente usado para descrever texturas simples onde a relação espacial não é tão significante como é para imagens de faces. O LBP é basicamente dividido em dois descritores de características: sendo um global e um outro local. O global é utilizado para discriminar a maioria dos objetos que não são faces, enquanto o local fornece informações detalhadas de face as quais podem ser usadas não apenas para detecção, mas também para reconhecimento de faces (LÓPEZ, 2010).

As representações das texturas globais e locais são calculadas dividindo a imagem

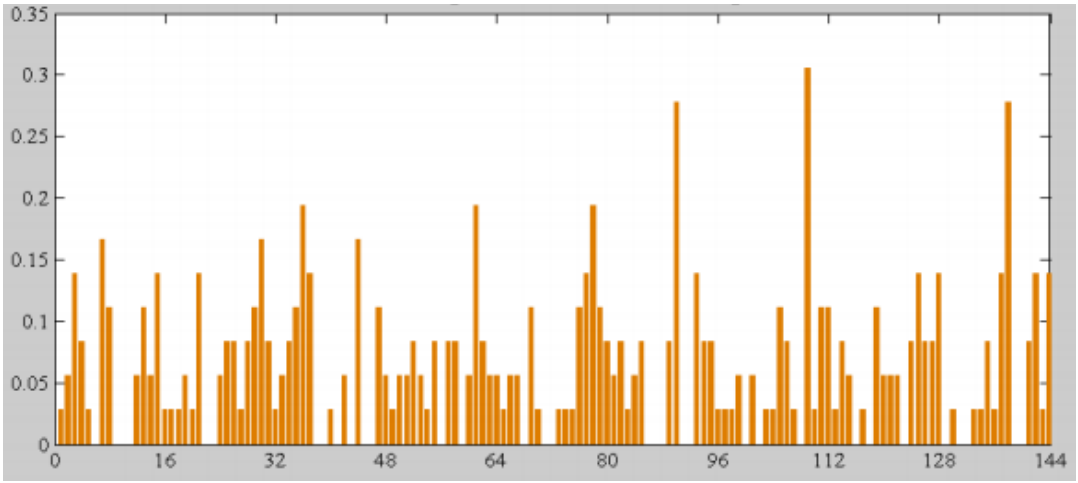
em blocos e computando o histograma de textura para cada um. Os blocos contém 9 *pixels*, sendo em formato de um retângulo 3x3. O valor do *pixel* central de cada bloco é comparado ao de seus vizinhos. Os *pixels* que tiverem o valor maior ou igual ao do pixel central recebem o valor 1, enquanto os outros recebem o valor 0. Após este processo, é lido o valor dos *pixels* atualizados do bloco em sentido horário, formando um número binário. Este número é convertido para decimal e então atribuído ao *pixel* central do bloco. O processo é ilustrado na Figura 5. Em seguida, o valor de cada bloco é convertido em um histograma, representado pela Figura 6.

Figura 5: Obtenção do valor do pixel central de um bloco



Fonte: López (2010)

Figura 6: Histograma de Padrões Binários Locais



Fonte: López (2010)

O resultado é então concatenado em um vetor. Este vetor é utilizado pelo classificador na detecção de faces.

2.6 Extração de características

Um grande problema no processo de reconhecimento facial é o grande número de informações a serem analisadas pelos algoritmos. Se os descritores conterem somente as características mais úteis ao classificador, o mesmo será mais rápido e ocupará menos memória (AGARWAL et al., 2010). Por isso, faz-se necessário reduzir a quantidade de características, ou seja, a dimensionalidade dos dados. A extração de características busca reduzir a dimensionalidade dos dados através da eliminação de informações redundantes (DUDA; HART; STORK, 2001). Contudo, a fim de não comprometer a qualidade da análise, não deve haver perda significativa de informações.

As técnicas de extração utilizadas em reconhecimento facial costumam atender situações específicas e podem ser classificadas em três categorias (TAN, 2006), sendo elas:

1. Locais: Esta técnica possui duas abordagens. A primeira calcula medidas relativas entre componentes faciais utilizando características geométricas como distância e forma. A segunda utiliza informações específicas da face como os olhos, nariz e boca;

2. Globais: Esta técnica representa cada amostra como uma matriz bidimensional;

3. Híbridas: Tais técnicas utilizam informações locais e globais, buscando preservar as vantagens e reduzir as desvantagens de ambas;

Para resolver o problema da dimensionalidade, Sandmann e Senaga (2002), propõem a utilização de métodos de redução de dimensionalidade como a Análise de Componentes Principais (PCA) (SANDMANN; SENAGA, 2002).

2.6.1 PCA

A PCA foi descrita primeiramente por Karl Pearson em 1901. Teve sua primeira aplicação em imagens de face proposta por Sirovich e Kirby (1987), visando a redução da dimensionalidade dos dados preservando informações importantes das imagens originais.

É uma técnica matemática que descreve um conjunto de dados utilizando “componentes principais”. Estes componentes principais são organizados em ordem de

importância. O primeiro componente possui mais informações que o seguinte e assim por diante. A PCA busca construir um conjunto de componentes que resumem os dados originais, reduzindo a dimensionalidade dos dados e preservando os componentes mais significativos (PENTEADO; MARANA, 2008).

Os componentes principais possuem as direções nas quais os dados de entrada têm as maiores variâncias enquanto os demais componentes são considerados menos importantes ou associados a ruídos (DINIZ et al., 2011).

2.7 Reconhecimento facial

O processo de reconhecimento de faces pode ser descrito como: dada uma imagem estática ou vídeo, identificar um ou mais indivíduos utilizando uma base de dados de faces previamente cadastradas (PENTEADO; MARANA, 2008).

Segundo Penteado e Marana (2008), são três as abordagens conhecidas para o reconhecimento de faces:

1. Imagem-para-image: Ambas, amostra e base de dados, são constituídas por imagens estáticas;

2. Vídeo-para-vídeo: Ambas, amostra e base de dados, são constituídas de vídeos;

3. Imagem-para-vídeo: A amostra é um vídeo. O vídeo é comparado com uma base de dados constituídas de imagens estáticas.

A biblioteca OpenCV possibilita, a partir da versão 2.4, o uso da classe FaceRecognizer para reconhecimento facial. Os algoritmos atualmente disponibilizados pela biblioteca são: Eigenfaces, Fisherfaces e *Local Binary Patterns Histograms* (LBPH) (OPENCV, 2018).

2.7.1 Eigenfaces

Para KSHIRSAGAR, BAVISKAR e GAIKWAD (2011), Eigenfaces é uma técnica

que não depende de formas geométricas da face, como olhos, nariz e boca, e busca um conjunto de características utilizando toda a informação da representação facial. A técnica, desenvolvida por Sirovich e Kirby (1987), funciona de forma semelhante a PCA, porém é utilizada uma leve otimização a fim de reduzir a matriz de covariância, consequentemente reduzindo o processamento necessário para calcular seus autovetores e autovalores. (DINIZ et al., 2011).

Para Diniz, Neto, Júnior e Fontes (2011), as Eigenfaces buscam identificar uma pequena amostra de características que possuem relevância no processo de diferenciação de uma face das outras. Essas características podem ser analisadas através da variação dos valores dos *pixels*, em um conjunto de imagens de faces.

A técnica projeta as imagens das faces em um espaço de características, denominado espaço de faces, melhorando a representação da variação entre as faces já conhecidas (OKABE; CARRO, 2014). Este espaço é definido pelas Eigenfaces, que são autovetores do conjunto de faces. Os autovetores descrevem a variação dos pixels associados a diferentes características faciais (DINIZ et al., 2011). Estes autovetores refletem em um conjunto de vetores com direções que representam a maioria das direções dos demais vetores presentes neste espaço (KÖRTING; FILH, 2004).

Segundo Körting e Filh (2004), os autovalores refletem a importância dos autovetores. Os autovetores com maiores autovalores refletirão em componentes mais importantes enquanto autovetores com autovalores nulos poderão ser descartados, desta forma, reduzindo a dimensionalidade dos dados.

Para Izo (2015), a técnica Eigenfaces é dividida em duas etapas:

1. Treinamento: Consiste em calcular as Eigenfaces, isto é, projetar as faces conhecidas da base de dados de imagem, no espaço de faces;

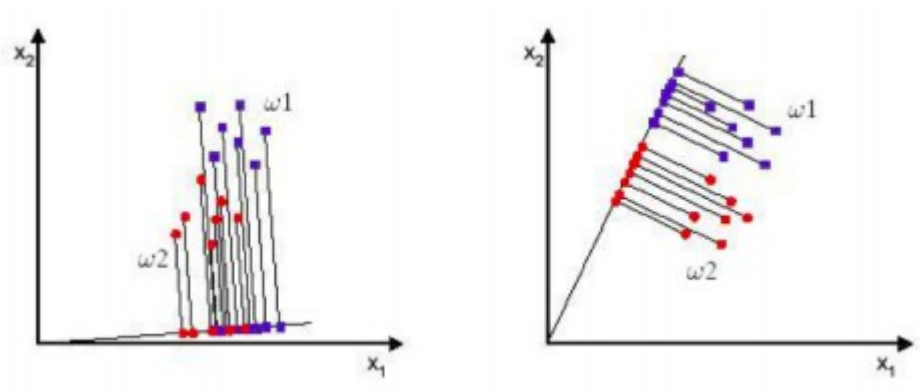
2. Reconhecimento: A identificação de uma face é realizada pela sua projeção no subespaço criado pelas Eigenfaces e em seguida, pela comparação da posição obtida com a posição de faces conhecidas;

2.7.2 Fisherfaces

O Fisherfaces foi primeiramente utilizado em sistemas de reconhecimento de fala e posteriormente aplicado em sistemas de reconhecimento facial (FUJIKAWA, 2016), visando ser uma alternativa com maior acurácia ao Eigenfaces (CARNEIRO, 2012).

A técnica Fisherfaces busca maximizar o raio de variância entre as classes e minimizar a variância dentro das classes (CARNEIRO, 2012). Para Fujikawa (2016), o algoritmo Fisherfaces visa identificar a melhor projeção em retas que maximizem a separabilidade dos conjuntos, podendo ser observado na Figura 7.

Figura 7: Reta de maior separabilidade dos conjuntos



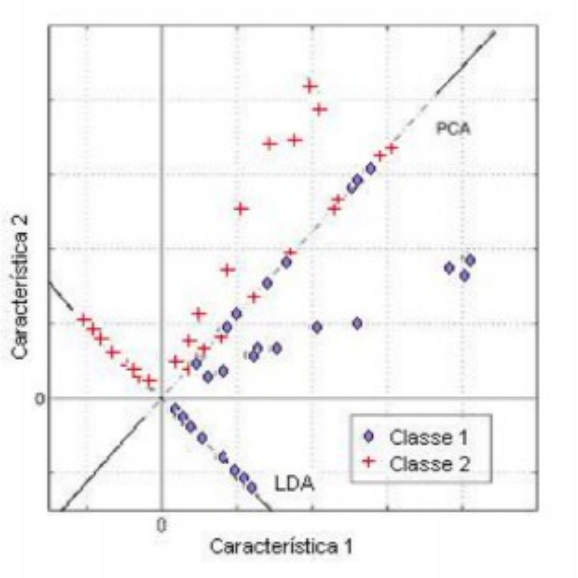
Fonte: Carneiro (2012)

Para Carneiro (2012) e Fujikawa (2016), a maior diferença entre a técnica Eigenfaces e a Fisherfaces é que a primeira não reduz o espalhamento das características dentro da classe. Enquanto o Eigenfaces utiliza PCA, o qual efetua uma compressão da imagem facial com mínimas perdas nos dados, porém sem diferenciar as informações discriminativas da imagem, o Fisherfaces utiliza a Análise de Discriminantes Linear (LDA), que busca não apenas comprimir a imagem, mas também maximizar a discriminação no processo (JESUS et al., 2015). A diferença das duas abordagens para o problema de duas classes é observada na Figura 8.

Para Jesus, Guimarães, Sapucaia, Pimentel, De Souza, Simões e Frias (2015), o método LDA baseia-se nas combinações lineares de variáveis independentes para obter-se a pontuação do objeto observado, posteriormente obtendo-se a probabilidade de tal objeto

pertencer a um dos grupos.

Figura 8: Comparativo PCA e LDA no problema de duas classes



Fonte: Carneiro (2012)

O reconhecimento é realizado através do processo de determinar a que classe uma face desconhecida pertence, pois o conjunto de imagens de treino de face é constituído de diversas classes, onde cada classe representa uma identidade de uma face conhecida (JESUS et al., 2015).

2.7.3 Local Binary Patterns Histograms

Conforme visto na seção 2.5.3, através do método LBP, imagens de faces podem ser representadas por histogramas. O processo de reconhecimento facial utilizando o algoritmo Local Binary Patterns Histograms (LBPH) é semelhante ao de detecção. O reconhecimento ocorre ao obter a identificação da face da base de dados com o histograma de valor mais próximo ao da face a ser identificada.

3 TECNOLOGIAS ENVOLVIDAS

Neste capítulo, são abordadas as tecnologias, metodologias e ferramentas utilizadas no desenvolvimento do trabalho, tais como hardware, linguagem de programação, biblioteca e metodologia.

3.1 Raspberry Pi

Raspberry Pi é um minicomputador criado pela Raspberry Pi Foundation com o intuito de estimular o ensino da ciência da computação nas universidades e escolas (CROTTI et al., 2013).

O Raspberry Pi opera da mesma forma que um computador tradicional, necessitando de um teclado para a entrada de comandos, uma tela e uma fonte de alimentação. Como qualquer outro computador, o Raspberry Pi usa um sistema operacional e o mais utilizado é uma distribuição Linux chamada Raspbian (VUJOVIĆ; MAKSIMOVIĆ, 2014).

Para Vujović e Maksimović (2014), uma das melhores coisas sobre é Raspberry Pi é que ele possui uma gama de utilizações, sendo bastante utilizado como um micro servidor web devido ao seu baixo custo (CROTTI et al., 2013).

O modelo de Raspberry Pi utilizado neste trabalho é o Raspberry Pi 3 Model B, visto na Figura 9.

Figura 9: Raspberry Pi 3 Model B

Fonte: RaspberryPi (2018)

Algumas das especificações do Raspberry Pi 3 Model B (RASPBERRYPI, 2018):

- Processador Quad Core 1.2GHz Broadcom BCM2837 64bit;
- 1GB RAM;
- 4 portas USB 2.0;
- 40 pinos GPIO;
- Porta HDMI;
- Porta CSI para conexão da câmera Raspberry Pi;

3.2 Java

Java é uma linguagem de programação orientada a objetos que foi desenvolvida pela Sun Microsystem em 1992 e anunciada formalmente em 1995. O intuito era desenvolver uma plataforma que pudesse rodar em qualquer dispositivo eletrônico, como televisão, vídeo cassete, entre outros (CAELUM, 2008). Segundo Sopchuk, Agner e Tafuri (2014), a

linguagem Java agora é utilizada para desenvolver aplicações corporativas de grande porte, melhorar a funcionalidade de servidores da Web e para diversos outros objetivos. Em 2009, a Oracle Corporation comprou a Sun Microsystem, tornando-se a dona da linguagem Java e fortalecendo sua marca (CAELUM, 2008).

Para Sopchuk, Agner e Tafuri (2014), o uso da linguagem Java é interessante em projetos onde há muita conectividade e plataformas heterogêneas (ambientes operacionais misturados). A linguagem Java roda em uma Máquina Virtual Java (JVM), que é uma espécie de interpretador, porém, bem mais complexo, assemelhando-se a uma "máquina de mentira", gerenciando memória, threads, pilha de execução, etc (CAELUM, 2008).

A maior característica da linguagem Java é sua portabilidade, sendo "Escreva uma vez, rode em qualquer lugar" seu slogan. Segundo Sopchuk, Agner e Tafuri (2014), a JVM é a principal responsável pela portabilidade provida pela plataforma Java, pois após ser instalada em um ambiente computacional, qualquer aplicação Java pode ser executada sobre a JVM. Desta forma, as aplicações não tem nenhum envolvimento direto com o sistema operacional, comunicando-se apenas com a JVM (CAELUM, 2008).

3.3 OpenCV

OpenCV é uma biblioteca de visão computacional de código aberto, projetada para eficiência computacional com grande foco em aplicações de tempo real (BRADSKI; KAEHLER, 2008). A biblioteca OpenCV foi desenvolvida pela Intel e possui mais de 500 funções que abrangem diversas áreas de visão computacional (MARENGONI; STRINGHINI, 2009). A OpenCV possui interfaces para as linguagem C++, Python e Java, com suporte aos sistemas operacionais Windows, Linux, Mac OS, iOS e Android (OPENCV, 2018).

De acordo com Bradski e Kaehler (2008), visão computacional é a transformação de dados de uma imagem estática ou vídeo em uma decisão ou uma nova representação. Qualquer uma das transformações visa atingir algum objetivo em particular. Um exemplo de decisão seria verificar a presença de um carro em uma determinada imagem ou identificar a quantidade de rostos presentes em uma imagem. Um exemplo de nova representação seria a estabilização de um vídeo ou transformar uma imagem colorida em uma imagem monocromática.

Segundo Bradski e Kaehler (2008), um dos objetivos da biblioteca OpenCV é prover uma infraestrutura de visão computacional de fácil uso, para ajudar pessoas a desenvolverem aplicações sofisticadas de visão computacional rapidamente.

Utilizada em diferentes regiões em todo o mundo, a biblioteca OpenCV possui uma comunidade superior a 47 mil usuários e seu número de downloads é estimado em mais de 14 milhões (OPENCV, 2018). Desde seu lançamento em 1999, a biblioteca OpenCV tem sido utilizada em diversas aplicações, produtos e pesquisas. Essas aplicações incluem monitoramento automático de sistemas de segurança, redução de ruído de imagens na área da medicina, calibração de câmeras, aplicações militares e muitas outras (BRADSKI; KAEHLER, 2008).

Os módulos da biblioteca utilizados neste trabalho são *Object Detection* e *Face Recognition* (OPENCV, 2018), para detecção de faces e reconhecimento de faces respectivamente.

3.4 Web Service

A interação entre aplicações via Internet exige um esforço extra de adequação de ambos os lados, para que as mensagens trocadas sejam compatíveis em alto grau de sintaxe e semântica (ABINADER; LINS, 2006). O *Web Service* possibilita este tipo de integração entre as aplicações, com um baixo grau de interdependência e pouco esforço de adequação.

Segundo Sampaio (2006), um *Web Service* é um aplicativo servidor que disponibiliza um ou mais serviços para seus clientes, de maneira fracamente acoplada. Outra forma de definir um *Web Service* é qualquer arquitetura que envolva o transporte de documentos do tipo *Extensible Markup Language* (XML) entre sistemas de plataformas diferentes (ABINADER; LINS, 2006).

De acordo com Curbera, Duftler, Khalaf, Nagy, Mukhi e Weerawarana (2002), o *framework Web Service* é divido em três áreas: Protocolos de Comunicação, Descrições de Serviços e Descoberta de Serviços, sendo as especificações mais notáveis e estáveis de cada área, respectivamente:

1. *Simple Object Access Protocol* (SOAP): permite a comunicação entre *Web Services*;

2. *Web Services Description Language* (WSDL): fornece uma descrição formal dos *Web Services* legível aos computadores;

3. *Universal Description, Discovery and Integration* (UDDI): diretório, o qual é um registro das descrições dos *Web Services*;

O *Web Service* expõe sua interface para seus usuários através de um documento XML chamado de WSDL. Usando um WSDL é possível descobrir quais são os tipos de dados, formato de mensagens e serviços disponibilizados pelo *Web Service*. Os clientes podem buscar Web Services de duas formas: através do WSDL, se possuírem acesso direto, ou utilizando um serviço de registro via interface UDDI (SAMPAIO, 2006).

4 TRABALHOS RELACIONADOS

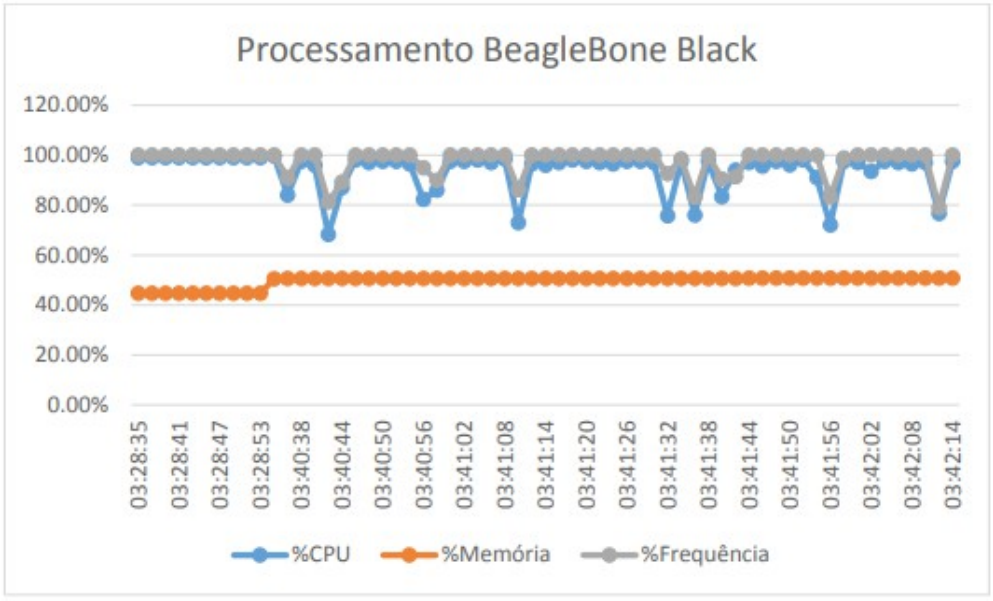
Neste capítulo serão apresentados trabalhos produzidos pela comunidade acadêmica, que possuem relação ao tema tratado neste trabalho. Ambos os trabalhos estão relacionados ao tema deste trabalho, pois o primeiro desenvolveu um sistema biométrico de reconhecimento facial em um sistema embarcado utilizando a biblioteca OpenCV, enquanto o segundo fez um comparativo entre os algoritmos de reconhecimento facial disponibilizados pela biblioteca OpenCV.

O estudo destes artigos auxiliou na definição de uma metodologia para desenvolver a aplicação de detecção facial em um sistema embarcado e a realizar os testes comparativos dos algoritmos.

4.1 Identificação Facial em Linux Embarcado

No trabalho realizado por Nascimento (2015), o autor desenvolveu um sistema de identificação facial utilizando a biblioteca OpenCV, para validação de presenças em salas de aula. O sistema foi implantado em sistema embarcado Beaglebone Black com o sistema operacional Linux. O autor realizou uma análise do processamento do sistema embarcado enquanto os algoritmos de detecção facial eram executados. Os resultados obtidos pelo autor podem ser observados na Figura 10.

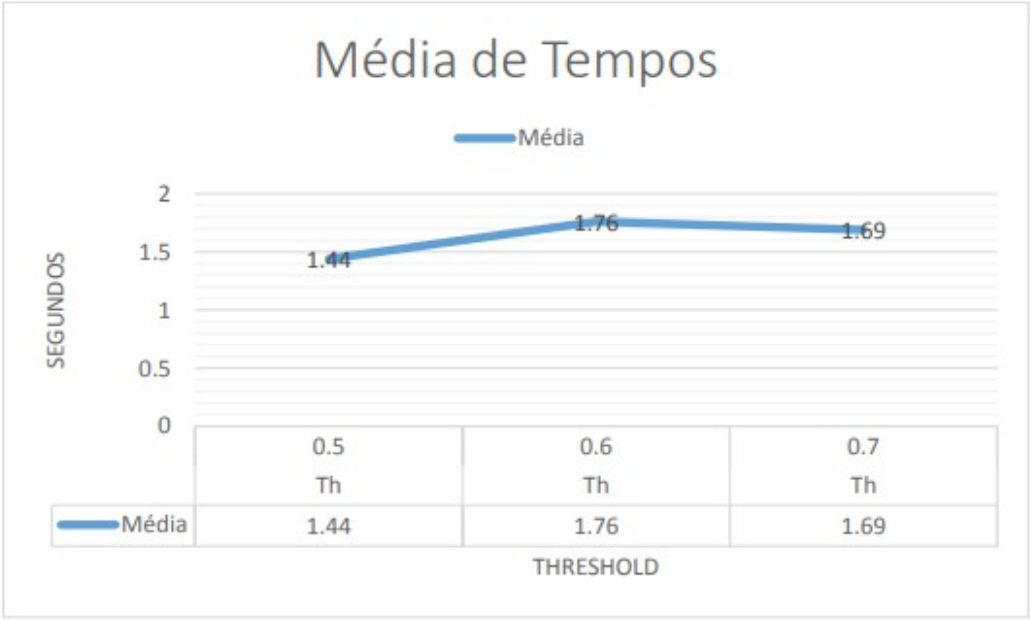
Figura 10: Gráfico de Processamento da BeagleBone Black



Fonte: Nascimento (2015)

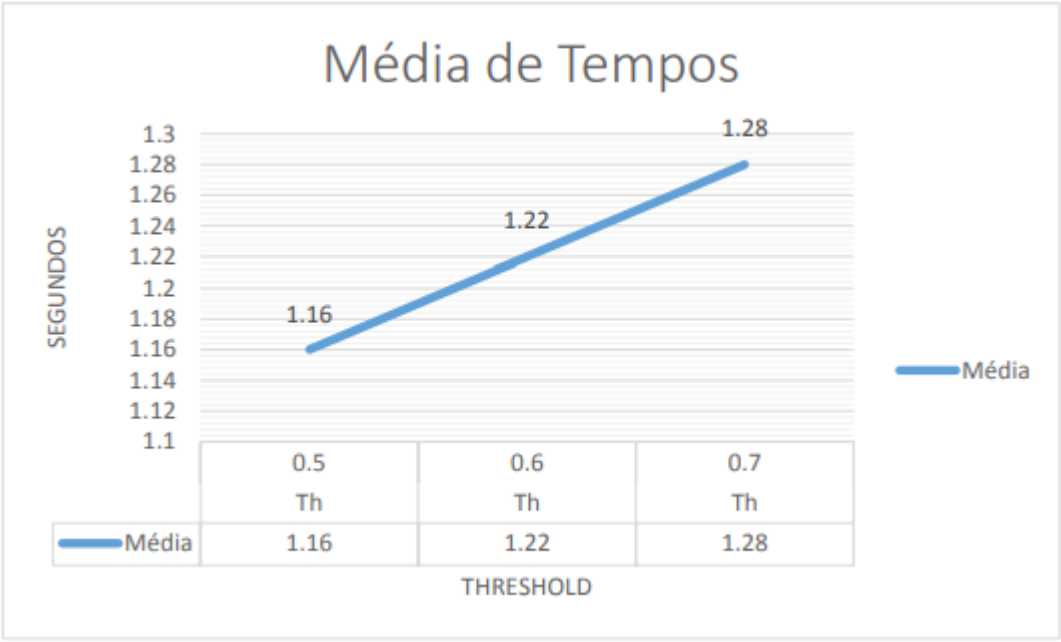
O autor realizou ainda testes relacionados ao tempo necessário para identificação de uma face utilizando os classificadores Haar e LBP, utilizando três limiares, e seus resultados podem ser observados na Figura 11 e Figura 12, respectivamente.

Figura 11: Gráfico de média de tempos (em segundos) x Threshold do algoritmo Haar



Fonte: Nascimento (2015)

Figura 12: Gráfico de média de tempos (em segundos) x Threshold do algoritmo LBP



Fonte: Nascimento (2015)

As conclusões do autor foram de que o algoritmo LBP teve melhores resultados na detecção de faces, tanto no tempo médio de processamento quanto na taxa de sucesso. O autor concluiu ainda que o tempo médio necessário para a detecção de uma face não é elevado.

4.2 Comparação de Algoritmos de Reconhecimento Facial utilizando OpenCV

O trabalho de Narang, Jain, Saxena e Arora (2018), assim como o anterior, baseia-se em um sistema de reconhecimento facial para validação de presença. Neste trabalho os autores comparam os algoritmos de reconhecimento facial fornecidos pela biblioteca OpenCV. A comparação dos resultados obtidos pelos autores é vista na Tabela 1.

Tabela 1: Comparativo de resultados

Critério	Eigenface	Fisherface	LBPH
Fator de confiança	2,000-3,000	100-400	2-5
Threshold	4000	400	7
Princípio da geração da amostra de dados	Baseado em componentes	Baseado em componentes	Baseado em pixels
Princípio básico	PCA	LDA	Histograma
Ruído de fundo	Máximo	Médio	Mínimo
Eficiência	Baixa	Maior que a do Eigenface	Maior entre os algoritmos

Fonte: Adaptado de Narang, Jain, Saxena e Arora (2018)

Os autores concluíram que dentre os algoritmos de reconhecimento facial da biblioteca OpenCV, o LBPH possui a maior precisão e eficiência. Além disso, concluíram que as mudanças de luminosidade não tiveram influência no sistema.

5 DESENVOLVIMENTO

Neste capítulo é apresentado o desenvolvimento do protótipo e das aplicações utilizadas neste trabalho para efetuar a comparação dos algoritmos de reconhecimento facial da biblioteca OpenCV.

5.1 Visão geral

Foi desenvolvido um protótipo para efetuar os testes comparativos de algoritmos de biometria facial. O protótipo é composto de um Raspberry Pi 3 Model B, integrado a uma câmera de 8MP de resolução e a um *speaker*. O sistema embarcado Raspberry Pi 3 foi escolhido a fim de obter um desempenho superior ao de Nascimento (2015), pois possui hardware mais robusto que o do sistema embarcado BeagleBone Black, utilizado no trabalho do mesmo, além de menor custo.

A aplicação que é executada no Raspberry Pi é responsável por fazer a detecção facial utilizando a biblioteca OpenCV e dar um *feedback* sonoro, positivo ou negativo, ao usuário após uma tentativa de reconhecimento facial. O protótipo também é encarregado de enviar um sinal para ativar a abertura de uma porta, que é simulada por um *Light Emitter Diode* (LED) acoplado aos pinos *General Purpose Input/Output* (GPIO) do Raspberry Pi, caso a resposta do reconhecimento facial seja positiva.

Um outro servidor de aplicação, rodando em um sistema Operacional Linux, é encarregado de fazer o reconhecimento facial, assim como permitir o cadastro e manutenção do banco de imagens dos usuários que devem ser reconhecidos. O servidor recebe chamadas via *webservice* do Raspberry Pi quando um rosto é detectado e retorna um sinal de acordo

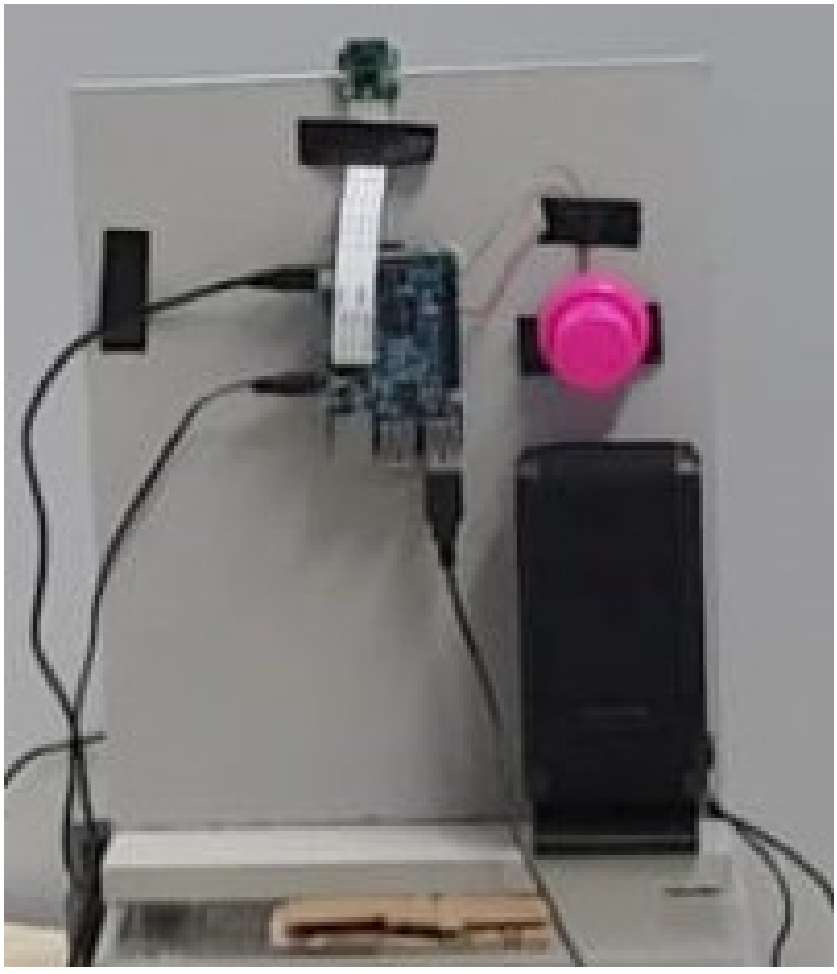
com o resultado do reconhecimento.

Tanto a aplicação que é executada no Raspberry Pi quanto a aplicação de reconhecimento facial que é executada no servidor Linux foram desenvolvidas utilizando a linguagem Java, que é a linguagem de programação utilizada na empresa Interact Solutions onde o protótipo foi desenvolvido.

5.2 Aplicação de detecção facial e *feedback*

A aplicação responsável pela detecção facial é executada em um servidor de aplicação Jetty no protótipo ilustrado na Figura 13, e segue quatro etapas básicas: Captura de um *frame*; Processamento de um *frame*; Chamada *webservice* para a aplicação de reconhecimento; *Feedback*;.

Figura 13: Protótipo



Fonte: Elaborado pelo autor (2018)

5.2.1 Captura de um *frame*

A captura dos *frames* da câmera integrada ao Raspberry Pi é feita utilizando a ferramenta Raspistill, que grava o último *frame* capturado em um diretório tmpfs. A imagem é capturada na resolução máxima suportada pela câmera, sendo 3280×2464 *pixels* de resolução. Antes de ser gravada, é selecionada uma *Region of Interest* (ROI) na parte superior central da imagem, reduzindo sua resolução para 1640×1232 *pixels*. Este recorte é feito pois não é esperado que faces estejam posicionadas fora desta área.

5.2.2 Processamento de um *frame*

O processamento do *frame* é feito em paralelo à etapa de captura de *frame*, e inicia-se pela leitura da imagem capturada na etapa descrita anteriormente, utilizando a biblioteca OpenCV. Como o arquivo foi salvo em um diretório tmpfs, a leitura é feita de forma ágil pois o arquivo já está na *Random Access Memory* (RAM).

Após a imagem ter sido lida e transformada em uma matriz da OpenCV, a mesma é duplicada e redimensionada proporcionalmente para uma altura de 600 *pixels*, pois não são necessárias resoluções altas para a detecção de faces. A imagem original é mantida para ser utilizada posteriormente. Em seguida a imagem é convertida para escalas de cinza e então equalizada para realçar o contraste e facilitar a detecção de faces.

A seguir, é feita a detecção das faces utilizando o classificador de faces frontais LBP fornecido pela biblioteca OpenCV. Foi utilizado o classificador LBP pois este classificador se mostrou mais rápido que os classificadores Haar, além recortar uma área menor da face, removendo desta forma o fundo ao redor da face que pode atrapalhar os algoritmos de reconhecimento.

Caso uma ou mais faces sejam detectadas, é então feita uma detecção de olhos, limitada à área em que a face foi detectada. Após a posição dos olhos ter sido identificada, é feita uma rotação na imagem com base no ângulo de diferença entre a altura do olhos detectados. Desta forma os olhos ficam alinhados horizontalmente, facilitando o reconhecimento. Em seguida é feito o recorte na imagem com a resolução original, utilizando a posição da face detectada. A imagem da face recortada é adicionada em uma lista para ser

enviada para reconhecimento.

Caso não sejam detectados os dois olhos na área da face, o processamento do *frame* é encerrado e inicia-se o processamento do *frame* seguinte.

5.2.3 Chamada *webservice* para aplicação de reconhecimento

Cada face detectada na etapa de processamento de *frame* é enviada para a aplicação de reconhecimento facial via *webservice*. A resposta da chamada é retornada em formato JSON, contendo o nome do indivíduo caso o mesmo tenha sido reconhecido. Após todas as faces terem sido enviadas para reconhecimento é feita a validação das respostas obtidas e então é dado um *feedback.*

5.2.4 Feedback

Caso uma única face tenha sido reconhecida na etapa anterior, o protótipo acende o LED por um instante e oferece um *feedback* sonoro, dando uma saudação utilizando o nome do indivíduo que foi reconhecido. Caso mais de uma face tenha sido reconhecida, o protótipo acende o LED por um instante e oferece um *feedback* sonoro, dando uma saudação genérica sem identificar os indivíduos que foram reconhecidos.

Quando nenhuma face é reconhecida, o protótipo dá apenas um *feedback* sonoro com o som de um bip.

5.3 Aplicação de reconhecimento facial

Esta aplicação roda em um servidor de aplicação Apache Tomcat em um sistema operacional Linux. A aplicação é responsável por receber chamadas via *webservice* com imagens de faces e responder com a identificação do indivíduo. Além disso é responsável por armazenar as faces dos usuários cadastrados e de registrar todas as tentativas de reconhecimento para posterior consulta.

5.3.1 Armazenamento

O armazenamento dos usuários e das tentativas de reconhecimento é feito em um banco de dados MySQL, enquanto as imagens das faces dos usuários cadastrados e imagens das faces usadas em tentativas de reconhecimento são armazenadas em arquivos PNG.

O banco de dados consiste em apenas duas tabelas, sendo que uma armazena o *id* e nome dos usuários e a outra armazena informações das tentativas de reconhecimento, sendo elas: data em que ocorreu; algoritmo utilizado; tempo que levou para o reconhecimento; *id* do indivíduo que o algoritmo reconheceu; percentual de diferença em relação ao *threshold*;.

As imagens das faces dos usuários são armazenadas em pastas com o mesmo *id* do usuário ao qual estão vinculadas. As imagens das tentativas de reconhecimento são armazenadas em uma pasta comum, sendo que no nome da cada arquivo de imagem é utilizada a data e a hora em que a tentativa de reconhecimento ocorreu.

5.3.2 Threshold

Os algoritmos da biblioteca OpenCV, ao efetuarem um reconhecimento, retornam o *id* do indivíduo que foi reconhecido na face informada e um valor chamado de *confidence* que define o grau de semelhança entre a face informada e o indivíduo que o algoritmo identificou. Quanto mais próximo de zero este valor, maior a semelhança. Desta forma, afim de definir se o algoritmo possui convicção suficiente para concluir que o reconhecimento foi verdadeiro, se faz necessário o uso de um valor de *threshold*.

O *threshold* age como um limite para o valor de *confidence*, sendo que reconhecimentos com valores de *confidence* maiores que o *threshold* são considerados falsos e a face é então tratada como não reconhecida.

O *threshold* utilizado neste trabalho é definido dinamicamente. Para isso, utilizam-se 20 faces de desconhecidos de diversos gêneros, etnias e idades. Cada face é submetida a uma tentativa de reconhecimento, e então, utiliza-se como *threshold* o menor valor de *confidence* obtido entre todas elas. Este processo é repetido para cada algoritmo. Desta forma cada algoritmo possui um valor diferente de *threshold*, que é recalculado sempre que o banco de

imagens da aplicação é modificado.

5.3.3 Reconhecimentos

Quando uma chamada via *webservice* é recebida, são feitas 3 tentativas de reconhecimento paralelamente, sendo uma para cada algoritmo de reconhecimento facial analisado neste trabalho. Após a conclusão de cada tentativa de reconhecimento é feito um registro no banco de dados com as informações da tentativa de reconhecimento. Quando as 3 tentativas de reconhecimento são concluídas, é selecionado o indivíduo que obteve o maior percentual de diferença em relação ao *threshold*. Este percentual é definido pela diferença entre o valor de *confidence* do reconhecimento e do *threshold*. Após identificar o indivíduo com o maior percentual, é verificado se o percentual é maior que um certo valor configurável, que será tratado neste trabalho como valor de rigorosidade. Em caso positivo, o indivíduo é usado como resposta da chamada *webservice*. Em caso negativo, o valor 'desconhecido' é usado como resposta da chamada *webservice*.

5.4 Interface visual

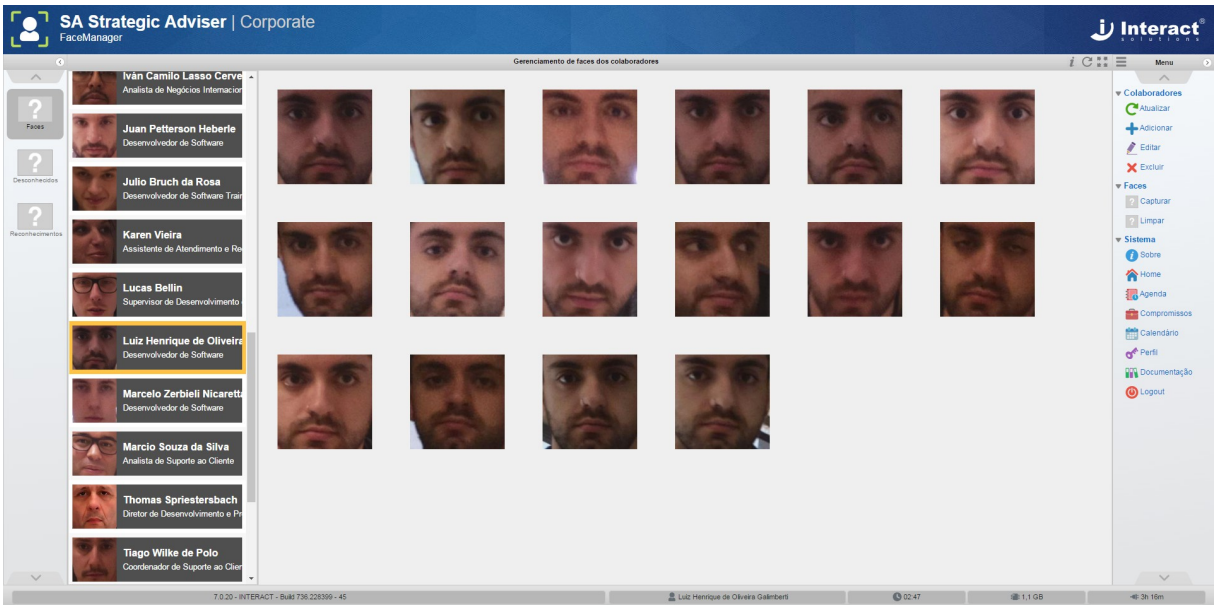
A interface visual para cadastro de usuários, cadastro de imagens e monitoramente de tentativas de reconhecimento foi desenvolvida de forma integrada ao Strategic Adviser, sistema web desenvolvido pela empresa Interact Solutions. A interface foi dividida em três telas que serão explicadas a seguir.

5.4.1 Cadastro de usuários e faces

Nesta tela, que pode ser vista na Figura 14, é possível adicionar usuários já cadastrados no sistema Strategic Adviser. Quando o cadastro de um usuário é efetuado, o *id* deste usuário no Strategic Adviser, juntamente com seu nome completo, são passados para a aplicação de reconhecimento facial via *webservice* para que a mesma efetue o cadastro do usuário. Nesta tela é possível também atribuir imagens de faces aos usuários cadastrados, sendo possível atribuir até 16 faces por usuário. As imagens de faces podem ser inseridas por

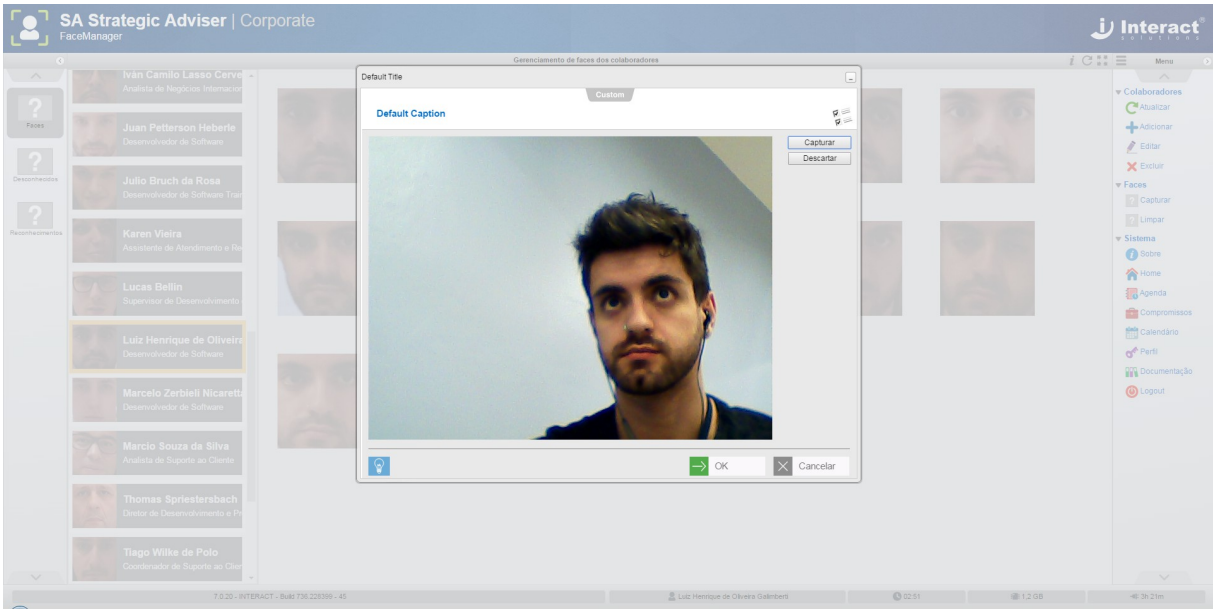
upload ou através de captura pela *webcam*, conforme Figura 15. Ambos os métodos de inserção de imagens possuem validação para que as imagens inseridas só sejam aceitas se contiverem faces. As imagens cadastradas são enviadas em formato base64 via *webservice* ao servidor Linux para então serem armazenadas.

Figura 14: Cadastro de usuários e faces



Fonte: Elaborado pelo autor (2018)

Figura 15: Cadastro de face por webcam

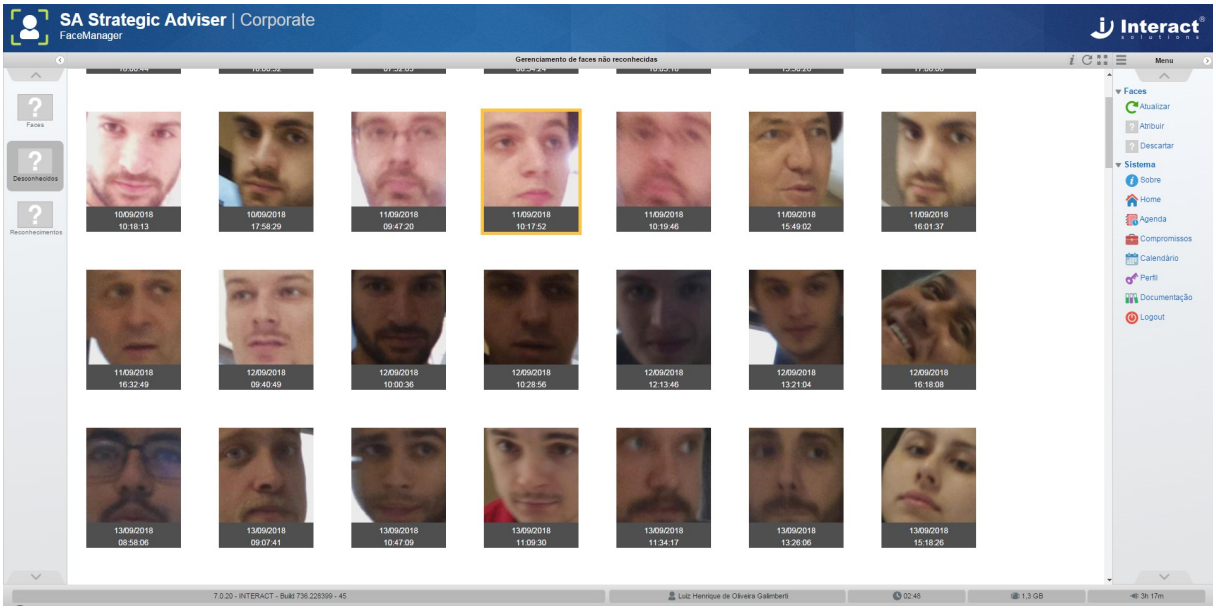


Fonte: Elaborado pelo autor (2018)

5.4.2 Faces desconhecidas

Nesta tela são exibidas todas as faces que o sistema não conseguiu reconhecer, assim como a data e o horário em que a tentativa de reconhecimento ocorreu. Estas faces podem ser atribuídas aos usuários cadastrados ou descartadas. A tela pode ser vista na Figura 16.

Figura 16: Faces desconhecidas

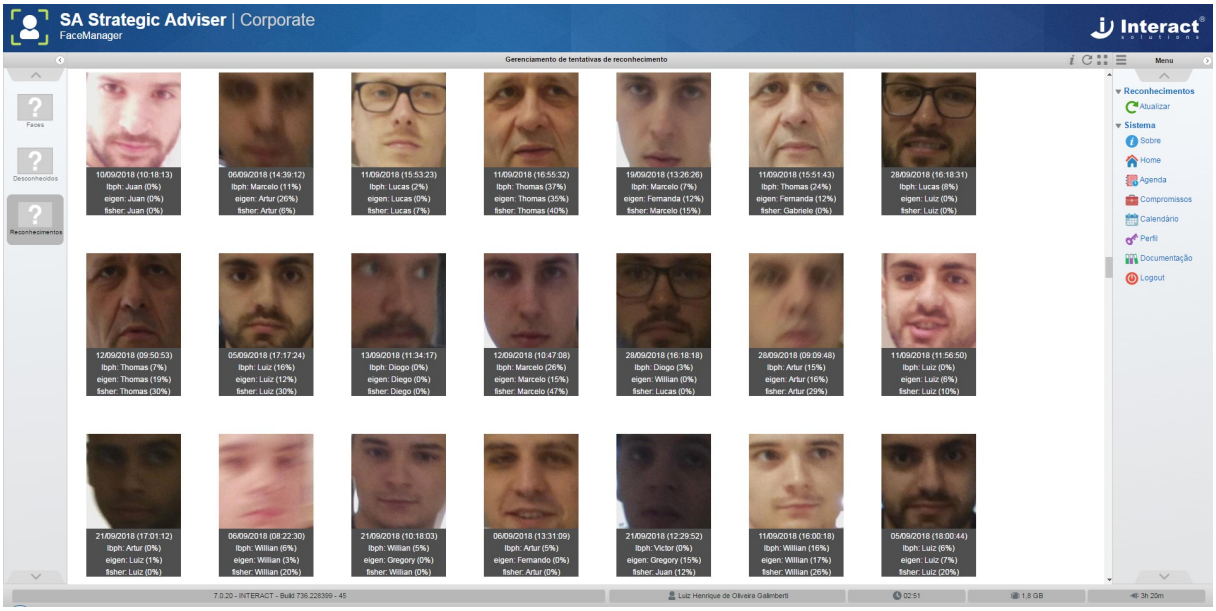


Fonte: Elaborado pelo autor (2018)

5.4.3 Tentativas de reconhecimento

Todas as tentativas de reconhecimento, sejam elas efetivas ou não, são listadas nesta tela, que pode ser vista na Figura 17. Abaixo de cada imagem são exibidas as informações de data e horário em que a tentativa de reconhecimento ocorreu, além dos percentuais de diferença em relação ao *threshold* de cada algoritmo.

Figura 17: Tentativas de Reconhecimento



Fonte: Elaborado pelo autor (2018)

6 RESULTADOS

Neste capítulo serão apresentadas as comparações dos resultados de cada algoritmo analisado, obtidos através do protótipo desenvolvido e descrito no capítulo anterior. A comparação foi feita utilizando as métricas apresentadas no Referencial Teórico. O valor de rigorosidade utilizado foi de 20% para que se tenha um baixo número de falsos positivos, que é o esperado em um sistema de controle de acesso. Foram analisadas quatro amostras diferentes, cada uma com um propósito. Além disso, foi feito um comparativo de velocidade dos algoritmos, um teste de segurança e uma verificação do desempenho geral do protótipo.

6.1 Amostra 1

Nesta amostra foram analisadas 105 tentativas de reconhecimento, obtidas em horários diversos e ao longo de duas semanas. Desta forma, a amostra aborda imagens em diferentes luminosidades e condições ambientais. A amostra também abrange indivíduos com diferentes quantidades de imagens cadastradas. Os resultados obtidos são exibidos na Tabela 2.

Tabela 2: Resultados da Amostra 1

	LPBH	Eigenfaces	Fisherfaces
Verdadeiros Positivos	8	4	10
Verdadeiros Negativos	42	45	45
Falsos Positivos	4	1	0
Falsos Negativos	51	55	50
Taxa de Falsa Rejeição	86,00%	93,00%	83,00%
Taxa de Falsa Aceitação	8,00%	2,00%	0,00%

Fonte: Elaborado pelo autor (2018)

É possível observar que o algoritmo Fisherfaces obteve melhores resultados, porém todos os algoritmos tiveram uma taxa de falsa rejeição incrivelmente alta. Um dos motivos para tal é a alta rigorosidade do sistema, requisito essencial para manter o número de falsos positivos próximo a zero. Outro motivo que justifica tais números será melhor visto na amostra seguinte.

6.2 Amostra 2

Este conjunto contempla 121 tentativas de reconhecimento de um único indivíduo, com 16 imagens de faces cadastradas. Desta forma é possível avaliar os resultados no melhor cenário possível. Esta amostra, assim como a anterior, aborda imagens em diferentes luminosidades e condições ambientais. Foram obtidos os seguintes resultados:

Tabela 3: Resultados da Amostra 2

	LPBH	Eigenfaces	Fisherfaces
Verdadeiros Positivos	22	18	104
Falsos Negativos	99	103	17
Taxa de Falsa Rejeição	81,81%	85,12%	14,04%

Fonte: Elaborado pelo autor (2018)

Novamente observa-se que o algoritmo Fisherfaces obteve melhores resultados em comparação aos demais. Comparando com a amostra anterior, nota-se que a quantidade de imagens cadastradas tem grande influência na taxa de acerto dos algoritmos.

Baseando-se na amostra 2, para aumentar a efetividade dos reconhecimentos, foi

proposta a utilização do algoritmo Fisherfaces como algoritmo principal e os demais algoritmos como secundários, seguindo as seguintes regras:

- Se o algoritmo principal obter um percentual de diferença do *threshold* menor que o valor de rigorosidade, os algoritmos secundários serão consultados.
- Se ambos os algoritmos secundários reconhecerem o mesmo indivíduo que o algoritmo principal, e a soma dos percentuais de diferença do *threshold* dos algoritmos secundários for maior que o valor de rigorosidade, o reconhecimento será tratado como Verdadeiro Positivo.

Seguindo a proposta acima, os resultados da amostra 2, conforme Tabela 4, seriam:

Tabela 4: Utilização dos três algoritmos em conjunto

	Resultados
Verdadeiros Positivos	108
Falsos Negativos	13
Taxa de Falsa Rejeição	10,74%

Fonte: Elaborado pelo autor (2018)

Como pode ser observado, utilizando o algoritmo com os melhores resultados, Fisherfaces, como algoritmo principal e os demais como algoritmos secundários, obtém-se um aumento no número de verdadeiros positivos e uma redução de quase 4% na taxa de falsa rejeição.

6.3 Amostra 3

A terceira amostra limita-se a um único indivíduo, utilizando óculos tanto nas imagens cadastradas quanto nas imagens captadas para reconhecimento. Foram analisadas 13 tentativas de reconhecimento. A Tabela 5 apresenta os resultados obtidos.

Tabela 5: Resultados da Amostra 3

	LPBH	Eigenfaces	Fisherfaces
Verdadeiros Positivos	6	2	8
Falsos Negativos	7	11	5
Taxa de Falsa Rejeição	53,84%	84,61%	38,46%

Fonte: Elaborado pelo autor (2018)

Como nas demais amostras, observa-se que o algoritmo Fisherfaces obteve resultados superiores. Também é possível verificar que todos os algoritmos analisados são capazes de reconhecer um indivíduo utilizando óculos.

6.4 Amostra 4

Para a 4ª amostra analisada foram feitas oito tentativas de reconhecimento de um único indivíduo. Tanto as imagens capturadas para reconhecimento quanto as imagens cadastradas para o usuário são mistas entre imagens do indivíduo utilizando óculos e imagens do indivíduo sem óculos. Os resultados obtidos são observados na Tabela 6.

Tabela 6: Resultados da Amostra 4

	LPBH	Eigenfaces	Fisherfaces
Verdadeiros Positivos	1	0	0
Falsos Negativos	7	8	8
Taxa de Falsa Rejeição	87,50%	100,00%	100,00%

Fonte: Elaborado pelo autor (2018)

Através destes resultados é possível verificar que os algoritmos têm dificuldades em reconhecer indivíduos quando suas fotos cadastradas possuem variação em sua aparência. Com exceção do algoritmo LBPH que faz a comparação das imagens individualmente, os demais algoritmos extraem as principais características da face presentes em todas as imagens cadastradas para determinado usuário. Este processo acaba sendo prejudicado em cenários como o desta amostra. Para tais casos, a melhor alternativa seria utilizar dois cadastros para este mesmo indivíduo. Um deles contendo apenas imagens com a utilização do óculos e o outro apenas imagens sem a utilização do óculos.

6.5 Comparativo de velocidade

Para a comparação do tempo necessário para efetuar a tentativa de reconhecimento foram selecionados o maior e o menor tempo registrados dentre todos os registros de tentativas de reconhecimento para cada algoritmo analisado. Os resultados obtidos, em milissegundos, foram os seguintes:

Tabela 7: Tempo para reconhecimento

	LPBH	**Eigenfaces**	**Fisherfaces**
Menor tempo (ms)	41	62	5
Maior tempo (ms)	65	176	23

Fonte: Elaborado pelo autor (2018)

Verifica-se que o algoritmo Fisherfaces é notavelmente mais rápido que os demais. É possível verificar também que embora o algoritmo LBPH faça a comparação de imagem a imagem, o mesmo obteve um desempenho superior ao algoritmo Eigenfaces.

6.6 Teste de segurança

Este teste objetivou verificar com que facilidade é possível burlar o sistema de controle de acesso utilizando imagens de faces, sejam elas impressas ou a partir da tela de um *smartphone*.

Ao posicionar uma imagem da face de um usuário cadastrado no sistema, apenas o algoritmo Fisherfaces reconheceu o indivíduo. Após adicionar esta mesma imagem no banco de imagens do usuário e repetir o teste, o algoritmo LBPH também passou a reconhecer o indivíduo.

Posicionando um crachá com a foto de um usuário cadastrado no sistema em frente a câmera do protótipo, mesmo gerando uma imagem de face desfocada, a mesma foi reconhecida pelo algoritmo Fisherfaces. Após adicionar esta mesma imagem no banco de imagens, o algoritmo Fisherfaces permaneceu como o único a reconhecer o indivíduo.

A partir destes testes verificou-se que o sistema de controle de acesso desenvolvido neste trabalho está sujeito a falhas de segurança de forma relativamente fácil utilizando-se de

imagens dos usuários cadastrados no sistema.

6.7 Desempenho geral do protótipo

O desempenho geral do protótipo pode ser observado na Tabela 8.

Tabela 8: Tempo para cada tarefa do protótipo (em milissegundos)

Captura de *frame*	~350ms
Processamento do *frame*	~250ms
Reconhecimento	~150ms

Fonte: Elaborado pelo autor (2018)

Considerando que a captura e o processamento dos *frames* é feito de forma paralela, ou seja, enquanto um *frame* está sendo processado um novo *frame* já está sendo capturado, o sistema consegue completar aproximadamente dois ciclos por segundo. Presumindo um tempo de exposição do indivíduo de dois segundos, será possível aproximadamente quatro tentativas de reconhecimento, sendo este um desempenho satisfatório considerando que assim haverá uma margem para imagens borradas ou desfocadas.

7 CONCLUSÃO

Com o objetivo de comparar os algoritmos de reconhecimento facial da biblioteca OpenCV, o presente trabalho apresentou o desenvolvimento de um protótipo de sistema de controle de acesso baseado em reconhecimento facial.

A partir dos resultados obtidos através do protótipo, foi possível comparar os algoritmos e constatar que o algoritmo Fisherfaces tem desempenho superior tanto na taxa de reconhecimentos quanto em velocidade. Além disso, constatou-se que a taxa de reconhecimento do sistema pode ser elevada ao utilizar os demais algoritmos em conjunto ao Fisherfaces e que para obter-se um desempenho satisfatório se faz necessário o cadastro de várias faces para um mesmo usuário.

A monografia abordou também o comportamento dos algoritmos em relação a utilização de óculos. Foi constatado que embora os algoritmos não tenham problemas em reconhecer indivíduos utilizando óculos, é necessário que sejam utilizados dois cadastros para um mesmo indivíduo caso o mesmo não utilize o óculos sempre.

As principais dificuldades encontradas no desenvolvimento do protótipo foram relacionadas ao desempenho e velocidade das detecções de faces, sendo necessário utilizar diversas combinações de resoluções e parâmetros de detecção até encontrar uma configuração que atingisse o desempenho esperado.

Propõe-se, como trabalhos futuros, o aperfeiçoamento da segurança de um sistema de controle de acesso como o desenvolvido neste trabalho, a fim de prevenir que o sistema possa ser burlado utilizando-se de imagens de faces dos usuários cadastrados no sistema.

REFERÊNCIAS

ABINADER, Jorge Abílio; LINS, Rafael Dueire. **Web Services em Java**. Brasport, 2006.

AGARWAL, M.; AGRAWAL, H.; JAIN, N.; KUMAR, M. **Face Recognition Using Principle Component Analysis, Eigenface and Neural Network**. Signal Acquisition and Processing, 2010.

BOECHAT, G. C. **Investigação de um Modelo de Arquitetura Biometrica Multimodal para Identificação Pessoal**. Universidade Federal do Pernambuco, 2008.

BOLLE, R. M., CONNELL, J. H., and RATHA, N. K. **Biometric perils and patches**. Elsevier Science, vol. 35, 2002.

BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. O'Reilly Media, 2008.

CAELUM. **Java e Orientação a Objetos**. CAELUM, 2008.

CARNEIRO, Larissa Natália das Virgens. **Reconhecimento de Face Invariante a Iluminação baseado em uma Abordagem Supervisionada**. Universidade Federal de Ouro Preto, 2012.

CAVALCANTI, G. D. C. **Sistemas Biométricos - Composição de Biometrias para Sistemas Multimodais de Verificação de Identidade Pessoal**. Universidade Federal de

Pernambuco, 2005.

COSTA, L. R.; OBELHEIRO, R. R.; FRAGA, J. S. **Introdução à Biometria**. Livro-texto de Minicursos - VI SBSeg, 2006.

CROTTI, Y.; DA SILVA, J. B.; MARCELINO, R.; VILSON, G.; CASAGRANDE, L. C. S. **Raspberry Pi e Experimentação Remota**. ICBL, 2013.

CURBERA, Francisco; DUFTLER, Matthew; KHALAF, Rania; NAGY, William; MUKHI, Nirmal; WEERAWARANA, Sanjiva. **Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI**. IEEE, 2002.

DE OLIVEIRA, Allisson Dantas. **Malaria System: uma ferramenta paradiagnóstico automático de malária emdispositivos móveis**. UFRPE, 2014.

DINIZ, Fábio Abrantes; NETO, Francisco Milton Mendes; JÚNIOR, Francisco das Chagas Lima; FONTES, Laysa Mabel de O. **RedFace: Um Sistema de Reconhecimento Facial Baseado em Técnicas de Análise de Componentes Principais e Autofaces**. Revista Brasileira de Computação Aplicada, 2011.

DO VAL, F. B. R.; MARCELINO, P.R.; NETO, J. J. **Uso de Técnicas Adaptativas no Reconhecimento Biométrico por Impressão Digital**. IX Workshop de Tecnologia Adaptativa, 2015.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification, Second Edition**. Wiley-Interscience, 2000.

FUJIKAWA, Cesar Shuji. **Reconhecimento Facial utilizando Descritores de Textura e Aprendizado Não Supervisionado**. UNESP, 2016.

HONG, Lin.; JAIN, Anil. **Integrating faces and fingerprints for personal identification**. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 12, 1998.

IZO, Flávio. P**rotótipo de um Sistema Web para Reconhecimento Facial utilizando a**

Metodologia Eigenfaces. UCAM, 2015.

JAIN, Anil; BOOLE, Ruud; PANKANTI, Sharath. **Biometrics: Personal identification in networked society**. Springer, 2005.

JAIN, Anil; HONG, Lin; PANKANTI, Sharath. **Biometric Identification**. Communications of the ACM, 2000.

JAIN, A.; ROSS, A.; PRABHAKR, S. **Human recognition using biometrics: an overview**. Annals of Telecommunications, v. 62, 2007.

JESUS, Leone; GUIMARÃES, Deivite; SAPUCAIA, Flávio; PIMENTEL, Fagner; DE SOUZA, Josemar Rodrigues; SIMÕES, Marco A. C.; FRIAS, Diego. **Analise de Métodos de Processamento de Imagens para Reconhecimento Facial utilizando Fisherfaces em Imagens sob Condições Desfavoráveis**. UNEB, 2015.

KÖRTING, Thales Sehn; FILH, Nelson Lopes Duarte. **Utilizando Eigenfaces para Reconhecimento de Imagens**. Fundação Universidade Federal do Rio Grande, 2004.

KSHIRSAGAR, V. P.; BAVISKAR, M. R.; GAIKWAD, M. E. **Face recognition using Eigenfaces**. Computer Research and Development, 2011.

LIU, S.; Silverman, M. **A Practical Guide to Biometric Security Technology**. IEEE Computer Society, 2001.

LÓPEZ. Laura Sánchez. **Local Binary Patterns applied to Face Detection and Recognition**. Universidade Politécnica da Catalunha, 2010.

MAGALHÃES, Paulo Sérgio; SANTOS, Henrique Dinis. **Biometria e autenticação**. Actas da 4ª Conferência da Associação Portuguesa de Sistemas de Informação, 2003.

MARENGONI, Maurício; STRINGHINI, Denise. **Tutorial: Introdução à Visão Computacional usando OpenCV**. RITA, 2009.

MILLER, Benjamin. **Vital signs of identity - Special Report: Biometrics**. IEEE Spectrum, 1994.

MORAES, A. F. **Método para Avaliação da Tecnologia Biométrica na Segurança de Aeroportos**. Universidade de São Paulo, 2006.

NARANG, Sudha; JAIN, Kriti; SAXENA, Megha; ARORA, Aashna. **Comparison of Face Recognition Algorithms Using Opencv for Attendance System**. International Journal of Scientific and Research Publications, 2018.

NASCIMENTO, Vitor. **Implementação de um Sistema de Identificação Facial utilizando Linux Embarcado**. USP, 2015.

NSTC. **Biometrics History**. Disponível em <https://www.hsdl.org/?view&did=463907>. Acesso em: 12 mai. 2018.

OKABE, Rogerio Kazuhiro; CARRO, Silvio Antonio. **Reconhecimento Facial em Imagens capturadas por Câmeras Digitais de Rede**. Colloquium Exactarum, 2014.

OPENCV. **Opencv**. Disponível em <https://opencv.org/>. Acesso em: 06 mai. 2018.

PAPAGEORGIOU, C. P.; POGGIO, Tomaso A.; OREN, Michael. **General framework for object detection**. Computer Vision, 1998.

PAPILOSCOPIA. **História**. Disponível em <http://www.papiloscopia.com.br/historia.html>. Acesso em: 06 mai. 2018.

PASQUALI, Luiz; ARAUJO, Marcos Elias Claudio de. **Datiloscopia - A determinação dos dedos**. LABPAM, 2006.

PENG, K.; CHEN, L.; RUAN, S.; KUKHAREV, G. **A robust algorithm for yes detection on gray intensity face without spectacles**. Journal of Computer Science and Technology, vol. 5, 2005.

PENTEADO, Bruno Elias; MARANA, Aparecido Nilceu. **Autenticação Biométrica On-Line de Usuários em Aplicações Web de Ensino a Distância**. UNESP, 2008.

PERRONNIN, Florent; JUNQUA, Jean-Claude; DUGELAY, Jean-Luc. **Biometrics Person Authentication: From Theory to Practice**. EURASIP, 2005.

PHILLIPS, J.; SYED, R.; HYEONJOON, M.; RAUSS, P. **The Feret evaluation methodology for face recognition algorithms**. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2000.

PRODOSSIMO, F. C.; CHIDAMBARAM, C.; LOPES, H. S. **Otimização da Detecção de Olhos em Imagens Faciais utilizando os Algoritmos Colônia de Abelhas Artificiais e Harmony Search**. Anais do X Congresso Brasileiro de Inteligência Computacional, 2011.

RASPBERRYPI. **Raspberry Pi**. Disponível em <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>. Acesso em: 26 mai. 2018.

ROWLEY, H. **Neural network-based face detection**. IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 20, no. 1, 1998.

SAKAI, T.; NAGAO, M.; KANADE, T. **Computer analysis and classification of photographs of human faces**. First USA-Japan Computer Conference, 1972.

SAMPAIO, Cleuton. **SOA e WebServices em Java**. Brasport, 2006.

SANDMANN, Humberto; SENAGA, Marcelo. **Bastet – Sistema de reconhecimento Facial**. Centro Universitário UniFEI, 2002.

SANTOS, A. L. dos. **Gerenciamento de identidades: Segurança da informação. Rio de Janeiro**. Brasport, 2007.

SIROVICH, L.; KIRBY, M. **Low-dimensional procedure for the characterization of human faces**. Optical Society of America, 1987.

SOPCHUK, Augusto Fernando; AGNER, Willian Ricardo Fialka; TAFURI, Leandro. **Uso da Tecnologia Java no desenvolvimento de um Software para Controle de Produção**. Revista Científica Semana Acadêmica, 2014.

SOUZA, Marcelo Barboza. **Controle de Acesso: Conceitos, Tecnologias e Benefícios**. Editora Sicurezza, 2010.

TAN, X. **Face recognition from a single image per person: A survey**. Pattern Recognition, Elsevier Science Inc., n. 9, 2006.

TOLBA, A. S.; EL-BAZ, A. H.; EL-HARBY, A. A. **Face recognition: a literature review, International Journal of Signal Processing**. vol. 11, no. 4, 2006.

VETTER, Ron. **Authentication by biometric verification**. Computer, vol. 43 no. 2, 2010.

VIEIRALVES, Evandro Luiz de Xerez; FILHO, Cícero Ferreira Fernandes Costa. **Avaliação do Desempenho de Sistemas de Reconhecimento de Impressões Digitais**. FUCAPI, NUTELI, 2013

VIOLA, P.; JONES, M. **Robust real-time object detection**. Cambridge Research Laboratory/Compaq Computer Corporation, 2001.

VUJOVIĆ, Vladimir; MAKSIMOVIĆ, Mirjana. **Raspberry Pi as a Wireless Sensor Node: Performances and Constraints**. MIPRO, 2014.

YANG, Ming-Hsuan; KRIEGMAN, David; AHUJA, Narendra. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. IEEE Computer Society, vol. 24, n. 1, 2002.

ZHAO, W.; CHELLAPPA, R.; PHILLIPS, J.; ROSENFELD, A. **Face recognition: a literature survey**. ACM Computing Surveys, vol. 35, no. 4, 2003.