



PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE ENGENHARIA / FACULDADE DE INFORMÁTICA



ENGENHARIA DE COMPUTAÇÃO

AUTOR

MARCELO COSTA DA CONCEIÇÃO

BIOMETRIA PESSOAL UTILIZANDO TÉCNICAS PARA A DETECÇÃO E RECONHECIMENTO FACIAL

Trabalho de Conclusão do Curso de Engenharia de Computação

Porto Alegre, Junho de 2014

AUTOR
MARCELO COSTA DA CONCEIÇÃO

**BIOMETRIA PESSOAL UTILIZANDO TÉCNICAS PARA A DETECÇÃO
E RECONHECIMENTO FACIAL**

Trabalho de conclusão de curso apresentado
como parte das atividades para obtenção do
título de Engenheiro de Computação da
Faculdade de Engenharia da Pontifícia
Universidade Católica do Rio Grande do Sul.

ORIENTADOR
PROF. DR. DÊNIS FERNANDES

Porto Alegre, Junho de 2014

Dedico este trabalho ao meu filho e à
minha esposa, Jaqueline Coutinho.

AGRADECIMENTOS

Agradeço principalmente à minha esposa, que desde que nos conhecemos, sempre me incentivou a superar os desafios que surgem. Agradeço à minha mãe, que está sempre ao meu lado, durante toda a minha vida. Agradeço a minha tia, por ser minha segunda mãe e me ajudar em momentos difíceis.

Agradeço em especial ao Prof. Dr. Dênis Fernandes por ter sido meu orientador neste projeto, provendo ajuda e dedicação em todas as etapas do desenvolvimento desse trabalho.

Talk is cheap. Show me the code.

Linus Torvalds

RESUMO

Sistemas computacionais para reconhecimento facial são de difícil implementação, e muitos pesquisadores devotam anos neste campo. Este trabalho tem como objetivo realizar uma avaliação de técnicas comumente utilizadas para sistemas de processamento de imagens capaz de detectar faces a partir de uma câmera de vídeo acoplada a um computador e comparar com faces previamente armazenadas em uma base de dados.

O projeto do sistema consiste em três partes principais: detecção facial, extração das características da face e reconhecimento facial. A detecção de faces utiliza o método proposto por Viola e Jones, que é baseado em filtros de Haar em cascata, que tem como base o treinamento de classificadores, agregando diversos classificadores fracos, gerando um classificador forte. Para a extração de características, foi utilizada a análise de componentes principais, que reduz a dimensionalidade do espaço enquanto preserva o máximo possível de informação, e para o reconhecimento foi utilizada uma métrica baseada em distância entre projeções. A implementação dos métodos de visão computacional serão realizadas com auxílio da biblioteca de processamento de imagens OpenCV, em linguagem de programação C/C++.

Palavras-chave: Principal Component Analysis. Viola-Jones. Haar-Cascade. OpenCV. Reconhecimento Facial.

ABSTRACT

Computer systems for face recognition are difficult to implement and many researchers devote years in this field. This paper aims to conduct a review of the commonly used techniques for image processing capable of detecting faces from a video camera attached to a computer and compare it with faces previously stored in a database system.

The system design consists of three main parts: face detection, facial feature extraction and face recognition. The face detection using the method proposed by Viola and Jones, which is based on Haar filters in cascade, which is based on the training of classifiers, adding several weak classifiers, generating a strong classifier. For feature extraction, the principal components analysis, which reduces the dimensionality of the space was used while preserving as much as possible information, and recognition based on a metric distance between projections was used. The implementation of the methods of computer vision will be accomplished using the OpenCV image processing library in C / C + + programming language.

Keywords: Principal Component Analysis. Viola-Jones. Haar-Cascade. OpenCV. Facial Recognition.

LISTA DE FIGURAS

Figura 1: Arquitetura básica de um sistema de reconhecimento facial.....	17
Figura 2: Sistema de processamento de imagens.....	21
Figura 3: Imagens obtidas em condições diferentes: (a) imagem normal; (b) imagem obtida por sensor infravermelho; (c) com interferência de iluminação.....	22
Figura 4: Exemplo de redução de ruídos usando média de múltiplas imagens: (a) imagem com ruído; (b)-(f) resultados para $M = 2, 4, 8, 16$ e 32 imagens.....	23
Figura 5: Vizinhança 3×3 ao redor de um ponto de coordenadas (x,y)	25
Figura 6: Esquema de processamento de imagens no domínio da frequência.....	25
Figura 7: Exemplos de histogramas.....	27
Figura 8: Imagens correspondentes aos histogramas da figura 7	28
Figura 9: Exemplo de segmentação: (a) imagem em 256 tons de cinza; (b) imagem binária.....	30
Figura 10: Operadores: (a) máscara genérica com a posição dos valores utilizados nas equações de Prewitt e Sobel; (b) máscaras dos operadores de Prewitt, horizontal e vertical; (c) máscaras dos operadores de Sobel.....	32
Figura 11: Detector Marr-Hildreth: (a) superfície da Laplaciana de uma Gaussiana; (b) máscara 5×5 representando a superfície.....	34
Figura 12: Comparativo entre algoritmos de detecção de borda: (a) Sobel, (b) Prewitt, (c) Canny.....	35
Figura 13: Segmentação de imagem utilizando a técnica de threshold. (a) imagem em nível de cinza; (b) histograma com os pontos de corte; (c) imagem binarizada..	36
Figura 14: Estrutura de um reconhecedor de padrões.....	37
Figura 15: O modelo apresentado indica locais onde se esperam encontrar olhos, boca e pele.....	38
Figura 16: Classificação supervisionada.....	39
Figura 17: Classificação não supervisionada.....	39
Figura 18: Imagem Integral.....	50

Figura 19: Cálculo de imagem integral.....	50
Figura 20: Cálculo de um retângulo.....	51
Figura 21: Quatro tipos de características base utilizadas na abordagem proposta por Viola e Jones.....	52
Figura 22: Conjunto estendido de características proposto por Lienhart e Maydt.....	53
Figura 23: Características selecionadas pelo algoritmo AdaBoost.....	55
Figura 24: Cascata de classificadores.....	55
Figura 25: Varredura em busca de padrões baseado na diferença entre áreas claras e escuras de uma região.....	56
Figura 26: Etapas do algoritmo Eigenface.....	60
Figura 27: Conjunto de imagens de treino. Fonte: AT&T Facedatabase.....	61
Figura 28: Exemplo de face média (abaixo) e Eigenfaces mais importantes para o conjunto de treino.....	63
Figura 29: Sistema implementado. (a) identificação positiva em relação às faces treinadas; (b) identificação negativa em relação às faces treinadas.....	64
Figura 30: Imagens para treinamento, geradas automaticamente.....	65
Figura 31: Imagens de entrada. (a) imagem original em resolução 640x480 pixels; (b) imagem pré-processada.....	68
Figura 32: Comparação entre imagens: (a) imagem reconstruída; (b) imagem pré-processada.....	70

LISTA DE EQUAÇÕES

Equação 1.....	24
Equação 2.....	26
Equação 3.....	31
Equação 4.....	31
Equação 5.....	32
Equação 6.....	32
Equação 7.....	33
Equação 8.....	35
Equação 9.....	35
Equação 10.....	35
Equação 11.....	35
Equação 12.....	49
Equação 13.....	50
Equação 14.....	51
Equação 15.....	54
Equação 16.....	54
Equação 17.....	54
Equação 18.....	61
Equação 19.....	62
Equação 20.....	62
Equação 21.....	62
Equação 22.....	62
Equação 23.....	63
Equação 24.....	63
Equação 25.....	69

LISTA DE ABREVIATURAS E SIGLAS

TIC	Tecnologias de Informação e Comunicações
SI	Sistemas de Informação
OpenCV	Open Source Computer Vision Library
GUI	Graphical User Interface
PCA	Principal Component Analysis
ACP	Análise de componentes Principais

LISTA DE TABELAS

Tabela 1: Algoritmos e comandos para reconhecimento de padrões no openCV.....	42
Tabela 2: Principais funções da biblioteca OpenCV.....	44
Tabela 3: Classificadores pré-treinados.....	67
Tabela 4: Código do erro L2: Raíz quadrada da soma do quadrado do erro.....	70

SUMÁRIO

1. INTRODUÇÃO.....	15
1.1 OBJETIVOS.....	17
1.2 MOTIVAÇÃO.....	18
1.3 ORGANIZAÇÃO.....	19
2 REFERENCIAL TEÓRICO.....	20
2.1 PROCESSAMENTO DE IMAGEM DIGITAL.....	20
2.1.1 Ruído.....	22
2.1.2 Filtragem no Domínio Espacial.....	24
2.1.3 Filtragem no Domínio da Frequência.....	25
2.1.4 Histograma.....	26
2.2 VISÃO COMPUTACIONAL.....	29
2.2.1 Segmentação.....	29
2.2.1.1 Segmentação por detecção de borda.....	31
2.2.1.2 Segmentação por Threshold.....	36
2.3 RECONHECIMENTO DE PADRÕES.....	37
2.3.1 Classificadores.....	38
2.3.1.1 Classificação Supervisionada.....	39
2.3.1.1.1 Conjunto de treinamento.....	40
2.3.1.1.2 Conjunto de características.....	40
2.3.1.1.3 Classificador.....	41
2.3.1.2 Etapas da classificação supervisionada.....	43
2.4 BIBLIOTECA OPENCV.....	44
2.4.1 Porquê a OpenCV?.....	46
3. MÉTODOS DE DETECÇÃO DE FACES.....	47

3.1 MÉTODO DE DETECÇÃO DE FACES VIOLA-JONES.....	48
3.1.1 Imagem Integral.....	49
3.2 OUTRAS ABORDAGENS PARA DETECÇÃO FACIAL.....	57
3.2.1 Métodos baseados em conhecimento humano e morfologia.....	57
3.2.2 Métodos baseados em casamento de padrões.....	58
4. MÉTODOS PARA RECONHECIMENTO DE FACES.....	59
4.1 ANÁLISE DE COMPONENTES PRINCIPAIS (PCA).....	59
4.1.1 Conjunto de faces.....	61
4.1.2 Localização da face média.....	62
4.1.3 Cálculo da matriz de covariância.....	62
4.1.4 Redução da dimensionalidade da matriz.....	63
5 METODOLOGIA.....	64
5.1 BANCO DE IMAGENS FACIAIS.....	65
5.2 TREINAMENTO A PARTIR DAS FACES SALVAS NO BANCO DE IMAGENS	66
5.3 DETECÇÃO DE FACES.....	67
5.4 PRÉ-PROCESSAMENTO DA IMAGEM DE ENTRADA.....	68
5.5 RECONHECIMENTO E IDENTIFICAÇÃO DA FACE.....	69
6 CONSIDERAÇÕES FINAIS.....	71
6.1 TRABALHOS FUTUROS.....	72
7. REFERÊNCIAS BIBLIOGRÁFICAS.....	73
APÊNDICE A – FLUXOGRAMA FUNCIONAL DO SISTEMA IMPLEMENTADO.....	76
APÊNDICE B – EXEMPLO DE ESTRUTURA DE FACES PARA TREINAMENTO	77
APÊNDICE C – EXEMPLO DE ESTRUTURA DE EIGENFACES GERADA.....	78

1. INTRODUÇÃO

Biometria [bio (vida) + metria (medida)] é o estudo estatístico das características físicas ou comportamentais dos seres humanos, como forma de identificá-los unicamente. O princípio básico desta técnica para identificação é: “*seu corpo, sua senha*”. As técnicas de reconhecimento por meio das características das pessoas já eram utilizadas na China no ano 800 DC, quando comerciantes confirmavam a identidade de seus clientes por meio da impressão de suas digitais em tábuas de barro.

O problema de estabelecer uma associação entre um indivíduo e uma identidade pode ser dividido em duas categorias: autenticação e identificação. Autenticação refere-se ao problema de confirmar ou negar uma alegada identidade de um indivíduo, enquanto identificação refere-se ao problema de estabelecer a identidade, desconhecida à partida, de um indivíduo [THIAN, 2001].

Com o aumento do poder computacional, novas técnicas mais eficientes começaram a ser empregadas no reconhecimento de indivíduos. Estas técnicas valem-se da definição de biometria e da alta capacidade de processamento dos computadores atuais para, por exemplo, permitir controle de acesso de um indivíduo a lugares restritos, a algum sistema a informações sigilosas, ou para outros fins como a identificação criminal, controle de ponto e comparação de informações.

Os atuais sistemas biométricos podem basear o seu funcionamento em características de diversas partes do corpo humano, por exemplo: os olhos, a palma da mão, as digitais do dedo, a retina ou íris dos olhos, a face, etc. A fundamentação é a de que cada indivíduo é único e possui características físicas e de comportamento (a voz, a maneira de andar, etc.) distintas, traços aos quais são característicos de cada ser humano, com isso, dispomos de uma maneira bastante

prática de reconhecimento e identificação de indivíduos, dificultando possíveis fraudes.

Com a utilização cada vez maior de Tecnologias de Informação e Comunicações (TIC) nos Sistemas de Informação (SI) das organizações, surgem com crescente evidência os problemas de segurança e, em particular, a questão da autenticação de usuários. Esta questão é hoje fundamental já que o acesso indevido a informação sensível pode provocar grandes prejuízos à organização. Neste trabalho descreve-se uma das técnicas utilizadas na autenticação, a biometria facial, como forma de aumentar a qualidade da autenticação. A necessidade de aumentar, de forma significativa, a segurança de acesso a sistemas e lugares, permitindo uma identificação pessoal não invasiva.

Embora o reconhecimento de faces seja uma tarefa simples para o ser humano, é extremamente complexo implementar esse processo em uma máquina, pois não sabemos, ao certo, como o cérebro humano realiza essa tarefa. O cérebro humano pode identificar corretamente uma pessoa, a partir de sua imagem facial, mesmo sobre as mais diversas condições, como variações de iluminação, observando apenas uma de suas características ou partes, e até mesmo com distorções ou deformações.

Um dos principais fatores que distinguem o reconhecimento facial dos outros sistemas biométricos reside no fato desta tecnologia poder ser utilizada para fins de vigilância. A identificação de criminosos procurados ou a localização de crianças desaparecidas, são apenas algumas das aplicações possíveis. Outras vantagens desta tecnologia incluem: A capacidade de capturar faces em locais públicos (o que pode minimizar as limitações jurídicas) e o fato dessa captura poder ser feita à distância (sem contacto físico), além da possibilidade de integração com os diversos sistemas já existentes.

1.1 OBJETIVOS

O reconhecimento biométrico facial tem sido um tema largamente difundido na área de visão computacional, e este trabalho, tem como base diversos artigos, estudos e livros sobre este assunto. Esta proposta baseia-se no desenvolvimento de um protótipo de processamento de imagens capaz de detectar faces, comparar com faces já armazenadas e armazenar informações em arquivo das faces reconhecidas.

A arquitetura básica do protótipo, consiste em receber como entrada o código correspondente a um dos usuários cadastrados (matrícula, código de barras, etc.), buscar e fazer o treinamento das imagens do respectivo usuário e identificar se a pessoa que está na frente da webcam corresponde ao código do usuário informado. Caso seja o usuário, mostra na tela o seu nome, fala uma mensagem via voz sintetizada, salva uma foto e escreve em um arquivo de relatório, o nome, a data e a hora. Caso não seja, somente mostra na tela a mensagem “NAO CADASTRADO”.

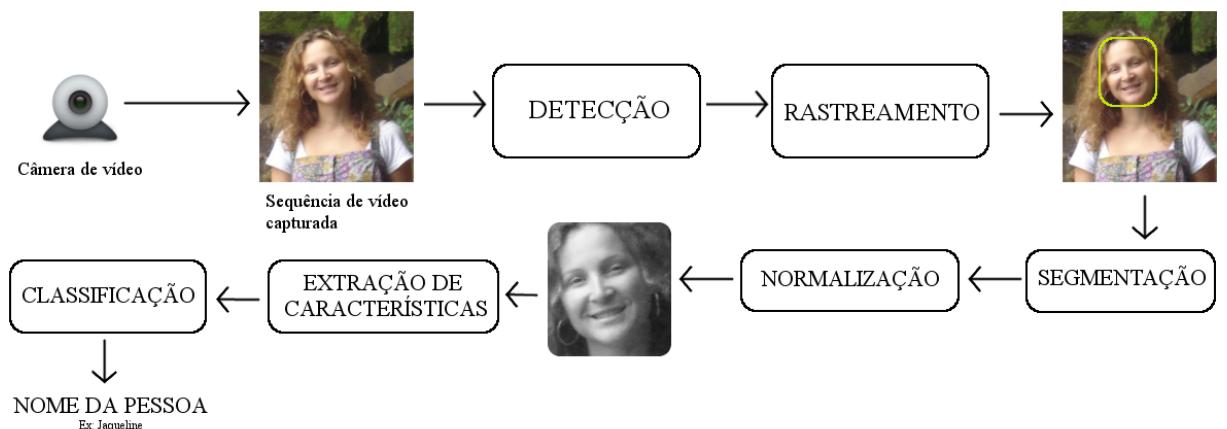


Figura 1: Arquitetura básica de um sistema de reconhecimento facial.

A detecção de faces utilizará o método proposto por Viola e Jones [VIOLA; JONES, 2001], baseado em filtros de Haar em cascata [WILSON; FERNANDES, 2006] e para o reconhecimento, escolhemos como modelo de representação da informação facial o método de Eigenfaces [LATA et al., 2009], que é baseado no

modelo PCA (Principal Component Analysis). Para a voz sintetizada, utilizamos o software opencource **eSpeak** (eSspeak.sourceforge.net).

A implementação dos métodos de visão computacional são realizadas com auxílio da biblioteca OpenCV [BAGGIO et al., 2012] em linguagem de programação C/C++.

1.2 MOTIVAÇÃO

Com a necessidade de proteger o crescente volume de informações, controlar o acesso a locais e sistemas, identificar criminosos, localizar pessoas desaparecidas, entre outras, com uma precisão aceitável, é que se procura no desenvolvimento de sistemas de segurança confiáveis. Isso é possível unindo a esses sistemas modalidades de biometria que, em geral, são menos suscetíveis a falhas. Baseado neste cenário, as pesquisas no campo de visão computacional, que permitam identificar métodos que auxiliem o reconhecimento automático e eficiente de faces humanas, tem grande valor tanto científico como prático.

1.3 ORGANIZAÇÃO

1. Introdução: Apresenta um histórico sobre biometria e reconhecimento de faces, bem como exemplos de aplicações, vantagens, técnicas e tecnologias envolvidas no processo.
2. Referencial Teórico: Nesta etapa, são apresentadas a teoria que envolve os métodos escolhidos para o processo de reconhecimento facial apresentado.
3. Métodos para Detecção de Faces: Apresenta a importância dos métodos para detecção de faces e descreve com detalhes o método Viola-Jones, que é utilizado neste trabalho.
4. Métodos para Reconhecimento de Faces: Descreve o método para reconhecimento de faces apresentado neste trabalho (PCA).
5. Implementação do Sistema: Descreve como a implementação do sistema será conduzida, tendo como objetivo final um sistema de reconhecimento facial automatizado.
6. Considerações Finais: Conclusões e trabalhos futuros.

2 REFERENCIAL TEÓRICO

Este capítulo aborda os termos necessários para compreensão desse trabalho que envolve as áreas de pesquisa de Processamento de Imagens Digitais e Visão Computacional. Na seção de Processamento de Imagens Digitais, serão apresentadas algumas das principais técnicas que envolvem o processamento de imagens digitais. A seção Visão Computacional, em que serão mostradas técnicas de segmentação que permitem identificar objetos em imagens em que serão apresentados alguns dos principais métodos para Reconhecimento de Padrões e Classificadores. Para finalizar o capítulo, serão descritos os principais itens do trabalho: A biblioteca OpenCV, a técnica de detecção de faces proposta por Viola e Jones e a técnica de reconhecimento facial usando Eigenfaces.

2.1 PROCESSAMENTO DE IMAGEM DIGITAL

O processamento digital de sinais consiste na análise dos sinais analógicos utilizando ferramentas matemáticas, podendo assim realizar transformações ou extrair informações desses sinais. Os sinais analógicos (mundo real), abrangem uma infinidade de itens, como som, luz, pressão, temperatura, entre outros. Para trabalhar com esses sinais na forma digital, é necessário que haja uma transformação, utilizando conversores analógico-digitais para transformar estas informações analógicas em sequências binárias.

Devido ao avanço das técnicas utilizadas, a evolução e a crescente redução do custo dos componentes eletrônicos, o processamento de sinais digitais está disseminado em diversas áreas do conhecimento como a física, química, biologia, astronomia, entre outras, como o processamento digital de imagens nas áreas de medicina e biometria.

O processamento de imagem digital consiste de um conjunto de operações matemáticas (algoritmos) aplicados sobre a imagem para gerar uma nova imagem, representação ou descrição de conteúdo (Figura 2). Essas transformações são necessárias em diversos segmentos, tais como, computação gráfica, visualização científica, recuperação de imagens por conteúdo e visão computacional.

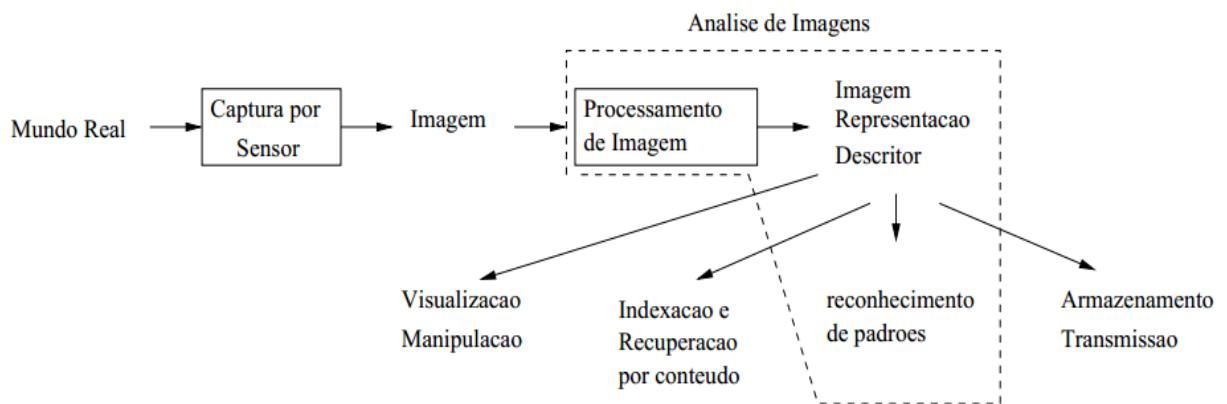


Figura 2: Sistema de processamento de imagens

Nesta pesquisa, o processamento de imagem digital será uma etapa anterior a etapa de visão computacional, isso se deve ao fato de que as imagens de onde queremos extrair alguma informação podem, em alguns casos, precisarem ser convertidas para um determinado formato ou tamanho, e ainda ser filtradas para remover ruídos provenientes do processo de aquisição da imagem. Os ruídos podem aparecer de diversas fontes, como por exemplo, qualidade do sensor utilizado, iluminação do ambiente, obstáculos, condições climáticas no momento da aquisição da imagem, entre outros.

A Figura 3 apresenta imagens em condições diferentes para exemplificar estes tipos de interferência.



Figura 3: Imagens obtidas em condições diferentes: (a) imagem normal; (b) imagem obtida por sensor infravermelho; (c) com interferência de iluminação.

2.1.1 Ruído

Ruído é o equivalente digital dos grãos dos filmes utilizados em câmeras analógicas. Para as imagens digitais, esses ruídos aparecem, normalmente, como manchas aleatórias em uma superfície originalmente suave e pode degradar a qualidade da imagem significativamente. O ruído aumenta proporcionalmente à sensibilidade escolhida, duração da exposição e temperatura e sempre há algum grau de ruído presente em qualquer aparelho eletrônico que transmite ou recebe sinais, mesmo sendo inevitável, o ruído pode se tornar tão pequeno relativamente ao sinal que pode ser considerado inexistente.

A razão entre o sinal e o ruído (SNR, do inglês “signal to noise ratio”) é uma maneira útil e universal de comparar as quantidades relativas de sinal e ruído para qualquer sistema eletrônico; razões altas terão pouco ruído visível enquanto o oposto vale para baixas razões. Lembrando que ruído não é apenas interferência no sinal de captura da imagem, mas também interferências que possam atrapalhar a interpretação ou o reconhecimento de objetos. E os filtros são as ferramentas básicas para remoção de ruídos em imagens. A Figura 4 mostra um exemplo de redução de ruídos, usando a técnica da média de múltiplas imagens.

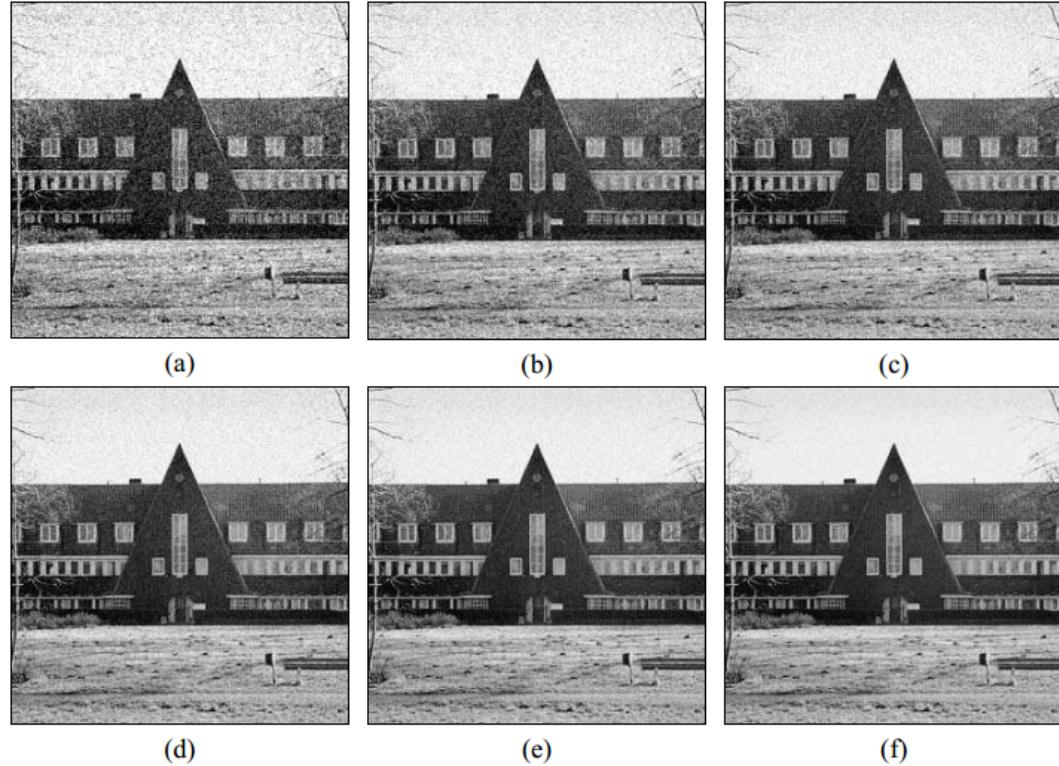


Figura 4: Exemplo de redução de ruídos usando média de múltiplas imagens: (a) imagem com ruído; (b)-(f) resultados para $M = 2, 4, 8, 16$ e 32 imagens

Pode-se entender que a filtragem de uma imagem, são técnicas de transformações, aplicadas a cada pixel da imagem, levando em conta os níveis de cinza de uma região vizinha de cada pixel desta imagem. Os filtros podem ser espaciais (filtros que atuam diretamente na imagem) ou de frequência, onde a imagem é inicialmente convertida para o domínio de frequência usando a Transformada de Fourier Discreta (DFT), normalmente em conjunto com algoritmos da Transformada Rápida de Fourier (FFT).

2.1.2 Filtragem no Domínio Espacial

As técnicas de filtragem no domínio espacial são aquelas que atuam diretamente sobre a matriz de pixels que é a imagem digitalizada, ou seja, se refere à imagem em si. Logo, as funções de processamento de imagens no domínio espacial podem ser expressas pela Equação 1.

$$g(x,y) = T[f(x,y)]$$

Equação 1

Onde: $g(x,y)$ é a imagem processada, $f(x,y)$ é a imagem original e T é um operador em f , definido em uma certa vizinhança de (x,y) . Além disso, o operador T pode também operar sobre um conjunto de imagens de entrada. A vizinhança normalmente definida ao redor de (x,y) , é a 8-vizinhança do pixel de referência, o que equivale a uma região 3×3 na qual o pixel central é o de referência, como indica a Figura 5. O centro dessa região é movido pixel a pixel, iniciando no canto superior esquerdo da figura e aplicando a cada localidade o operador T para calcular o valor de g naquele ponto.

A filtragem no domínio espacial, baseia-se na utilização de máscaras e essas máscaras são pequenas matrizes bidimensionais e os valores de seus coeficientes determinam o objetivo a ser alcançado durante o processamento [GONZALES; WOODS,1992].

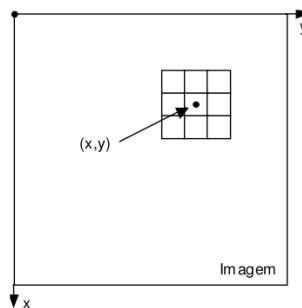


Figura 5: Vizinhança 3x3
ao redor de um ponto de
coordenadas (x,y) .

2.1.3 Filtragem no Domínio da Frequênciā

A filtragem no domínio da frequência tem suas técnicas fundamentadas no teorema da convolução, que é uma das propriedades da transformada de Fourier, portanto, o processamento de imagens digitais no domínio da frequência é realizado, usualmente através de três passos conforme mostrado na Figura 6.

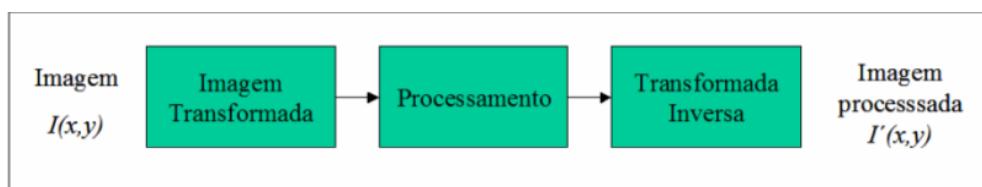


Figura 6: Esquema de processamento de imagens no domínio da frequência.

- A imagem é transformada do domínio espacial para o da frequência, usando a transformada de Fourier;
- Operações são realizadas;
- Para que a imagem possa ser exibida, ocorre o processo inverso, onde a imagem que está no domínio da frequência passa para o domínio espacial. Esta etapa é realizada através da transformada inversa de Fourier.

2.1.4 Histograma

Segundo Ogê Filho e Hugo Neto [FILHO; NETO, 1999], o histograma de uma imagem é simplesmente 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).

Cada elemento deste conjunto é calculado como:

$$P_r(r_k) = \frac{n_k}{n}$$

Equação 2

onde:

$$0 \leq r_k \leq 1$$

$k = 0, 1, \dots, L - 1$, onde L é o número de níveis de cinza da imagem digitalizada;

n = número total de pixels na imagem;

$P_r(r_k)$ = probabilidade do k -ésimo nível de cinza;

n_k = número de pixels cujo nível de cinza corresponde a k .

Um histograma apresenta várias características importantes. A primeira delas é que cada $P_r(r_k)$ fornece, como sugere a notação, a probabilidade de um pixel da imagem apresentar nível de cinza r_k , portanto, um histograma nada mais é que uma função de distribuição de probabilidades e como tal deve obedecer aos axiomas e teoremas da teoria de probabilidade.

A Figura 7 apresenta cinco exemplos de tipos de histogramas freqüentemente encontrados em imagens. O histograma da Figura 7(a) apresenta grande concentração de pixels nos valores mais baixos de cinza, correspondendo a uma imagem predominantemente escura. Na Figura 7(b), os pixels estão concentrados em valores próximos ao limite superior da escala de cinza, caracterizando uma imagem clara. Na parte (c) da figura, os pixels estão agrupados em torno de valores intermediários de cinza, correspondendo a uma imagem de brilho médio. Nas figuras (a), (b) e (c) a maioria dos pixels está concentrada em uma estreita faixa da escala de cinza, significando que as imagens correspondentes apresentam baixo contraste. A Figura 7(d) corresponde a uma imagem com pixels distribuídos ao longo de toda a escala de cinza. É comum dizer que uma imagem com estas características apresenta um bom contraste. A Figura 7(e) mostra um histograma tipicamente bimodal, isto é, apresentando duas concentrações de pixels, uma delas em torno de valores escuros e outra na região clara do histograma.

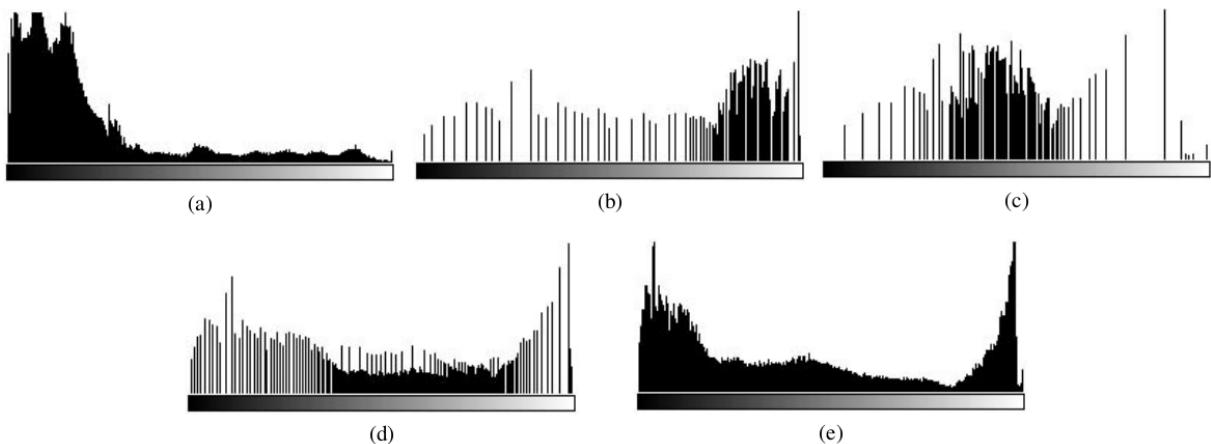


Figura 7: Exemplos de histogramas

Pode-se dizer que a imagem correspondente apresenta alto contraste entre as duas concentrações, uma vez que elas se encontram razoavelmente espaçadas. Para verificar a relação entre imagens e respectivos histogramas, a Figura 8 mostra cinco imagens monocromáticas cujos correspondentes histogramas são aqueles da Figura 7.

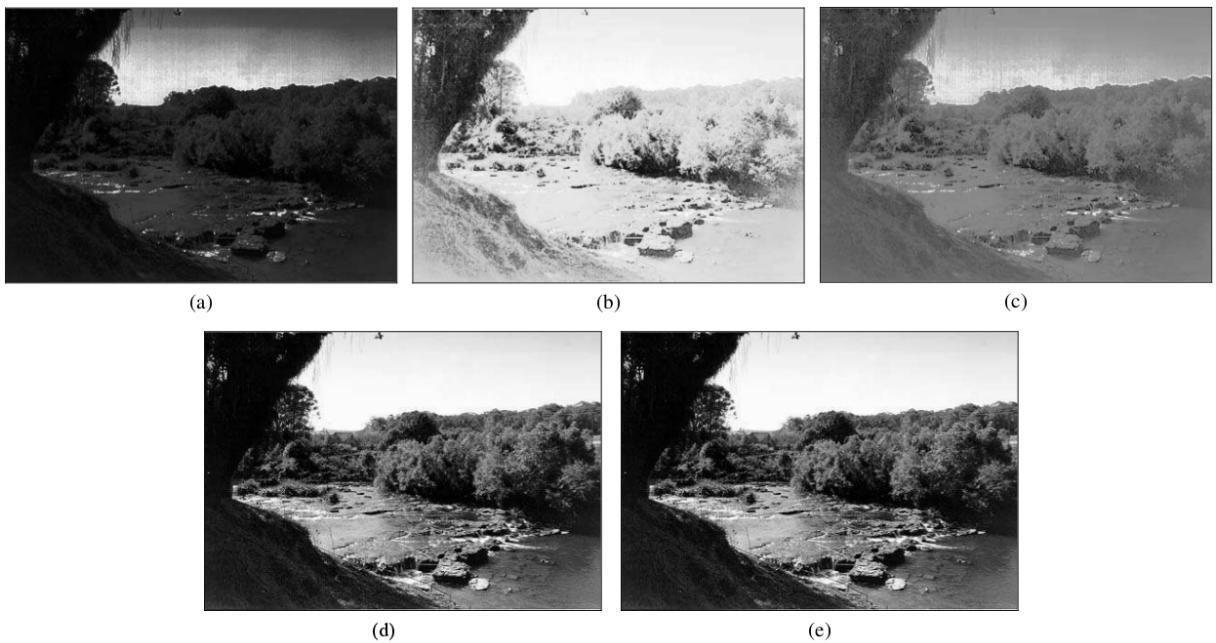


Figura 8: Imagens correspondentes aos histogramas da figura 7

Para computar o histograma de uma imagem monocromática, inicializa-se com zero, todos os elementos de um vetor de L elementos, onde L é o número de tons de cinza possíveis. Em seguida, percorre-se a imagem, pixel a pixel, e incrementa-se a posição do vetor cujo índice corresponde ao tom de cinza do pixel visitado. Após toda a imagem ter sido percorrida, cada elemento do vetor conterá o número de pixels cujo tom de cinza equivale ao índice do elemento.

Concluindo esta Seção, convém enfatizar que, embora o histograma de uma imagem forneça diversas informações qualitativas e quantitativas sobre ela (ex: predominância de pixels claros ou escuros, etc.), outras conclusões de caráter qualitativo (ex: qualidade da imagem, presença ou não de ruído etc.) somente podem ser extraídas dispondendo-se da imagem propriamente dita. Tal fato pode ser confirmado a partir de uma análise das Figuras 7 e 8. O conceito de histograma também é aplicável a imagens coloridas. Neste caso, a imagem é decomposta em seus componentes (por exemplo, componentes RGB) e para cada componente é calculado o histograma correspondente.

2.2 VISÃO COMPUTACIONAL

Visão computacional é a ciência responsável pela visão de uma máquina, pela forma como um computador enxerga o meio à sua volta, extraíndo informações significativas a partir de imagens capturadas por câmeras de vídeo, sensores, entre outros dispositivos. Estas informações permitem reconhecer, manipular e pensar sobre os objetos que compõem uma imagem, portanto, a visão computacional procura emular a visão humana e possui como entrada uma imagem e a saída é uma interpretação da imagem como um todo, ou parcialmente [ROSENFELD; KAK, 1982].

Como visto anteriormente, no processamento de imagens ficou caracterizado a eliminação de ruídos e melhoria no contraste das imagens. Neste item começamos a migrar para os processos mais relacionados com a visão computacional. Veremos inicialmente um processo de segmentação, para em seguida analisar processos mais cognitivos, como o reconhecimento de padrões.

2.2.1 Segmentação

Em visão computacional, segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões (conjunto de pixels) ou objetos, com o objetivo de simplificar e/ou mudar a representação de uma imagem para facilitar a sua análise. Segmentação de imagens é tipicamente usada para localizar objetos e formas (linhas, curvas, etc) em imagens. Este processo é geralmente guiado por características do objeto ou região, como por exemplo, cor ou proximidade.

O nível de detalhamento em um processo de segmentação depende da tarefa a ser executada e da resolução da imagem que se tem, por exemplo, se procuramos por uma casa em imagens que foram obtidas do nível da rua, estamos procurando

regiões que ocupam uma boa porcentagem da imagem (regiões grandes), porém, se procuramos por casas a partir de imagens de satélite, estamos procurando regiões pequenas. Embora a tarefa seja a mesma, a resolução das imagens é diferente e o tratamento utilizado no processo de segmentação pode ser diferente também.

O resultado da segmentação de imagens é um conjunto de regiões/objetos ou um conjunto de contornos extraídos da imagem, como resultado, cada um dos pixels em uma mesma região é similar com referência a alguma característica ou propriedade computacional, tais como cor, intensidade, textura ou continuidade. A Figura 9 mostra um exemplo bem simples de segmentação. Uma imagem de ferro fundido (Figura 9-a), em 256 tons de cinza, é segmentada, gerando uma imagem binária (Figura 9-b), onde as partículas de grafita (regiões de pixels brancos) distinguem-se da matriz (fundo preto).

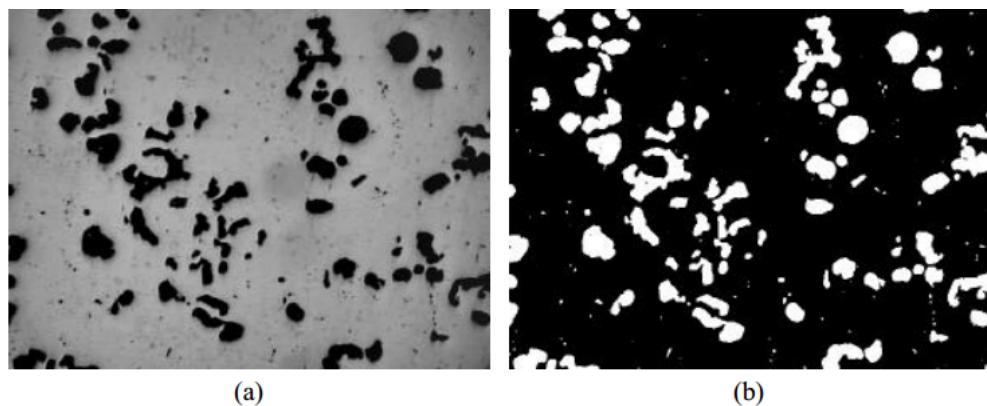


Figura 9: Exemplo de segmentação: (a) imagem em 256 tons de cinza; (b) imagem binária.

A segmentação costuma ser uma etapa crítica, pois é através dela que se reconhece e se identifica os objetos de interesse, sobre os quais será feita a análise. Existem diversas técnicas que podem ser utilizadas para a segmentação e nosso objetivo não é esgotar estas técnicas, mas apresentar algumas das mais utilizadas para reconhecimento de padrões.

As técnicas de segmentação que serão apresentadas neste trabalho, podem ser classificadas em dois grupos:

- Segmentação por detecção de borda;
- Segmentação por threshold (corte);

2.2.1.1 Segmentação por detecção de borda

Uma borda em uma imagem é caracterizada por uma mudança, normalmente abrupta, no nível de intensidade dos pixels. Os detectores de borda são definidos para encontrar este tipo de variação nos pixels e quando estes pixels estão próximos, eles podem ser conectados formando uma borda ou um contorno e assim definindo uma região ou objeto. Variações nos níveis de intensidade dos pixels podem ser determinadas pelas derivadas primeira e/ou derivada segunda.

Alguns métodos de determinação de borda utilizam estas técnicas. O processo consiste em definir as máscaras que caracterizem estas variações e em seguida fazer a convolução da imagem pela máscara. Os operadores mais comuns, são os operadores de Prewitt [PREWITT, 1970] e de Sobel [SOBEL; FELDMAN, 1968]. As expressões matemáticas que definem o operador de Prewitt, são apresentadas nas Equações 3 e 4.

$$g(x) = \frac{\partial f(x, y)}{\partial(x)} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

Equação 3

$$g(y) = \frac{\partial f(x, y)}{\partial(y)} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

Equação 4

As expressões matemáticas que definem o operador de Sobel, são apresentadas nas Equações 5 e 6.

$$g(x) = \frac{\partial f(x, y)}{\partial(x)} = (z_7 + 2*z_8 + z_9) - (z_1 + 2*z_2 + z_3)$$

Equação 5

$$g(y) = \frac{\partial f(x, y)}{\partial(y)} = (z_3 + 2*z_6 + z_9) - (z_1 + 2*z_4 + z_7)$$

Equação 6

Note que o operador de Sobel é semelhante ao de Prewitt sendo a única diferença o valor 2 que multiplica o termo central nas linhas da máscara. Este fator tende a suavizar o resultado do operador, atenuando ruídos. A Figura 10 mostra as máscaras relacionadas a cada um destes operadores para detecção de bordas verticais e horizontais. Para determinarmos bordas nas diagonais basta rotacionar a máscara do ângulo desejado.

Z_1	Z_2	Z_3
Z_4	Z_5	Z_6
Z_7	Z_8	Z_9

(a)

-1	-1	-1
0	0	0
1	1	1

(b) Prewitt

-1	0	1
-1	0	1
-1	0	1

-1	-2	-1
0	0	0
1	2	1

(c) Sobel

-1	0	1
-2	0	2
-1	0	1

Figura 10: Operadores: (a) máscara genérica com a posição dos valores utilizados nas equações de Prewitt e Sobel; (b) máscaras dos operadores de Prewitt, horizontal e vertical; (c) máscaras dos operadores de Sobel.

É possível ainda definir operadores que atuam utilizando a derivada segunda dos pixels (chamados de operadores laplacionais), estes operadores são baseados no fato que quando existe uma borda, a derivada segunda atravessa o eixo das abcissas (este efeito é conhecido como zero-crossing). Um dos primeiros algoritmos detectores de borda utilizando este conceito foi o de Marr-Hildreth [MARR; HILDRETH, 1980], que foi baseado nos seguintes conceitos:

- A mudança de intensidade não é independente da escala da imagem, logo os operadores devem ter tamanhos diferentes.
- A mudança de intensidade de forma abrupta faz com que a derivada segunda atravesses o eixo das abcissas.

O operador Laplaciano é simétrico, uma característica da curva gaussiana, o que evita o uso de operadores múltiplos e tem um efeito de suavização da imagem. O tamanho da máscara deve ser definido de acordo com o valor de β (*desvio padrão de probabilidades associadas*), normalmente este valor é um número ímpar maior que $6*\beta$, uma vez que 99,7% do volume de uma Gaussiana está definido entre $\pm 3\beta$. Note que os coeficientes da máscara somam zero, isto garante que em locais onde a imagem é homogênea não é acrescentado valor algum na imagem resultante. Marr e Hildreth definiram o operador Laplaciano de uma função Gaussiana conforme a Equação 7.

$$\nabla^2 G(x, y) = \left(\frac{x^2 + y^2 + 2\beta^2}{\beta^4} \right) * e^{-\frac{x^2+y^2}{2\beta^2}}$$

Equação 7

O formato da curva e a máscara 5x5 que representa a função são apresentados na Figura 11.

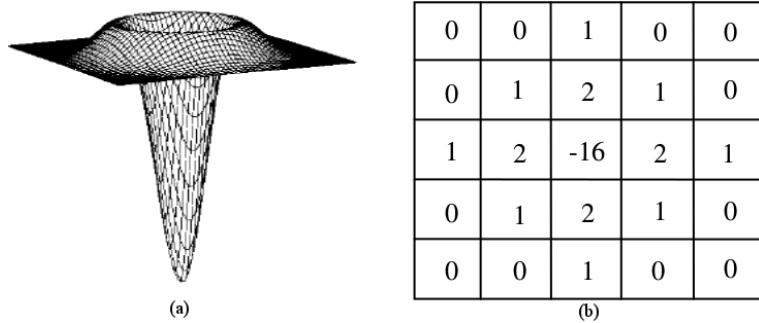


Figura 11: Detector Marr-Hildreth: (a) superfície da Laplaciana de uma Gaussiana; (b) máscara 5x5 representando a superfície.

Para finalizar as técnicas de detecção de borda, apresentamos a técnica desenvolvida por John Canny [CANNY, 1986], embora seja a mais complexa, é também a de melhor desempenho entre as técnicas discutidas até aqui. O detector de bordas de Canny utiliza um algoritmo multi-estágios para detectar uma ampla margem de bordas na imagem e possui três objetivos básicos:

- Boa Detecção: O algoritmo deve ser capaz de identificar todas as bordas possíveis na imagem;
- Boa Localização: As bordas encontradas devem estar o mais próximo possível das bordas da imagem original;
- Resposta Mínima: Cada borda da imagem deve ser marcada apenas uma vez. O ruído da imagem não deve criar falsas bordas.

Para satisfazer tais condições, Canny utilizou um cálculo de variações, visando encontrar uma função que otimizasse o funcional desejado, definindo um detector de borda baseado na primeira derivada de uma gaussiana. Numa primeira etapa a imagem é convoluida por uma função gaussiana, em seguida são determinados a magnitude e a direção, conforme indicado pelas expressões nas equações 8, 9, 10 e 11.

Os valores de gi podem ser obtidos a partir dos operadores de Prewitt ou de Sobel. $M(x,y)$ é uma imagem que contém a informação da magnitude em cada pixel e $\alpha(x, y)$ é uma imagem que contém a direção da normal à borda para cada pixel. A ideia é, para cada pixel p , verifica se pelo menos um dos vizinhos de p possui a mesma direção que p , se sim, marca o pixel com $M(x,y)$ (as coordenadas de p), senão marca a imagem de saída com 0.

$$g(x, y) = f(x, y) \text{conv} G(x, y) \quad M(x, y) = \sqrt{g_x^2 + g_y^2}$$

Equação 8

Equação 9

$$\alpha(x, y) = \tan^{-1}\left(\frac{g_y}{g_x}\right) \quad gi = \frac{\partial g(x, y)}{\partial i}$$

Equação 10

Equação 11

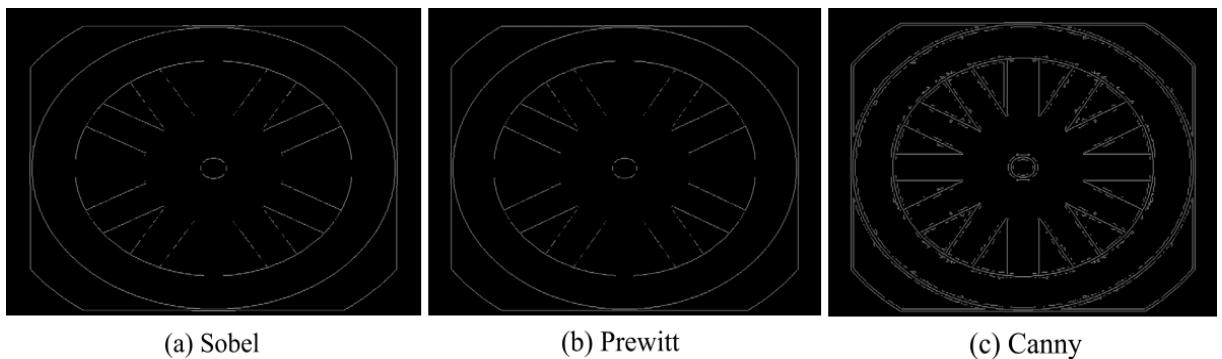


Figura 12: Comparativo entre algoritmos de detecção de borda: (a) Sobel, (b) Prewitt, (c) Canny

2.2.1.2 Segmentação por Threshold

A segmentação de imagens por Threshold é uma das mais simples de ser implementada, é rápida em termos computacionais e utiliza de propriedades intuitivas para criar a imagem segmentada. A segmentação por threshold partitiona uma imagem diretamente em regiões, baseado simplesmente nos valores de intensidade e/ou propriedades destes valores. A ideia central é verificar no histograma da imagem quantas regiões existem (picos e vales) e segmentar a imagem baseado nesta informação. Como se pode verificar na Figura 13, temos um histograma de dois picos. O primeiro deles é o mais escuro e corresponde às células. O segundo pico (maior) corresponde ao fundo, que é maior pois o fundo é o elemento predominante da imagem.

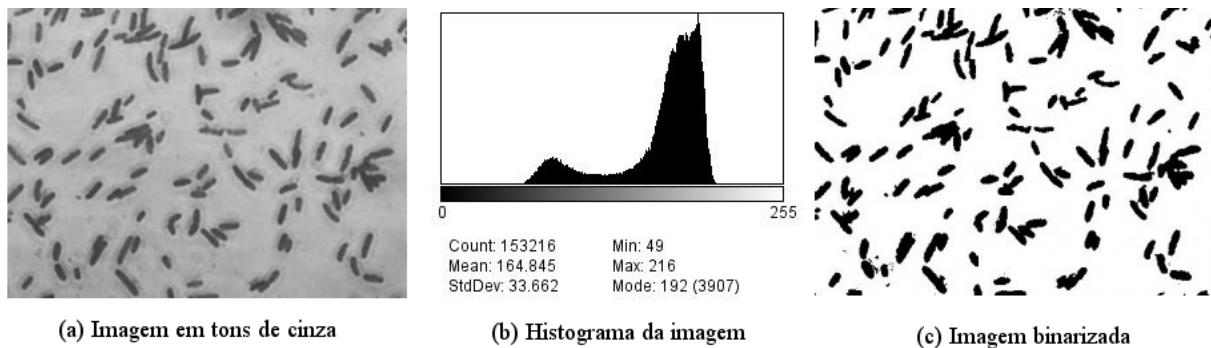


Figura 13: Segmentação de imagem utilizando a técnica de threshold. (a) imagem em nível de cinza; (b) histograma com os pontos de corte; (c) imagem binarizada.

O que o threshold faz é dividir a imagem por um ponto de corte no histograma, normalmente chamado de nível de threshold. Depois de determinado o nível do threshold, o programa divide a imagem em preto (pixels com valores abaixo do nível) e branco (tudo o que tiver um valor acima do nível estabelecido). O resultado é uma imagem em que os softwares de processamento de imagens podem separar o fundo dos objetos de interesse, para numa etapa posterior rotular e medir os objetos.

2.3 RECONHECIMENTO DE PADRÕES

Basicamente, o reconhecimento de padrões é a área de pesquisa que tem por objetivo a classificação de objetos (padrões) em um número de categorias ou classes. Segundo Jorge Marques [MARQUES; 2005], é mais simples escolher dentre um conjunto limitado de hipóteses, qual delas se adapta melhor às observações feitas do que olhar para um objeto genérico e identificá-lo. Este problema de decisão pode ser resolvido por métodos matemáticos se for suficientemente definido.

A estrutura clássica de um sistema de reconhecimento de padrões constitui-se por dois blocos, um de extração de características, chamadas de padrões e um classificador. O primeiro bloco seleciona através dos sensores apenas as informações mais relevantes para a decisão, chamadas de características. O classificador utiliza as características para escolher a hipótese ou classe que melhor soluciona o problema definido. A Figura 14 apresenta a estrutura de um reconhecedor de padrões.

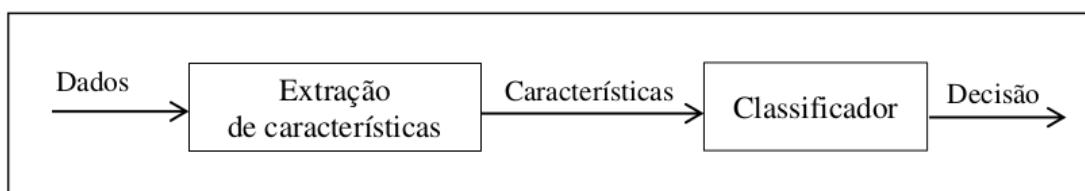


Figura 14: Estrutura de um reconhecedor de padrões.

Um objeto pode ser definido por mais de um padrão (textura, forma, cor, dimensões, etc) e o reconhecimento individual destes padrões pode facilitar o reconhecimento do objeto como um todo. O reconhecimento de padrões pode ainda ser uma mistura de técnicas, por exemplo, no processo de reconhecimento de faces apresentado em [CÂNDIDO, MARENCONI; 2009], é utilizado um modelo estrutural para determinar o local mais provável para se encontrar partes de uma face (boca, olhos e pele), conforme apresentado na Figura 15.

Cada uma destas partes pode ser reconhecida utilizando diferentes técnicas, 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.

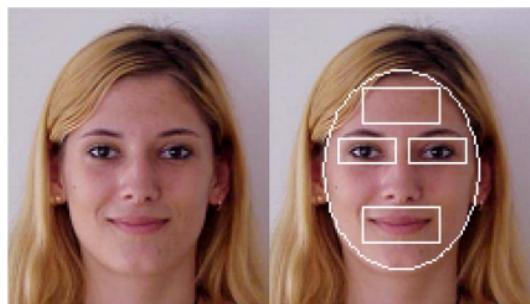


Figura 15: O modelo apresentado indica locais onde se esperam encontrar olhos, boca e pele.

2.3.1 Classificadores

Os classificadores podem ser utilizados para classificar os objetos em uma imagem. A partir de seus respetivos atributos, constrói-se um espaço de características, onde cada objeto é representado por um vetor, formado por seus atributos, que assume a forma de um ponto neste espaço. Então, a classificação dos objetos é feita através da análise do posicionamento [GOMES, 2001].

Existem dois tipos de classificação: supervisionada e não-supervisionada. Na classificação supervisionada (Figura 16), as classes são definidas por padrões conhecidos, de modo que um objeto é atribuído a uma classe de acordo com sua proximidade aos padrões conhecidos de cada classe no espaço de características. Na classificação não-supervisionada (Figura 17), não há nenhuma informação sobre as classes e a classificação agrupa os pixels segundo suas características espectrais, organizando-os em agrupamentos denominados clusters.

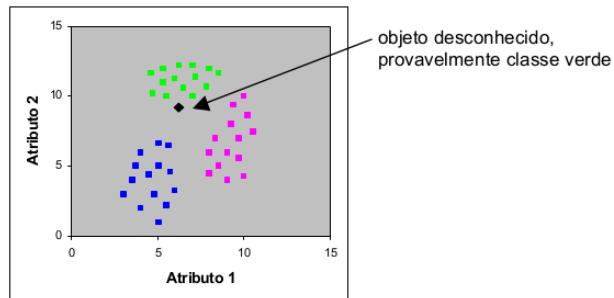


Figura 16: Classificação supervisionada.

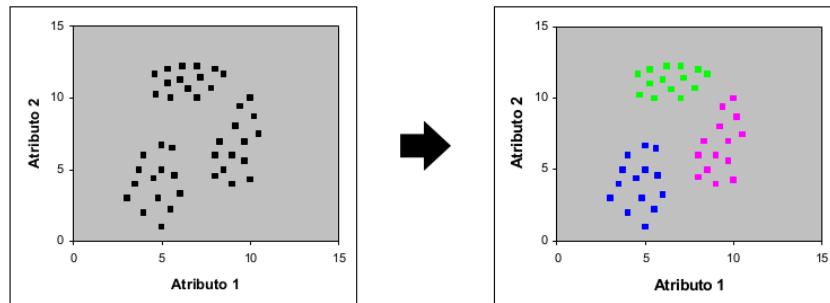


Figura 17: Classificação não supervisionada.

Este trabalho apresentará somente a classificação supervisionada.

2.3.1.1 Classificação Supervisionada

A classificação supervisionada atribui uma classe a um objeto desconhecido através de uma medida de similaridade de suas características com as características de objetos padrões conhecidos. Assim, classificação supervisionada demanda:

- Um conjunto de treinamento;
- Um conjunto de características;
- Um classificador.

2.3.1.1.1 Conjunto de treinamento

O conjunto de treinamento é um conjunto constituído por objetos padrões conhecidos e rotulados (previamente classificados). Ele deve ter o tamanho suficiente para ter uma razoável estatística, minimizando o efeito de ruídos, e deve apresentar grande variabilidade dentro de cada classe para representar bem as diversas classes.

2.3.1.1.2 Conjunto de características

O conjunto de características é o conjunto de atributos que define o espaço de características. Deve ser um conjunto de atributos que caracterize bem os objetos, agrupando os objetos similares e separando os distintos. No entanto, o conjunto de características precisa ser robusto para poder agrupar as diferenças dentro de cada classe e sensível o bastante para discriminar essas classes. Assim, os atributos do conjunto de características são escolhidos visando a separação das classes para possibilitar a classificação.

Na prática, raramente um único atributo é suficiente para distinguir duas ou mais classes, sendo geralmente utilizados vários atributos. Por outro lado, a adição de mais atributos ao conjunto de características não significa sempre uma melhora da classificação. O aumento da dimensionalidade do espaço de características torna-se necessário para se ter uma estatística de boa aceitação. Além disso, um grande conjunto de características pode reduzir a generalidade do sistema de classificação [RAUDYS; JAIN, 1991]. Deste modo, deve-se utilizar o menor conjunto de características possível, combinando e/ou selecionando atributos dentre os atributos disponíveis.

A técnica clássica de combinação de atributos é a análise de componentes principais (PCA), que constrói atributos a partir da combinação linear de outros que maximiza a variância entre as classes.

2.3.1.1.3 Classificador

O classificador é o algoritmo que mensura a similaridade entre os objetos e as classes. Existem dois tipos principais de classificador: estatístico (não-paramétrico e paramétrico) e conexionista (redes neurais). Os classificadores não-paramétricos são os mais simples, eles utilizam uma função de distância para determinar as fronteiras de decisão que dividem o espaço de características em sub-espacos, discriminando as classes [GOMES, 2001]. Os classificadores não-paramétricos mais comuns são os classificadores de Distância Euclidiana e de Mahalanobis.

O classificador de Distância Euclidiana determina as fronteiras de decisão a partir das distâncias euclidianas entre os centróides de cada classe no espaço de características. Devido à sua natureza radial, este classificador só apresenta bons resultados quando os clusters, que representam as classes, são convexos. Tal problema é minimizado com a utilização do classificador de Mahalanobis, pois este leva em consideração a forma dos clusters.

Os classificadores paramétricos discriminam as classes através da estimativa de parâmetros da distribuição dos pontos de cada classe no espaço de características, como a densidade de probabilidade (classificador de Bayes). As redes neurais são classificadores mais complexos, inspirados na estrutura neural de organismos inteligentes, que aprendem através da experiência.

É difícil encontrar técnicas estruturais prontas em bibliotecas, uma vez que estas técnicas dependem da estrutura de cada objeto, mas as técnicas baseadas em teoria de decisão são mais gerais e podem ser adaptadas a diferentes tipos de objetos. A biblioteca OpenCV [BRADSKI; KAEHLER, 2008] possui diversas técnicas nesta categoria, conforme apresentado na Tabela 1.

Algoritmo	Descrição	Comando
Mahalanobis	Uma medida de distância que verifica a elasticidade do espaço dos dados.	<i>cvMahalanobis</i>
K-médias	Um algoritmo de agrupamento não supervisionado.	<i>cvKMeans2</i>
Classificador Bayesiano	Um classificador que assume que as características são Gaussianas e estatisticamente independentes	<i>cvNormalBayesClassifier</i>
Árvore de Decisão	Um classificador discriminativo	<i>cvDTree</i>
Boosting	Classificador de grupo discriminativo. A classificação geral depende da combinação de pesos dada a cada classificador.	<i>cvBoost</i>
Árvores Randômicas	Um conjunto de árvores de decisão combinadas.	<i>cvRTrees</i>
Classificador Haar	Classificador baseado em boosting.	<i>cvHaarDetectObjects</i>
Maximização esperada	Agrupador não supervisionado baseado na técnica generativa.	---
K-vizinhos	O classificador discriminativo mais simples possível.	---
Redes Neurais	Baseado nos perceptrons de múltiplas camadas.	---
Support Vector Machine	Classificador discriminativo capaz de fazer regressões.	---

Tabela 1: Algoritmos e comandos para reconhecimento de padrões no openCV.

2.3.1.2 Etapas da classificação supervisionada

Um procedimento de classificação supervisionada consiste em três etapas: treinamento, validação e classificação. Treina-se o sistema com os padrões do conjunto de treinamento e faz-se sua validação através da classificação de um conjunto de objetos conhecidos [GOMES, 2001].

O treinamento prepara o sistema para a classificação. A partir do classificador e do conjunto de treinamento, são determinadas funções discriminantes, que são as fronteiras de decisão que dividem o espaço de características em sub-espacos, que representam, cada um, uma classe.

A validação mede a qualidade do sistema e verifica sua capacidade de generalização. É realizada através do cálculo da taxa de acerto da classificação de um conjunto de objetos conhecidos (conjunto de validação) diferentes dos objetos do conjunto de treinamento. No entanto, quando o número de objetos conhecidos disponíveis é pequeno, privilegia-se a formação do conjunto de treinamento, sendo a validação feita através da classificação do próprio conjunto de treinamento (auto-validação), mas neste caso, há apenas uma estimativa da qualidade do sistema.

2.4 BIBLIOTECA OPENCV

OpenCV (Open Source Computer Vision Library) [BRADSKI; KAEHLER, 2008]. Originalmente, desenvolvida pela Intel, em 2000, é uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de Visão computacional. A Tabela 2 apresenta uma descrição mais detalhada de algumas das principais funções suportadas pela OpenCV.

Função	Descrição
Manipulação de imagens.	Alocação, desalocação, cópia, modificação, conversão.
I/O de vídeo e imagem.	Input e output baseado em arquivo e câmera.
Manipulação de matrizes e vetores e retinas de álgebra linear.	Produto, solução, <i>single value decomposition</i> (SVD).
Diversas estruturas de dados dinâmicas.	Listas, filas, pilhas, árvores, grafos.
Processamento de imagens básico.	Filtragem, detecção de bordas, detecção de cantos, interpolação e amostragem, conversão de cor, operações morfológicas, histogramas, pirâmides de imagens.
Análise estrutural.	Componentes conexos, processamento de contornos, transformação de distância, momento diverso, casamento de templates, aproximação poligonal, montagem de linha, montagem de elipse, triangularização de Delaunay.
Calibração de câmera.	Busca, rastreamento e calibração de padrões, calibração, estimativa de matrizes fundamentais, estimativa homográfica, correspondência estéreo.
Análise de movimentos.	Fluxo óptico, segmentação de movimentos, rastreamento.
Reconhecimento de objetos.	Métodos-eigen, Hidden Markov Models (HMM).
GUI básica.	Exibição de vídeo e imagens, tratamento de entrada do teclado e do mouse, barras de rolagens.
Rotulamento (Labeling) de imagens.	Linha, cônica, polígono, texto, desenho de texto.

Tabela 2: Principais funções da biblioteca OpenCV.

A OpenCV possui módulos de Processamento de Imagens e Vídeo I/O, Estrutura de dados, Álgebra Linear, GUI (Interface Gráfica do Usuário) Básica com sistema de janelas independentes, Controle de mouse e teclado, além de mais de 350 algoritmos de Visão computacional como: Filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros. Conta também com processamento de imagens em tempo real.

A biblioteca OpenCV foi desenvolvida nas linguagens de programação C/C++, e permite suporte a programadores que utilizam Java, Python e Visual Basic e desejam incorporar a biblioteca a seus aplicativos. Possui uma estrutura modular, o que significa que inclui várias bibliotecas compartilhadas ou estáticas. A seguir são apresentados seus principais módulos:

- **core** - Módulo compacto com as definições básicas usadas por todos os outros módulos.
- **imgproc** - Módulo de processamento de imagens, que inclui filtragem de imagens, transformações geométricas em imagens, conversão de cores, histogramas, etc.
- **video** - Módulo de análise de vídeo, que inclui estimativa de movimento, subtração de fundo, e algoritmos de rastreamento de objeto.
- **calib3d** – Multi-visualizador básico de algoritmos geométricos, calibração de câmera simples e estéreo, estimativa de pose do objeto, algoritmos de correspondência estéreo, e elementos da reconstrução 3D.
- **features2d** - Detectores de características salientes, descritores e descriptor por correspondências.

- **objdetect** – Detecção de objetos e instância de classe pré definidas(faces, olhos, pessoas, carros, entre outros).
- **highgui** - Interface para captura de vídeo, imagem e codecs de vídeo, bem como recursos de interface com o usuário.
- **gpu** - Algoritmos de aceleração por GPU (Graphics Processing Unit) para diferentes módulos da OpenCV.

2.4.1 Porquê a OpenCV?

A principal razão que motivou tal escolha foi pelo fato da biblioteca oferecer uma vasta quantidade de funções para processamento de imagens e possuir uma classe específica para reconhecimento facial (FaceRecognizer), que utiliza conceitos matemáticos clássicos de reconhecimento de padrões (Eigenfaces, Fisherfaces e Local Binary Patterns Histograms). Além disso, é open source e oferece boa portabilidade, sendo compatível com Windows, Linux, Unix-Like, e também com sistemas móveis como Android e IOS.

A biblioteca OpenCV possui uma documentação bastante abrangente e diversos fóruns e grupos de discussão, resultando em um suporte eficiente para os usuários da biblioteca. Dentre os documentos, o “*opencv2refman.pdf*”, disponível em: <http://docs.opencv.org/opencv2refman.pdf>, merece uma atenção especial, pois este é o manual de referência da OpenCV. Neste documento pode ser encontrada a descrição em detalhes das funções da biblioteca e códigos de exemplos. O manual também explica conceitos gerais referentes à OpenCV, como a definição de tipos de dados específicos e guias de implementação utilizados em processamento de imagens.

3. MÉTODOS DE DETECÇÃO DE FACES

Uma das etapas mais importantes em um Sistema de Reconhecimento Facial é a etapa de detecção da face em uma determinada imagem/vídeo. A vantagem de se detectar a face, em um primeiro momento, é que após esta fase, a procura pelas características fica limitada apenas em uma determinada região da imagem, diminuindo a velocidade de detecção e melhorando a precisão na extração de características [LOPES, 2009].

Entretanto, a detecção de faces a partir de uma simples imagem é uma tarefa desafiadora, pois a grande variabilidade de condições, como escala, localização, orientação, pose (perfil frontal), expressões faciais, obstáculos, luminosidade, etc. alteram a aparência geral das faces. A meta de detecção de faces é determinar se existem ou não quaisquer rostos na imagem e, se estiver presente, identifica que há faces na imagem/vídeo. Os desafios associados com a detecção de rosto podem ser atribuídos aos seguintes fatores [YANG; KRIEGMAN, AHUJA, 2002]:

- Pose: As imagens de face variam de acordo com a posição da câmera;
- Expressão Facial: A expressão da face influencia diretamente na aparência da imagem de face;
- Presença de Elementos: A presença de elementos como barba, bigode e óculos que podem modificar as características em termos de tamanho, luminosidade, etc.
- Ocultação: no caso de imagens feitas em ambientes não controlados, as faces podem aparecer, parcial ou totalmente sobrepostas, por objetos ou até mesmo por outras faces.

A seguir serão apresentados alguns métodos computacionais para a detecção de faces em imagens.

3.1 MÉTODO DE DETECÇÃO DE FACES VIOLA-JONES

A detecção e segmentação da face presente em vídeos e/ou imagens é o primeiro passo do processo de reconhecimento automático de faces. Esta etapa é considerada crítica, visto que uma falha nesta fase pode comprometer todo o processo. Analisando vários algoritmos de detecção de objetos (face e olhos, em particular), observou-se que o algoritmo Haar-Cascade, possui excelente desempenho e robustez necessários para o projeto. Além disso, diferenciando de outros algoritmos disponíveis, ele é implementado em uma imagem de escala de cinza e é quase insensível ao tamanho do objeto.

Este método foi apresentado em 2001 por Paul Viola e Michael Jones, sob o artigo intitulado “Robust Real-Time Face Detection” [VIOLA; JONES, 2001]. O algoritmo apresentado teve tanto sucesso que hoje ele é muito perto de ser o padrão para resolução de tarefas de detecção de faces. O método, a princípio restringiu-se a detecção de face frontal não inclinada, mas o trabalho futuro apresentado por Rainer Lienhart e Jochen Maydt [LIENHART; MAYDT, 2002] aprimorou-o, adicionando características rotacionais, que permitiram tolerância à inclinação.

O princípio básico do algoritmo Viola-Jones é varrer uma janela capaz de detectar faces através da imagem de entrada. A abordagem padrão era de redimensionar a imagem de entrada para tamanhos diferentes e depois varrer as imagens criadas com detector de tamanho fixo. Esta abordagem costuma consumir tempo devido aos cálculos de imagens de diferentes tamanhos.

Ao contrário desta abordagem, o algoritmo Viola-Jones redimensiona o próprio detector, ao invés da imagem, e varre a imagem várias vezes, cada vez com detector de tamanho diferente. Poderíamos a princípio pensar que ambas as abordagens consomem o mesmo tempo de processamento, porém Viola e Jones propuseram o desenvolvimento de um detector invariante de escala que requer o mesmo número de cálculos, independentemente do tamanho. Este detector é construído usando a chamada, “imagem integral”.

Segundo Viola e Jones, o método de combinação sucessiva de classificadores em estrutura cascata, que aumenta drasticamente a velocidade de detecção focando em regiões promissoras da imagem, já que muitas vezes é possível determinar de forma rápida onde na imagem a ocorrência do objeto de interesse é provável. Assim, o processamento mais complexo é reservado apenas para estas regiões.

3.1.1 Imagem Integral

Imagen integral é uma representação da imagem original que permite avaliar eficientemente a soma dos valores dos pixels (intensidade dos níveis de cinza) de uma área retangular em uma sub-região da imagem [WILSON; FERNANDEZ, 2006]. A Equação 12 mostra matematicamente esta representação.

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

Equação 12

A Figura 18, apresenta este conceito. Supomos que temos uma imagem 3x3, com intensidade de todos os pixels igual a 1, como mostrado na Figura 18(a). O valor da imagem integral será o que é mostrado na Figura 18(b).

Figura 18: Imagem Integral.

Assim, o valor para cada pixel pode ser obtido facilmente por uma única varredura da imagem, já que pode ser calculado pelos valores de pixels adjacentes:

$$ii(x, y) = i(x, y) + ii(x-1, y) + ii(x, y-1) - ii(x-1, y-1)$$

Equação 13

ii representa imagem integral e i o valor da imagem original, como podemos conferir na Figura 19.

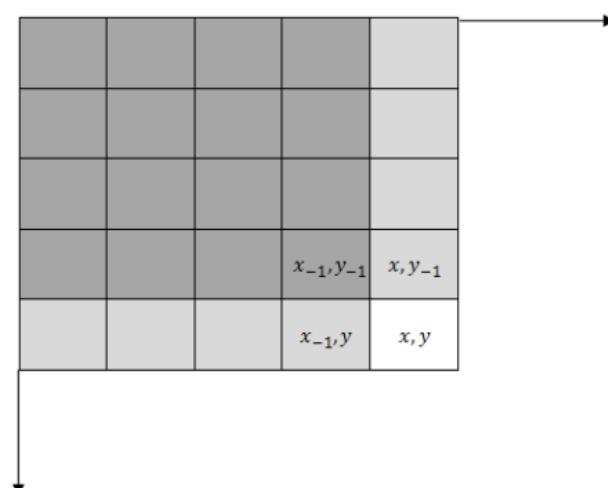


Figura 19: Cálculo de imagem integral.

Isto nos permite calcular a soma de pixels em um dado retângulo de tamanho qualquer usando apenas quatro valores básicos. Este conceito é demonstrado na Figura 20. Dados os valores da imagem integral nos pontos A B C e D, a soma do retângulo branco definido pelos pontos A e C é calculada como apresentado na Equação 14:

$$\sum_{(x, y) \in ABCD} i(x, y) = ii(A) + ii(D) - ii(B) - II(C)$$

Equação 14

Assim, o cálculo da soma de retângulos arbitrários é feita em tempo constante.

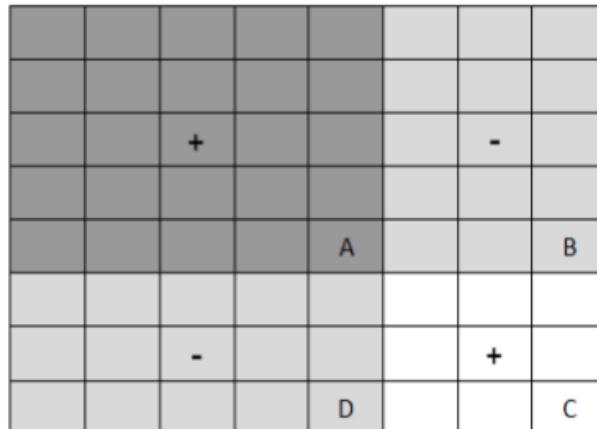


Figura 20: Cálculo de um retângulo.

O algoritmo é composto de três partes. A primeira delas é a representação da imagem em um espaço de características baseadas nos filtros de Haar. As características base são calculadas subtraindo-se a soma dos valores dos pixels da região branca em relação à soma dos valores dos pixels da região preta. A Figura 21 mostra os quatro tipos de características que foram propostas na abordagem original de Viola e Jones. Para calcular a característica da Figura 21(a), são necessárias oito consultas à tabela de soma de áreas (imagem integral), respectivamente para os oito pontos indicados.

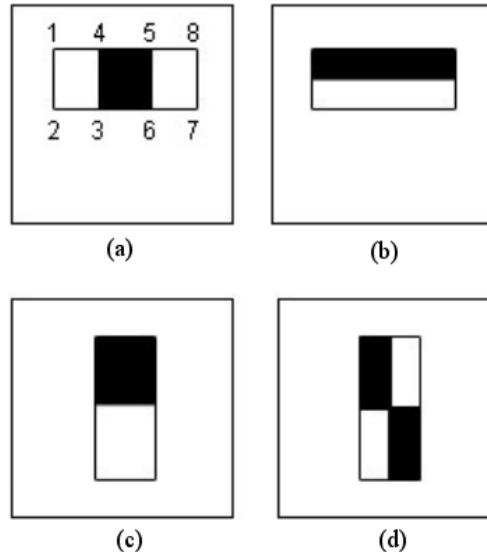


Figura 21: Quatro tipos de características base utilizadas na abordagem proposta por Viola e Jones.

Cada tipo de característica pode ajudar a reconhecer um determinado padrão, principalmente quando combinados em cascata. Por exemplo, a característica da Figura 21(b) permite identificar uma área na imagem onde há uma diferença de intensidade significativa entre a parte superior e a parte inferior de uma região. Essa característica pode ser aplicada no processo de detecção de faces, uma vez que frequentemente a região dos olhos é mais escura que outras regiões do rosto. Deve-se considerar também que existem outros padrões com o mesmo perfil e que não são faces, daí a necessidade de combinar várias características para refinar a busca.

A fim de aumentar a qualidade na detecção de faces, foram adicionadas ao algoritmo, características rotacionais que permitiram tolerância à inclinação [LIENHART; MAYDT, 2002]. Este conjunto inclui um novo tipo de característica e versões rotacionadas das características utilizadas na abordagem original. Além disso, a característica de quatro retângulos não é utilizada.

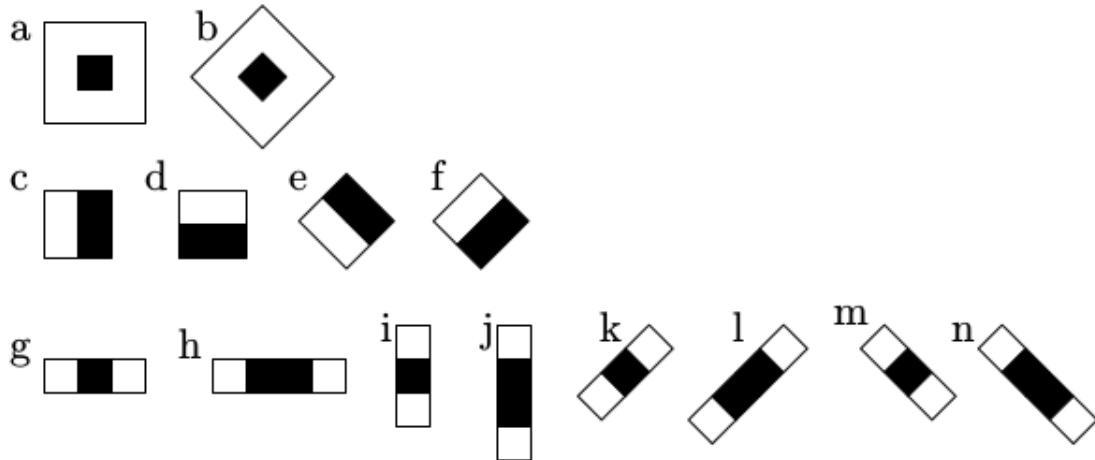


Figura 22: Conjunto estendido de características proposto por Lienhart e Maydt.

A segunda etapa no algoritmo de Viola-Jones é o treinamento de classificadores. Dado um conjunto de características deve-se treinar o sistema com imagens positivas (faces) e imagens negativas (tudo menos faces). Para isso devemos utilizar um algoritmo de treinamento que aprenda funções de classificação. Uma opção é utilizar um algoritmo de aprendizagem que use o método Boosting, que consiste em encontrar um classificador de alta precisão combinando-se muitos classificadores “fracos”, onde cada um desses classificadores fracos possui uma precisão média com uma taxa de acertos de pelo menos 51%.

Neste trabalho será abordado o algoritmo de aprendizagem de máquina denominado AdaBoost (Adaptive Boosting). O algoritmo AdaBoost, proposto por Yoav Freund e Robert Schapire [FREUND; SCHAPIRE, 1996] é um algoritmo meta-heurístico, e pode ser utilizado para aumentar a performance de outros algoritmos de aprendizagem. O nome AdaBoost deriva de Adaptive Boosting (em português, impulso ou estímulo adaptativo). O AdaBoost é adaptável no sentido de que as classificações subsequentes feitas são ajustadas a favor das instâncias classificadas negativamente por classificações anteriores.

AdaBoost é um algoritmo capaz de selecionar as características mais relevantes e que tem como premissa construir um classificador “forte” como uma combinação linear de vários classificadores fracos. A Equação 15 expressa essa ideia:

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x)$$

Equação 15

A função $h_t(x)$ representa classificadores fracos, e pode assumir valores 0 ou 1, respectivamente para exemplos negativos e positivos e (x) representa uma janela, tipicamente de 24x24. Mais especificamente, um classificador fraco pode ser expresso em função da característica (f) , de um threshold θ e de uma polaridade (p) , como mostra a Equação 16:

$$h(x, f, p, \theta) = \begin{cases} 1, & \text{se } pf(x) < p\theta \\ 0, & \text{caso contrário} \end{cases}$$

Equação 16

O classificador forte é dado pela função $H(x)$:

$$H(x) = \begin{cases} 1, & f(x) \geq \frac{1}{2} \sum \alpha_t \\ 0, & \text{caso contrário} \end{cases}$$

Equação 17

O algoritmo AdaBoost pode ser utilizado tanto para que este escolha quais características são mais adequadas, como também para treinar classificadores com estas características escolhidas. A Figura 23 mostra algumas características selecionadas pelo algoritmo para identificação de faces.

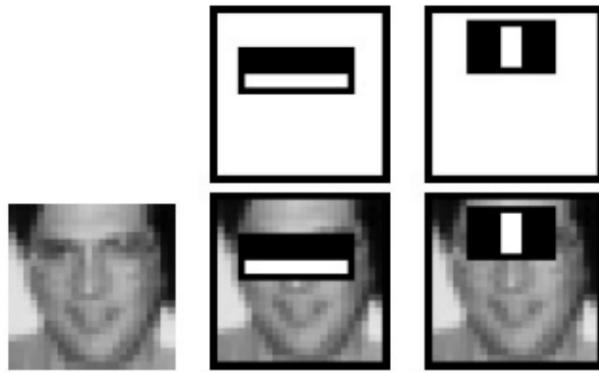


Figura 23: Características selecionadas pelo algoritmo AdaBoost.

A terceira e última etapa consiste em combinar classificadores fortes em cascata de modo a processar eficientemente regiões da imagem em busca de um padrão. Cada estágio na cascata aplica um classificador mais específico e complexo do que o anterior, de modo que o algoritmo rejeite rapidamente regiões que sejam muito distintas da característica procurada e termine o processo de procura, neste caso, evitando que os estágios posteriores sejam executados desnecessariamente., fazendo com que muitos dos cenários e panos de fundo sejam descartados nos primeiros estágios e apenas faces e objetos semelhantes sejam melhor analisados. A Figura 24 ilustra o processo.

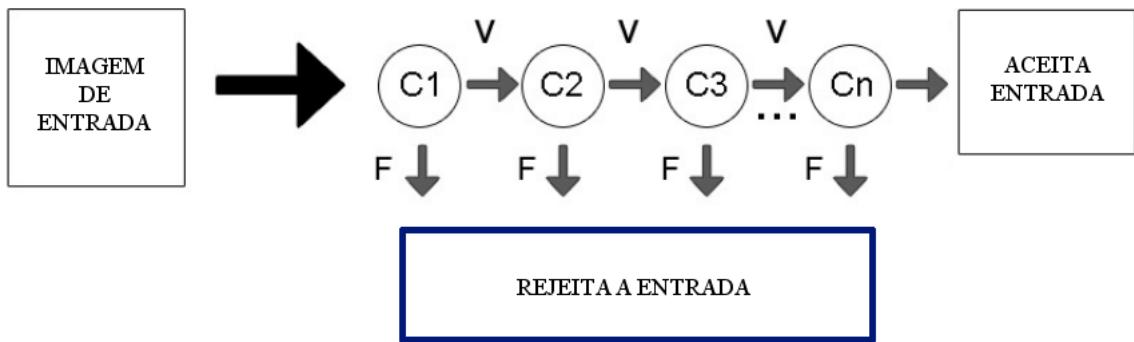


Figura 24: Cascata de classificadores.

O algoritmo de Viola-Jones, se bem treinado, tem uma boa precisão, e consegue detectar a maioria das faces que são captadas frontalmente, além de

apresentar um baixo número de falsos positivos. No entanto, o algoritmo falha quando tenta detectar faces de perfil, ou quando os olhos estão cobertos (usando-se óculos de sol, por exemplo), ou ainda, quando a iluminação ambiente é desfavorável. É possível fazer detecções em múltipla escala mudando a escala da imagem ou mudando a escala da janela de detecção. A Figura 25 ilustra como as janelas de detecção varrem a imagem em busca dos padrões definidos pelas características do tipo Haar, lembrando que em que cada região, vários classificadores são aplicados, e que a detecção só é considerada positiva caso a região seja classificada positivamente para todos os classificadores da cascata.



Figura 25: Varredura em busca de padrões baseado na diferença entre áreas claras e escuras de uma região.

3.2 OUTRAS ABORDAGENS PARA DETECÇÃO FACIAL

Existem diversas outras abordagens que tentam solucionar o problema de detecção facial [COSTA, 2003]. Algumas dessas soluções utilizam variações das ideias propostas por Viola e Jones. De um modo geral, as abordagens de detecção facial podem ser divididas em três grupos: métodos baseados em conhecimento humano e morfologia, métodos baseados em casamento de padrões e métodos baseados em aparência.

3.2.1 Métodos baseados em conhecimento humano e morfologia

Neste tipo de abordagem, os métodos de detecção facial são desenvolvidos a partir do conhecimento humano em relação a rostos, ou seja, uma face pode ser caracterizada por dois olhos dispostos com simetria vertical, nariz e boca, dentro de uma região aproximadamente elíptica. Pelo tamanho e posição de algumas dessas características, também é possível estimar o tamanho e a posição relativa das demais características.

O principal problema nesta abordagem é a dificuldade de traduzir em regras simples e bem definidas o conhecimento humano. Se as regras forem muito detalhadas (específicas), algumas faces podem ser rejeitadas (por exemplo, havendo a oclusão de alguma característica facial). Se ao contrário, as regras forem muito genéricas, falsos positivos podem ser gerados. Além disso, esta abordagem restringe-se apenas à detecção de faces que apareçam em uma (ou poucas) posição(ões).

Se considerarmos a possibilidade de as faces aparecerem rotacionadas, a simetria dos olhos pode aparecer horizontalmente e não mais verticalmente. Considerando também a possibilidade da ocorrência de faces vistas de perfil, seria

difícil enumerar todas as regras possíveis a serem testadas. Em testes experimentais realizados por Yang, Hriegman e Ahuja, o sistema identificou corretamente 50 de 60 faces (~ 83% de acerto) e, além disso, ainda apresentou falsos positivos em 28 imagens [YANG; KRIEGMAN; AHUJA, 2002].

3.2.2 Métodos baseados em casamento de padrões

Um exemplo de padrões é que o contorno de uma face pode ser parametrizado pela equação de uma elipse ou por diversos segmentos de reta, ou seja, um modelo de face padrão (normalmente frontal) é construído manualmente ou através de representações matemáticas apropriadas [YANG; KRIEGMAN; AHUJA, 2002]. O valor de correlação com a imagem padrão é calculado para o contorno da face, olhos, boca e nariz, independentemente e a existência de uma face é baseada nestes valores de correlação. Essa abordagem tem a vantagem de ser simples em relação à implementação, entretanto, o casamento de padrões tem se mostrado pouco eficiente para detecção facial, uma vez que não é identificada corretamente, as variações de escala, posição e forma.

O uso de padrões deformáveis vem tentando amenizar tais problemas, visto que eles se adaptam às variações morfológicas das características faciais, contudo, esse tipo de abordagem apresenta alguns problemas em relação à inicialização e seu uso muitas vezes torna-se complicado em função de o contorno das faces ser, em diversos casos, pouco definido. Os padrões deformáveis apresentam ainda elevada sensibilidade ao ruído e alto custo computacional [COSTA, 2003].

4. MÉTODOS PARA RECONHECIMENTO DE FACES

Geralmente as técnicas que pertencem a este grupo, necessitam de várias imagens e, a partir delas, aprendem ou codificam somente o que é necessário para realizar a detecção da característica de interesse. Exemplos de tal abordagem são as Redes Neurais, Eigenfaces e Modelos Ocultos de Markov [LOPES, 2009].

Nesta classe de algoritmos surgem 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. Um dos métodos mais consagrados é o de Autofaces (Eigenfaces), derivado da Transformada de Karhunen-Loève [MELLO, 2003], também conhecida por Análise de Componentes Principais (PCA), que será melhor detalhado na próxima seção.

4.1 ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

A análise da componente principal é uma técnica matemática que descreve um conjunto de dados usando “componentes principais”, escrita como combinações lineares dos dados originais. As componentes principais são determinadas em ordem decrescente de importância. A primeira componente possui mais informação do que a segunda e assim por diante. O método PCA tem por objetivo principal o processamento de dados a fim de reduzi-los ou eliminar possível redundância de informação [SANTOS, 2005].

Outra característica deste método, consiste na representação do conjunto de dados através de combinações lineares de suas variáveis que melhor o representam, portanto, o método PCA pode ser aplicado a diversos problemas cujas variáveis estão correlacionadas entre si e não apenas para o reconhecimento biométrico facial, objetivo desta proposta. Aplicado ao reconhecimento de faces, o

PCA mostra-se bastante propício, pois sua ideia de redução da dimensionalidade do conjunto de dados, que possui como característica principal um grande número de variáveis correlacionadas. O referido método, que também é conhecido como Transformada discreta de Karhunen-Loève (KLT) [MELLO, 2003], ou ainda Transformada de Hotelling, envolve uma base de Álgebra Linear e Estatística.

Ao longo do tempo, foram propostas algumas abordagens que utilizam o método PCA para representar de forma eficiente as fotos de faces, mas foi por Turk e Pentland [TURK; PENTLAND, 1991] que essa técnica obteve sucesso na classificação, permitindo o reconhecimento automatizado de faces, sendo conhecido amplamente como Eigenface. Eigenfaces são uma decomposição de um conjunto de imagens em autovalores e autovetores possibilitando a reconstrução de imagens, desde que se tenha um conjunto de treinamento grande o suficiente.

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 algoritmo Eigenface é dividido nas seguintes etapas: obtenção de um conjunto de faces, localização da face média, cálculo da matriz de covariância e redução da dimensionalidade da matriz.

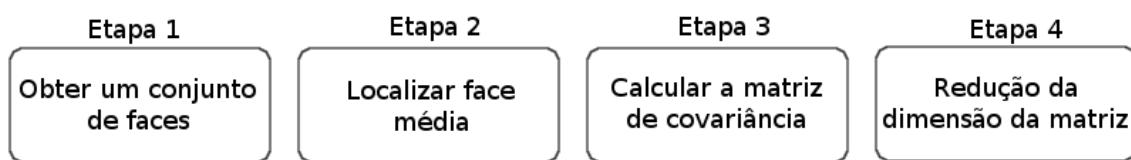


Figura 26: Etapas do algoritmo Eigenface.

Cada uma destas etapas serão melhor descritas nas seções subsequentes.

4.1.1 Conjunto de faces

Segundo Baggio, [BAGGIO; et al. , 2012] para esta etapa é necessário que as imagens tenham as mesmas características, como tamanho (altura e largura da imagem), condições de iluminação e elementos faciais facilmente localizados. Esse conjunto normalmente é formado por imagens em escala de cinza e equalizadas, e denominamos como conjunto de treinamento. Um exemplo desse conjunto é apresentado na Figura 27.



Figura 27: Conjunto de imagens de treino. Fonte: AT&T Facedatabase.

Cada imagem é convertida em um vetor de pixels, sendo determinada a dimensão desta imagem a partir do cálculo: $M = \text{Largura} * \text{Altura}$, na sequência, um vetor V_i de tamanho M é gerado e este vetor representa uma imagem da base. Com isso, no final deste passo tem-se um conjunto Γ com N imagens, como apresentado na Equação 18.

$$\Gamma = \Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_N$$

Equação 18

4.1.2 Localização da face média

Para eliminar informação redundante na face, isto é, uma imagem de face poderia ser representada com poucas componentes principais devido à sua redundância, é feito uma média aritmética do somatório dos vetores das faces. O vetor da face média ($\bar{\Gamma}$) é definido pela Equação 19.

$$\bar{\Gamma} = \frac{1}{N} \sum_i \Gamma_i$$

Equação 19

Esta etapa busca a diferença entre cada imagem base e a imagem média, sendo utilizado para gerar a matriz de covariância. A Equação 20, representa essa operação.

$$\Phi = \Gamma_i - \bar{\Gamma}$$

Equação 20

4.1.3 Cálculo da matriz de covariância

Após a subtração da imagem média de cada imagem, gera-se uma matriz A , como mostrado na Equação 21.

$$A = (\Phi_1, \Phi_2, \dots, \Phi_N)$$

Equação 21

Assim podemos calcular a matriz de covariância, com a transposição de A , ou seja A^t , conforme mostra a Equação 22.

$$C = A * A^T$$

Equação 22

4.1.4 Redução da dimensionalidade da matriz.

Por fim, na quarta etapa, são aplicadas a redução de dimensionalidade na matriz A , através do método de Análise de Componentes Principais (PCA), que necessita da matriz de covariância C , criando em uma nova matriz P com os autovalores μ e autovetores v mais significativos da matriz C .

$$P = v * \Phi$$

Equação 23

Em seguida o novo padrão passa para a fase de classificação, conforme é apresentado na Equação 24.

$$\Omega = P^t * \Phi$$

Equação 24

A Figura 28 apresenta as 10 principais eigenfaces e a face média do conjunto de treinamento correspondente a um dos usuários utilizados como exemplo neste trabalho.

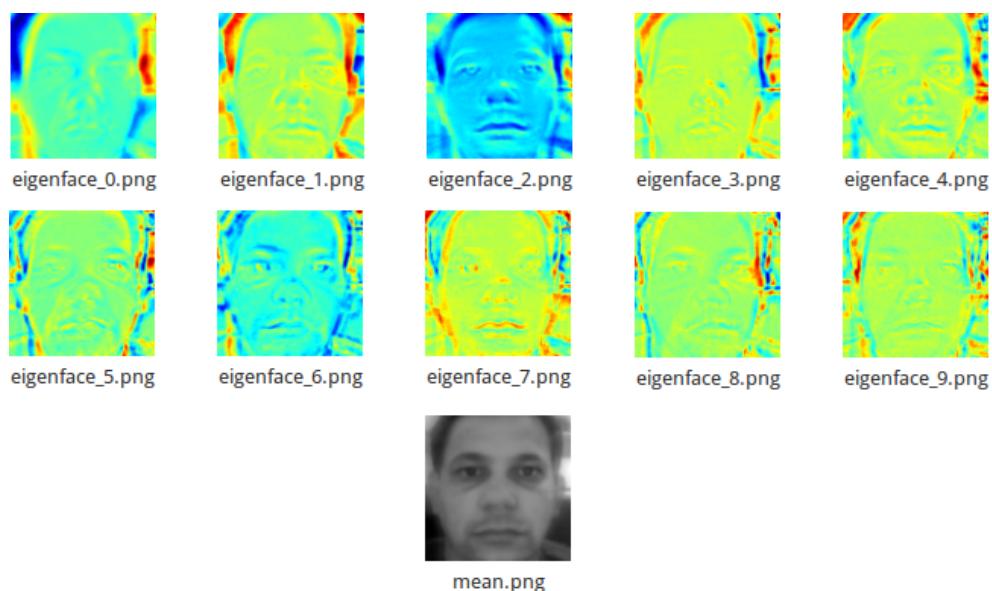


Figura 28: Exemplo de face média (abaixo) e Eigenfaces mais importantes para o conjunto de treino.

5 METODOLOGIA

Este capítulo apresenta a metodologia que utilizamos para o desenvolvimento do sistema de reconhecimento facial. O fluxograma completo, pode ser conferido no *Apêndice A*. O projeto envolve basicamente cinco etapas:

- Criação do conjunto de faces que serão treinadas;
- Treinar as faces e criar um modelo;
- Detecção de faces no vídeo;
- Pré-processamento da imagem de entrada;
- Reconhecimento e identificação da face.

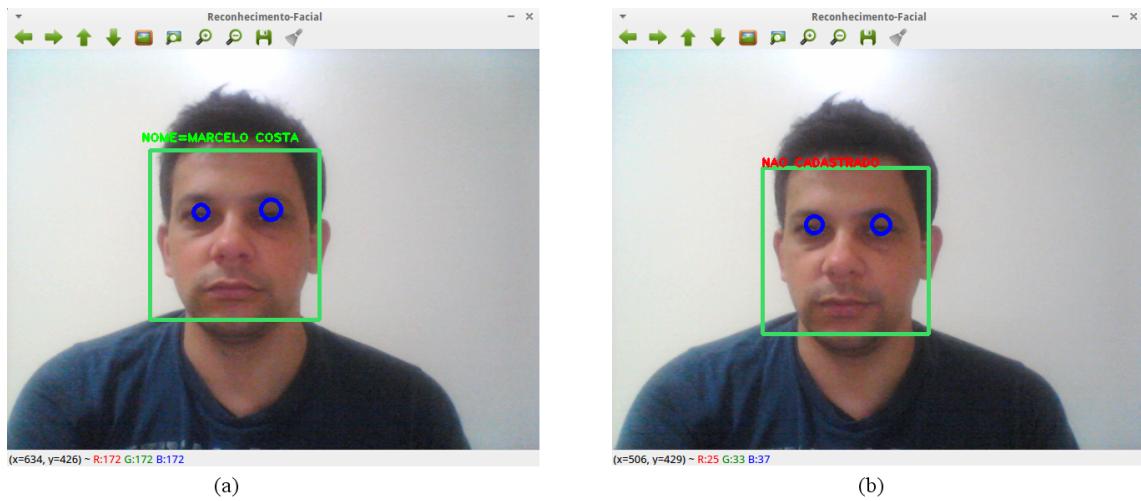


Figura 29: Sistema implementado. (a) identificação positiva em relação às faces treinadas; (b) identificação negativa em relação às faces treinadas.

Neste capítulo serão mostrados os métodos escolhidos e apresentadas as ferramentas usadas para o desenvolvimento de cada uma destas etapas.

5.1 BANCO DE IMAGENS FACIAIS

Para testar e avaliar um sistema de reconhecimento facial é imprescindível que se tenha um banco de dados com imagens de pessoas. Como o objetivo do trabalho é detecção de faces em um ambiente controlado, optou-se por adquirir imagens somente do rosto, por meio de uma webcam acoplada a um notebook na resolução de 100x100 pixels, em escala de cinza e equalizadas sob ambiente fechado, com iluminação predominantemente baseado em luz fluorescente.

Para facilitar a aquisição das imagens de face já pré-processadas, utilizamos o script “gera_fotos_treinamento.py” que, com o auxílio de um classificador para detecção de rostos, identifica e recorta somente a face da pessoa identificada na webcam e em 5 segundos armazena 20 imagens na resolução de 100x100 pixels, em escala de cinza e equalizadas por histograma.



Figura 30: Imagens para treinamento, geradas automaticamente.

Após a criação das imagens para treinamento, criamos um simples arquivo CSV para ser lido e consequentemente fazer o correto treinamento. Basicamente o arquivo CSV deve conter o caminho do arquivo da imagem e o respectivo rótulo (label). Um exemplo deste arquivo é mostrado no *Apêndice B*.

5.2 TREINAMENTO A PARTIR DAS FACES SALVAS NO BANCO DE IMAGENS

Com a base de dados de imagens concluída, devemos treinar o sistema para aprender sobre as imagens salvas. Há diversos algoritmos na literatura, mas o escolhido neste trabalho foi o de PCA *Eigenfaces*, que foi descrito anteriormente. As imagens estão no tamanho 100x100pixels, o que dá 10 mil pixels. Esse tamanho de matriz não é de total aproveitamento, por isso o uso do PCA, que reduz aos componentes principais. O Eigenface projeta todas as amostras de treinamento em um único subespaço, ou seja, no final do processo será gerada uma imagem média contendo as imagens de treinamento.

Ao final do processo a matriz treinada é salva em um arquivo para ser posteriormente carregado. Este arquivo armazena as seguintes informações das imagens treinadas:

- A quantidade de linhas e colunas da matriz de autovalores;
- Número de imagens do indivíduo;
- A quantidade de linhas e colunas da matriz de autovalores;
- A matriz de autovalores.

A partir destas informações, poderemos capturar a identificação do usuário (prediction), gerar a face média e reconstruir a imagem para fins de comparação. Um exemplo desta estrutura é mostrado no *Apêndice C*.

5.3 DETECÇÃO DE FACES

Este módulo do sistema recebe como entrada uma imagem que pode ou não conter faces humanas, e como saída retorna imagens somente das faces, caso estas existam. O algoritmo escolhido foi o de Viola-Jones por sua eficiência e baixa taxa de erros. Ao identificar a face a mesma é demarcada e recortada para que na etapa seguinte seja feito a extração de características somente na região da face. Além disso, o sistema transforma a imagem capturada em tons de cinza e a equaliza via histograma, pois quanto maior a padronização das imagens em relação às imagens de treinamento, maior será a eficiência do algoritmo.

A biblioteca OpenCV tem vários detectores pré-treinados que podem ser utilizados em diferentes propostas e não há necessidade de fazer um classificador para cada indivíduo, pois a grande maioria os seres humanos possuem os mesmos padrões, porém suas características são únicas. A tabela a seguir, lista alguns dos mais populares classificadores:

Tipo de classificador	Arquivo XML
Detector de face frontal (default)	haarcascade_frontalface_default.xml
Detector de face frontal (fast Haar)	haarcascade_frontalface_alt2.xml
Detector de face frontal (fast LBP)	lbpcascade_frontalface.xml
Detector de olhos com óculos	haarcascade_eye_tree_eyeglasses.xml
Detector olho esquerdo (mesmo fechado)	haarcascade_lefteye_2splits.xml
Detector olho esquerdo (mesmo fechado)	haarcascade_righteye_2splits
Detector de olhos	haarcascade_eye.xml
Detector de nariz	haarcascade_mcs_nose.xml
Detector de boca	haarcascade_mcs_mouth.xml
Detector de pessoa inteira	haarcascade_fullbody.xml

Tabela 3: Classificadores pré-treinados

Neste trabalho, foram utilizados os detectores de face frontal (default) e os de olhos esquerdo e direito.

5.4 PRÉ-PROCESSAMENTO DA IMAGEM DE ENTRADA

O pré-processamento da imagem de entrada consiste em fazer ajustes na imagem detectada no vídeo. Nesta etapa, as imagens de faces detectadas pelo algoritmo Viola-Jones são normalizadas e corrigidas de acordo com as imagens de treinamento, ou seja, serão convertidas para tons de cinza, equalizadas e redimensionadas para o tamanho de 100x100 pixels. Com esse procedimento, há uma considerável melhora no reconhecimento da face [BAGGIO; et al., 2012].

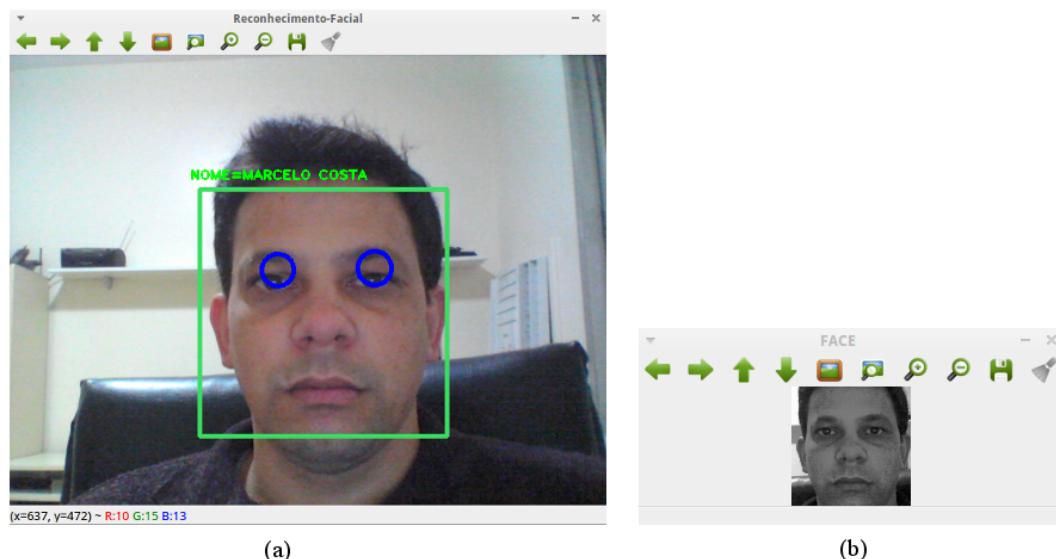


Figura 31: Imagens de entrada. (a) imagem original em resolução 640x480 pixels; (b) imagem pré-processada.

5.5 RECONHECIMENTO E IDENTIFICAÇÃO DA FACE.

Após calculados os autovetores e autovalores e salvos em um arquivo (eigenfaces.yml), como mostrado no capítulo 5.2 e obtida a imagem pré-processada, são necessários, no caso deste trabalho, dois cálculos entre a imagem de entrada pré-processada e as imagens treinadas, para que o reconhecimento seja concluído. De acordo com nossos testes, os dois cálculos propostos, dão uma maior confiabilidade quando utilizados em conjunto.

O primeiro é o cálculo da soma da distância quadrada entre os pontos da imagem pré-processada com cada uma das faces de treinamento e suas respectivas projeções, ou seja, é baseado na distância euclidiana entre o k-vizinho mais próximo (K-Nearest Neighbors) encontrado nas imagens treinadas. Com esse procedimento, temos um limiar (threshold) heurístico. Esse valor de limiar é alterado de acordo com a imagem de entrada, iluminação, distância da câmera, etc. A Equação 25 apresenta a forma tradicional da distância euclidiana.

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

Equação 25

Para o segundo cálculo, utilizamos a proposta desenvolvida por Baggio [BAGGIO, et al.;2012], em que a face é reconstruída, a partir dos coeficientes de projeção no subespaço dos autovalores e autovetores, onde podemos comparar a imagem reconstruída com a imagem pré-processada, através do cálculo de erro relativo L2. O erro relativo L2, baseia-se na raiz quadrada da soma do erro quadrático de cada pixel, entre a imagem reconstruída e a pré-processada, ou seja, para comparar a semelhança das duas imagens, pixel por pixel, envolve em apenas subtrair uma imagem da outra, somando o valor quadrado dela, e em seguida, obter a raiz quadrada do mesmo, encontrando o erro relativo L2.



Figura 32: Comparação entre imagens: (a) imagem reconstruída; (b) imagem pré-processada.

Caso a diferença entre as imagens seja menor que $0,3$, implica que elas são muito similares. Para Eigenfaces, em que existe um vetor próprio para cada face, de modo que a reconstrução tende a ser muito semelhante a face pré-processada, podemos utilizar um limiar de $0,5$. Para ter mais confiabilidade ao usuários reconhecidos, utilizamos o valor de limiar (threshold) em $0,4$.

```
// Compare two images by getting the L2 error (square-root of sum of squared error).
double getSimilarity(const Mat A, const Mat B) {
    if (A.rows > 0 && A.rows == B.rows && A.cols > 0 && A.cols == B.cols) {
        // Calculate the L2 relative error between the 2 images.
        double errorL2 = norm(A, B, CV_L2);
        // Convert to a reasonable scale, since L2 error is summed across all pixels of the image.
        double similarity = errorL2 / (double)(A.rows * A.cols);
        return similarity;
    }
}
```

Tabela 4: Código do erro L2: Raíz quadrada da soma do quadrado do erro

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou técnicas clássicas em sistemas de reconhecimento facial, cobrindo desde a parte teórica até os passos necessários para o desenvolvimento de um programa computacional. Foi verificado que a implementação de um sistema de reconhecimento facial exige algoritmos capazes de lidar com variações de pose do indivíduo, ruídos na imagem de entrada e diferentes configurações de iluminação ambiente.

O sistema desenvolvido é constituído em módulos de detecção facial, extração de características, pré-processamento e de reconhecimento. O módulo de detecção facial foi baseado no algoritmo de Viola-Jones e detectou com eficiência 92% das faces apresentadas. O módulo de extração de características, baseado na análise de componentes principais (PCA), mostrou-se eficiente e rápido ao calcular o subespaço de características (autovalores e autovetores).

No pré-processamento, fundamental para obter um cálculo preciso entre a imagem de entrada e as imagens treinadas, foram atingidos todos os objetivos propostos.

Na parte de reconhecimento, foram aplicadas duas métricas para melhorar a identificação, o que garantiu maior confiabilidade de que a imagem de entrada corresponde a pessoa que se espera encontrar. Foram atingidos razoáveis níveis de eficiência em faces plenamente frontais e, além disso, mostrou-se tolerante a ruído e conseguiu identificar indivíduos mesmo com a região dos olhos cobertos. Em contrapartida, quando as regiões de interesse não estavam alinhadas, a taxa de erros aumentou significativamente.

6.1 TRABALHOS FUTUROS

Este trabalho permitiu o aprendizado em diversos campos do processamento de imagens, por isso, os trabalhos futuros podem envolver somente partes deste trabalho. Um exemplo, seria a detecção de objetos (carros, pessoas, etc), para fins de contagem, mas para melhorias em relação ao reconhecimento facial automatizado, colocamos os seguintes itens:

- Aumento do banco de faces, incluindo mais usos de acessórios, poses, iluminação e oclusão de regiões da face;
- Implementar outros métodos de detecção da face;
- Implementar métodos para diferenciar fotos de pessoas reais;
- Implementar outros métodos de classificação de faces;
- Novas alternativas no pré-processamento das imagens de entrada;
- Alinhamento da face de entrada.

7. REFERÊNCIAS BIBLIOGRÁFICAS

BAGGIO, Daniel L., et al. **Mastering OpenCV with Practical Computer Vision Projects**. Packt Publishing Ltd., ISBN 978-1-84951-782-9, 2012.

BRADSKI, Gary; KAEHLER, Adrian. **Learning OpenCV**. O'Reilly Media, 2008. ISBN: 978-0-596-51613-0.

CÂNDIDO, Jorge; MARENCONI, Maurício. **Combining Information in a Bayesian Network for Face Detection**. Brazilian Journal of Probability and Statistics, 2009, vol. 23, nº 2, p. 179–195, DOI: 10.1214/08-BJPS021.

CANNY, John. **A Computational Approach for Edge Detection**. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, nº 6, p. 679-698, 1986.

COSTA, Guilherme H. **Deteção Facial: Autofaces versus Antifaces**. Dissertação de Mestrado em Engenharia Elétrica. Florianópolis: Universidade Federal de Santa Catarina, 2003.

DINIZ, Fábio A., et al. **RedFace: um sistema de reconhecimento facial baseado em técnicas de análise de componentes principais e autofaces: comparação com diferentes classificadores**. Revista Brasileira de Computação Aplicada (ISSN 2176-6649), Passo Fundo, vol. 5, nº 1, p. 42-54, 2013.

FILHO, Ogê M.; NETO, Huvo V. **Processamento Digital de Imagens**. Brasport, 1999. ISBN 8574520098.

FREUND, Yoav; SCHAPIRE, Robert E. **A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting**. AT&T Labs, 1996

GOMES, Otávio F. M. **Processamento e Análise de Imagens Aplicados à Caracterização Automática de Materiais**. Dissertação de Mestrado em Ciência da Engenharia Metalúrgica. Rio de Janeiro: Departamento de Ciência de Materiais e Metalurgia, Pontifícia Universidade Católica do Rio de Janeiro, 2001.

GONZALES, Rafael C.; WOODS, Richard E. **Digital Image Processing**. Prentice Hall, ISBN 0-201-18075-8, 1992.

LATA, Y. V., et al. **Facial Recognition using eigenfaces by PCA”.** International Journal of Recent Trends in Engineering, vol. 1, nº 1, 2009.

LIENHART, Rainer; MAYDT; Lienhart. **An Extended Set of Haar-Like Features for Rapid Object Detection.** IEEE ICIP, vol. 1, p. 900-903, 2002.

LOPES, Eduardo C. **Detecção de Faces e Características Faciais.** Relatório Técnico nº 45. Porto Alegre: Pós-Graduação em Ciência da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, 2009.

MARQUES, Jorge S. **Reconhecimento de Padrões: Métodos Estatísticos e Neurais.** 2^a ed. Lisboa: IST Press, 2005.

MARR, David; HILDRETH, Ellen C. **Theory of Edge Detection.** Proceedings of The Royal Society of London, vol. B207, p. 187-217, 1980.

MELLO, Ricardo N. B. **Estudo comparativo da Transformada Karhunen-Loève na Compressão de Imagens.** Dissertação de mestrado. Porto Alegre: UFRGS/Programa de Pós-Graduação em Engenharia Elétrica, 2003.

RAUDYS, Sarunas J.; JAIN Anil K. **Small Sample Size Effects in Statistical Pattern Recognition: Recommendations for Practitioners.** IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, nº. 3, p. 252-264, 1991.

ROSENFELD, Azriel; KAK, Avinash C. **Digital Picture Processing.** Academic Press, Inc., 1982.

SANTOS, Anderson R. **Identificação de faces humanas através de PCA-LDA e Redes Neurais SOM.** Dissertação de mestrado. São Paulo: USP/Departamento de Engenharia Elétrica, 2005.

SOBEL, Irwin; FELDMAN, Gary. **A 3 × 3 Isotropic Gradient Operator for Image Processing.** Presented at the Stanford Artificial Project, 1968.

THIAN, N. **Biometric Authentication System.** Dissertação de mestrado, Penang, Malásia: USM, 2001

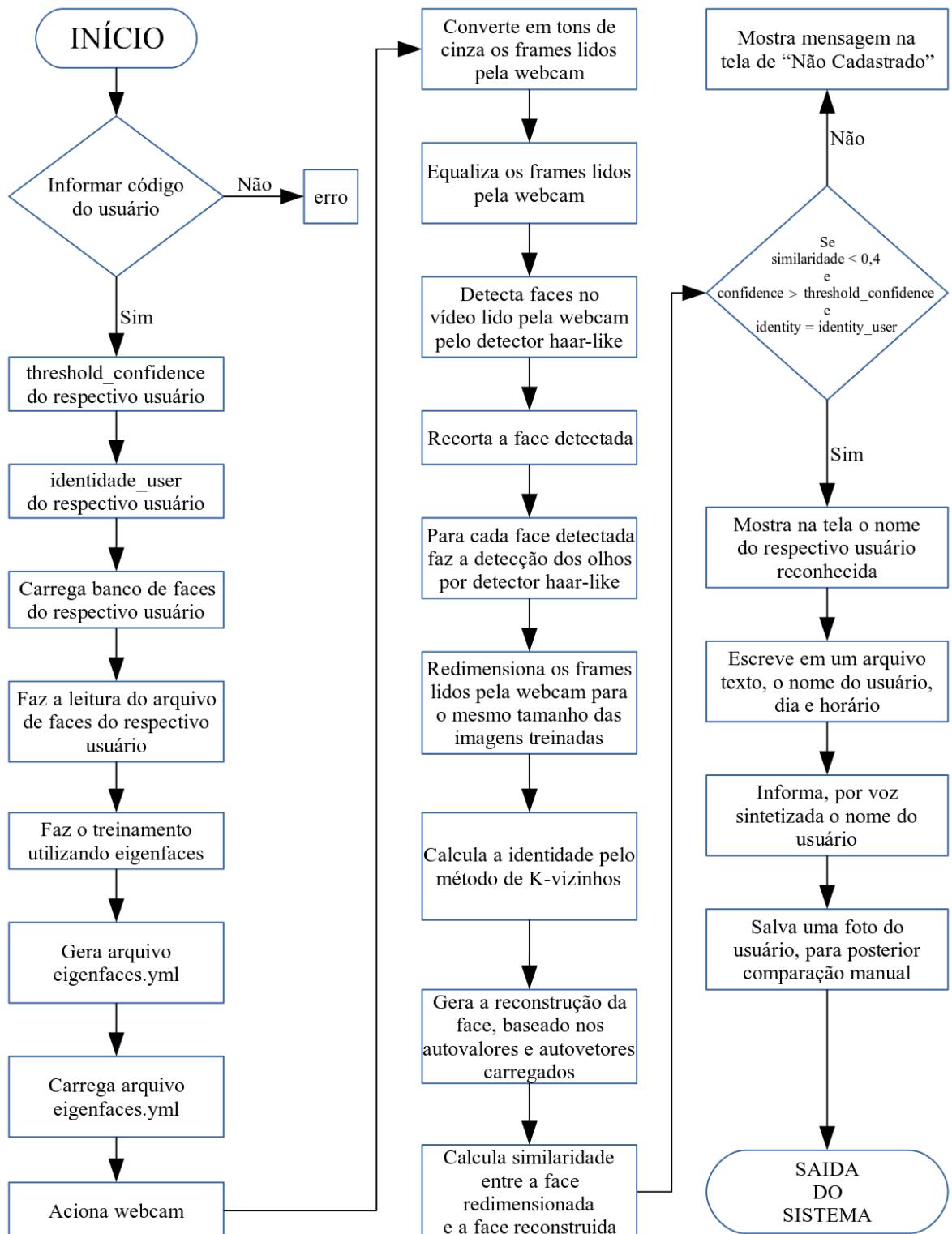
TURK, Matthew; PENTLAND, Alex. **Eigenfaces for Recognition.** Journal of Cognitive Neuroscience, MITPress, vol. 3, nº 1, p. 71–86, 1991.

VIOLA, Paul; JONES, Michael J. **Robust Real-Time Face Detection.** International Journal of Computer Vision 57(2), p. 137-154, 2001.

WILSON, Phillip I.; FERNANDEZ, Jonh. **Facial feature detection using Haar classifiers.** Journal of Computing Sciences in Colleges archive, vol. 21, p. 127-133, 2006.

YANG, Ming H.; KRIEGMAN; David J.; AHUJA, Narendra. **Detecting Faces in Images: A Survey.** IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, n° 1, 2002.

APÊNDICE A – FLUXOGRAMA FUNCIONAL DO SISTEMA IMPLEMENTADO



APÊNDICE B – EXEMPLO DE ESTRUTURA DE FACES PARA TREINAMENTO

imagens_treinamento_usuarios/marcelo/1.jpg;0
imagens_treinamento_usuarios/marcelo/2.jpg;0
imagens_treinamento_usuarios/marcelo/3.jpg;0
imagens_treinamento_usuarios/marcelo/4.jpg;0
imagens_treinamento_usuarios/marcelo/5.jpg;0
imagens_treinamento_usuarios/marcelo/6.jpg;0
imagens_treinamento_usuarios/marcelo/7.jpg;0
imagens_treinamento_usuarios/marcelo/8.jpg;0
imagens_treinamento_usuarios/marcelo/9.jpg;0
imagens_treinamento_usuarios/marcelo/10.jpg;0
imagens_treinamento_usuarios/marcelo/11.jpg;0
imagens_treinamento_usuarios/marcelo/12.jpg;0
imagens_treinamento_usuarios/marcelo/13.jpg;0
imagens_treinamento_usuarios/marcelo/14.jpg;0
imagens_treinamento_usuarios/marcelo/15.jpg;0
imagens_treinamento_usuarios/marcelo/16.jpg;0
imagens_treinamento_usuarios/marcelo/17.jpg;0
imagens_treinamento_usuarios/marcelo/18.jpg;0
imagens_treinamento_usuarios/marcelo/19.jpg;0
imagens_treinamento_usuarios/marcelo/20.jpg;0
imagens_treinamento_usuarios/marcelo/21.jpg;0
imagens_treinamento_usuarios/marcelo/22.jpg;0
imagens_treinamento_usuarios/marcelo/23.jpg;0
imagens_treinamento_usuarios/marcelo/24.jpg;0
imagens_treinamento_usuarios/marcelo/25.jpg;0
imagens_treinamento_usuarios/marcelo/26.jpg;0
imagens_treinamento_usuarios/marcelo/27.jpg;0
imagens_treinamento_usuarios/marcelo/28.jpg;0
imagens_treinamento_usuarios/marcelo/29.jpg;0
imagens_treinamento_usuarios/marcelo/30.jpg;0

...

imagens_treinamento_usuarios/marcelo/93.jpg;0
imagens_treinamento_usuarios/marcelo/94.jpg;0
imagens_treinamento_usuarios/marcelo/95.jpg;0
imagens_treinamento_usuarios/marcelo/96.jpg;0
imagens_treinamento_usuarios/marcelo/97.jpg;0
imagens_treinamento_usuarios/marcelo/98.jpg;0
imagens_treinamento_usuarios/marcelo/99.jpg;0
imagens_treinamento_usuarios/marcelo/100.jpg;0

APÊNDICE C – EXEMPLO DE ESTRUTURA DE EIGENFACES GERADA

