UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Investigação do Problema de Detecção de Faces com Variações de Orientação

Eanes Torres Pereira

Campina Grande, Paraíba, Brasil
Junho de 2012

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Investigação do Problema de Detecção de Faces com Variações de Orientação

Eanes Torres Pereira

Tese submetida à coordenação do Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande, Campus I, como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Modelos Computacionais e Cognitivos

Herman Martins Gomes
João Marques de Carvalho
Orientadores

Campina Grande, Paraíba, Brasil
Junho de 2012

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

P436i Pereira, Eanes Torres.
Investigação do problema de detecção de faces com variações de orientação / Eanes Torres Pereira. - Campina Grande, 2012.
212 f. : il., color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientadores: Prof. Herman Martins Gomes, Prof. João Marques de Carvalho.
Referências.

1. Ciência da Computação. 2. Visão Computacional. 3. Detecção de Faces. 4. Invariância por Treinamento. 5. Cascatas de Classificadores. 6. Paralelismo Híbrido. I.Título.

CDU 004(043)

Universidade Federal de Campina Grande - UFCG
Centro de Engenharia Elétrica e Informática - CEEI
Programa de Pós-Graduação em Ciência da Computação - PPGCC
Av. Aprígio Veloso,,882 - 58429-140, Campina Grande, PB
Fone: (83) 2101-1124 - Fax: (83) 2101-1123 - Email: copin@copin.ufcg.edu.br

PARECER FINAL DO JULGAMENTO DA TESE DO(A) DOUTORANDO(A)

EANES TORRES PEREIRA

TÍTULO: "INVESTIGAÇÃO DO PROBLEMA DE DETECÇÃO DE FACES COM VARIAÇÕES DE ORIENTAÇÃO"

COMISSÃO EXAMINADORA	CONCEITO
HERMAN MARTINS GOMES, Ph.D Orientador(a)	Aprovado
JOÃO MARQUES DE CARVALHO, Ph.D Orientador(a)	Aprovado
JOSÉ EUSTAQUIO RANGEL DE QUEIROZ, D.Sc Examinador(a)	Aprovado
LEANDRO BALBY MARINHO, Dr. Examinador(a)	Aprovado
CLÁUDIO ROSITO JUNG, Dr. Examinador(a)	
ROGERIO SCHMIDT FERIS, Ph.D Examinador(a)	

INSTITUTO DE INFORMÁTICA
UFRGS

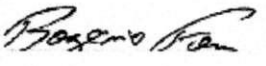
Declaro para os devidos fins que participei como examinador externo da banca de doutorado de Eanes Torres Pereira, intitulada "Investigação do Problema de Detecção de Faces com Variações de Orientação". A banca ocorreu no dia 25 de junho de 2012 às 14h nas dependências da UFCG e minha participação ocorreu à distância por meio virtual, utilizando a ferramenta Skype. Após a análise da tese e da apresentação realizada pelo candidato, atribuí o conceito **Aprovado**, com o estabelecimento de um prazo de 60 dias para a implementação das recomendações e correções apontadas na ficha de avaliação e no documento de tese.

Porto Alegre, 25 de junho de 2012.

Cláudio Rosito Jung
Professor Adjunto
INF-UFRGS

26/05/2012

Declaro para os devidos fins que participei como examinador externo da banca de doutorado de Eanes Torres Pereira, intitulada "Investigação do Problema de Detecção de Faces com Variações de Orientação". A banca ocorreu no dia 25 de junho de 2012 às 14h nas dependências da UFCG e minha participação ocorreu à distância por meio virtual, utilizando a ferramenta Skype. Após a análise da tese e da apresentação realizada pelo candidato, atribuí o conceito Aprovado, com o estabelecimento de um prazo de 60 dias para a implementação das recomendações e correções apontadas na ficha de avaliação e no documento de tese.



****

Rogerio Schmidt Feris
Research Scientist, IBM Master Inventor
Exploratory Computer Vision Group
IBM T. J. Watson Research Center
Phone: 914-784-6704
Fax: 914-784-7455
email: rsferis@us.ibm.com
http://rogerioferis.com

Resumo

Nesta tese, investiga-se o problema da detecção de faces que apresentam grandes variações de orientação. Foram identificados fatores capazes de influenciar os resultados quando determinadas métricas de avaliação são utilizadas. Por exemplo, se a métrica empregada leva em consideração as áreas de detecção obtidas pelos classificadores e as áreas rotuladas por humanos (*groundtruth*), a forma como as imagens detectadas são marcadas inteferirá nos resultados. Em relação ao aspecto de recorte das faces, os resultados experimentais comprovam que se forem incluídas regiões externas da face para treinamento, os resultados de detecção são melhorados.

Para lidar com todos esses fatores, foi proposta e implementada uma abordagem para a detecção de faces que explora a invariância por treinamento para gerar uma árvore de classificadores com menor complexidade computacional do que outras abordagens propostas na literatura, capaz de lidar com grandes variações de orientação no plano da imagem. A fim de tornar factível o treinamento dos classificadores dessa árvore, é apresentada uma abordagem híbrida de paralelização para o método de treinamento de classificadores proposto por Viola e Jones (2004). A abordagem de detecção de faces proposta obteve resultados superiores àqueles obtidos por Rowley, Baluja e Kanade (1998b) e Viola e Jones (2004). Apenas uma das abordagens concorrentes, aquela proposta por Huang et al. (2007), obteve resultados superiores, porém por uma pequena diferença. Apesar disso, a abordagem proposta nesta tese possui menor complexidade computacional em termos de quantidade de níveis da árvore de classificadores e quantidade de nós de processamento.

Abstract

In this thesis, the problem of detecting faces that present high variations of orientation is investigated. Some factores were identified that may influence the detection results when some evaluation metrics are used. For example, if the applied metric takes in consideration the detected areas obtained by the classifiers and the human labeled areas (groundtruth), the way as the detected images are marked will interfere in the computed results. In relation to the face image cropping aspect, the experimental results show that if external regions of the faces are included for training, the detection results will be better.

To deal with all those factors, it was proposed and implemented an approach to face detection that explores the invariance by training to yield classifier tree with lower computational complexity than other approaches in the state of the art, and able to deal with high angle in-plane orientations. To make the training of the cascades of classifiers feasible, a hybrid parallel approach of the training method of Viola e Jones (2004) was proposed. The parallel approach is able to achieve superlinear speedup, as it was demonstrated in the experiments. The face detection approach obtained higher results than those obtained by Rowley, Baluja e Kanade (1998a), and Viola e Jones (2004). Only one of the evaluated approaches obtained higher results, that proposed by Huang et al. (2007). However, the approach proposed in this thesis has lower computational complexity in terms of quantity of levels in the classifier tree, and quantity of processing nodes.

Agradecimentos

Todas as pessoas com as quais temos contato influenciam de alguma forma nossas vidas. Portanto, sendo esta tese resultado da influência de todos aqueles que foram meus professores gostaria de agradecer-lhes por sua participação, mesmo que indireta, na produção deste trabalho.

Dentre meus professores, agradeço especialmente àqueles que nos últimos anos se tornaram meus mestres *de facto*: Herman Martins Gomes e João Marques de Carvalho.

Sou muito grato, também, aos membros da banca examinadora que, desde a época de qualificação da tese, nos ajudaram bastante com seus comentários, críticas e idéias para moldar este trabalho.

Não são apenas os professores profissionais que nos ensinam, agradeço a todos os amigos e conhecidos que de alguma forma me ajudaram nesse percurso. Só o fato de terem se interessado pelo trabalho ou perguntado sobre a pesquisa já é motivo para que eu lhes agradeça.

Agradeço à minha esposa por sua paciência comigo durante esta fase na qual tive muitos períodos de rabugisse e insolência.

Agradeço à minha família pela dedicação e compreensão pela minha ausência durante esses anos.

Enfim, agradeço a todos os meus amigos, especialmente aos colegas de laboratório e de trabalho por todas as idéias decorrentes de nossas conversas.

Que os méritos deste trabalho se estendam a todos!

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Siglas e Abreviaturas

- AG: Algoritmo Genético
- ANOVA: Analysis of Variance
- AVG: *Average*
- API: *Application Programming Interface*
- CMU: *Canergie Mellon University*
- DP: Desvio Padrão
- EER: *Equal Error Rate*
- FDDB: *Face Detection Data Base*
- FM: *F-measure*
- FP: *Falso Positivo*
- FRT: *Face Ratios Template*
- HS: *Harr Scan*
- INTLBP: *Integral Local Binary Pattern*
- LBP: *Local Binary Pattern*
- LBP^{ri}: *Rotation Invariant Local Binary Pattern*
- LDA: *Linear Discriminant Analysis*

- MIT: *Massachusetts Institute of Technology*
- MLP: *Multilayer Perceptron*
- OCI: *Object Class Invariant*
- PCA: *Principal Component Analysis*
- RBF: *Radial Basis Function*
- ROC: *Receiver Operating Characteristic*
- ROF: Razões Otimizadas de Faces
- SIFT: *Scale Invariant Feature Transform*
- SVM: *Support Vector Machine*
- VP: Verdadeiro Positivo

Lista de Algoritmos

Capítulo 1

Introdução

O problema central desta pesquisa é a detecção automática, em imagens digitais, de faces humanas com variações de orientação. Neste capítulo, serão introduzidos, dentre outros elementos, os fatores que motivaram e justificam o desenvolvimento desta tese. Além disso, serão explicitados os objetivos da pesquisa e sua relevância para a área de Visão Computacional.

1.1 Motivação

O uso de computadores e de suas mídias, especialmente de imagens e vídeo, está se popularizando bastante. A utilização de imagens digitalizadas varia desde os casos considerados menos científicos, os quais englobam, por exemplo, as imagens de identificação de usuários em sites de relacionamento pela Internet até utilizações científicas que englobam processamento de imagens de telescópios espaciais ou o reconhecimento de criminosos utilizando imagens obtidas por câmeras de vídeo instaladas em locais públicos.

No contexto de monitoramento de vídeo, é comum existirem câmeras de segurança instaladas nos mais variados lugares, desde pequenos estabelecimentos comerciais até grandes aeroportos internacionais. Em lugares bastante movimentados é difícil ter funcionários trabalhando apenas no monitoramento das imagens em busca de suspeitos. O ser humano é passível de distrações e pode deixar escapar detalhes importantes. Esse ponto fraco pode ser

tratado de forma automatizada. Um sistema de reconhecimento de faces poderia ser acoplado ao sistema de câmeras de segurança com o intuito de identificar potenciais suspeitos.

Uma necessidade que vem aumentando com o avanço das tecnologias de aquisição de imagens digitais é a capacidade de agrupar fotografias digitais por temas específicos e um desses temas pode ser a presença de pessoas. Supondo que alguém tenha milhares de fotografias da família armazenadas em seu computador, seria bastante trabalhoso para essa pessoa selecionar o subconjunto dessas fotografias que contenha, por exemplo, apenas imagens de um filho específico. Neste problema, mais uma vez, um sistema de reconhecimento de faces seria bastante útil [1].

No entanto, para que seja possível reconhecer determinado objeto em uma imagem, esse objeto deve ser detectado. Ou seja, a localização espacial da área de interesse do objeto na imagem deve ser definida. Logo, um problema que deve ser tratado antes do reconhecimento de objetos é sua detecção e localização. Na próxima seção, será apresentada a descrição do problema tratado nesta tese.

1.2 Definição do Problema

De modo geral, a detecção de faces envolve identificar a existência de faces em uma imagem digital e indicar sua localização na imagem por meio, por exemplo, das coordenadas dos cantos superior esquerdo e inferior direito de um retângulo que contém a face. O modo tradicional de realizar a detecção de faces é percorrer a imagem pixel a pixel, determinar uma região de interesse e classificar a região usando um classificador previamente treinado. O ato de percorrer a imagem é realizado por meio da aplicação de deslizamento de janela (*sliding window*).

Há várias formas de escrutínio de uma imagem por uma janela deslizante. Duas das formas mais usadas envolvem redimensionar a imagem ou redimensionar a janela de interesse ao fim de cada deslizamento, até atingir uma dimensão limite. A principal dificuldade reside, então, no treinamento de classificadores adequados. Alguns dos detectores mais conhecidos

[1] O software Picasa da empresa Google realiza agrupamento de fotografias por reconhecimento facial

utilizaram abordagens de treinamento de classificadores por aparência, conforme descrito a seguir:

- Sung e Poggio (1998): redes neurais treinadas com características obtidas de distribuições probabilísticas de modelos de faces.
- Rowley, Baluja e Kanade (1998a): redes neurais treinadas com intensidades dos pixels das imagens;
- Schneiderman e Kanade (2000a): redes neurais treinadas com características do tipo *wavelet*;
- Viola e Jones (2001), Viola e Jones (2004): cascata de classificadores fracos treinados com características do tipo Haar e combinação otimizada por Adaboost;
- Huang et al. (2007): árvore de classificadores treinados com características esparsas e *vector boosting*.

De acordo com Yang, Kriegman e Ahuja (2002), os métodos para a detecção de faces em imagens podem ser agrupados em quatro categorias, mas, para efeito de simplificação, essas abordagens podem ser reagrupadas em duas, a saber: baseadas em características faciais e baseadas em aparência. As abordagens baseadas em características fazem uso de atributos que são inerentes às faces, tais como o fato de possuírem dois olhos, uma boca e um nariz. Além disso, as abordagens baseadas em características fazem uso das relações existentes entre os atributos da face. Por exemplo, Lin e Fan (2000) utilizam o fato de que os olhos e o centro da boca formam uma região triangular na face para detectarem faces em imagens.

Por outro lado, as abordagens baseadas em aparência consideram o problema de detecção de faces como um problema de classificação das imagens em duas categorias: face ou não-face. Geralmente, as abordagens baseadas em aparência utilizam treinamento de Redes Neurais, criação de modelos por meio de Máquinas de Vetores de Suporte (SVM - *Support Vector Machine*) ou de outros métodos estatísticos para classificarem as imagens. Esse é o caso do trabalho apresentado por Rowley, Baluja e Kanade (1998a).

Os detectores mencionados anteriormente lidaram com o fato de que uma imagem de face possui pelo menos quatro possibilidades de variação: orientação (no plano, ou fora do

plano da imagem), iluminação, expressão facial e oclusão. Na Figura 1.1, há amostras de faces que possuem as variações mencionadas.

De modo geral, os detectores têm lidado com esses fatores de duas formas: simplificando-os ou desconsiderando-os. Um dos princípios adotados é treinar classificadores para faixas de orientação específicas. Por exemplo, são treinados classificadores para imagens de faces frontais com pequenas variações de rotação no plano (± 10 graus). A iluminação tem sido tratada por meio de pré-processamento das imagens utilizando, por exemplo, equalização de histogramas.

A variação de expressões faciais tem sido tratada de modo satisfatório por meio da generalização dos classificadores, ou seja, as imagens de treinamento contém faces que possuem diferentes expressões faciais. No entanto, não há, dentre os artigos mencionados, nenhum que lide explicitamente com o problema da oclusão. Alguns dos detectores obtém resultados razoáveis para oclusão, mas não têm como objetivo tratar esse problema.

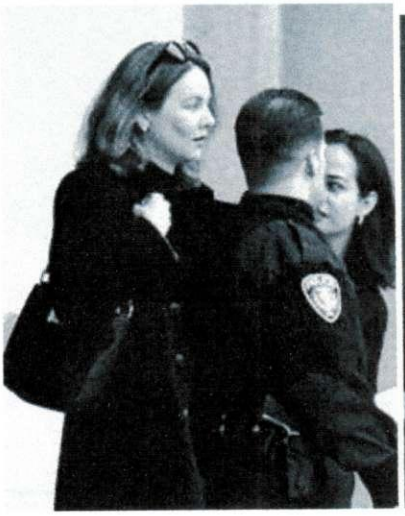
As imagens mostradas na Figura 1.1 foram obtidas das bases CMU-MIT (imagens (a), (c), (d), (e)) e FDDB (imagem (b)). Um problema que afeta a comparação dos resultados de detecção obtidos por diversas abordagens é a ausência de uma descrição sistemática da abordagem de recorte das faces usadas para treinar os classificadores. Outro problema é a ausência de padronização de uma base para ser usada como teste. Isso dificulta a comparação de abordagens diferentes porque há o risco, por exemplo, de um determinado detector ser testado usando imagens que fazem parte do conjunto de treinamento.

Para solucionar o problema apresentado, esta tese visa confirmar ou refutar as seguintes hipóteses:

- A abrangência do recorte das imagens de treinamento influencia o resultado dos classificadores;
- A abrangência da marcação da imagem detectada influencia as estatísticas de acerto dependendo do tipo de métrica usada para a avaliação; e
- A exploração da invariância por treinamento permite que a quantidade de nós de árvores de classificadores seja reduzida.

Rotação no plano

Expressão facial e oclusão



Oclusão e rotação fora do plano

Rotação fora do plano e iluminação

Iluminação e oclusão

Figura 1.1: Exemplos de variações que uma imagem de face pode apresentar.

1.3 Objetivos

O método proposto nesta tese objetiva detectar faces em qualquer grau de rotação no plano da imagem, com taxas de precisão iguais ou superiores àquelas dos detectores de faces existentes, porém utilizando menor quantidade de características e menor tempo necessário para gerar modelos em relação a métodos como aqueles propostos por Rowley, Baluja e Kanade (1998b), Viola e Jones (2004) e Huang et al. (2007). Para tanto, foi concebida uma abordagem que utiliza o compartilhamento de características entre diferentes poses das faces, permitindo a obtenção da invariância por treinamento de cascatas de características do tipo Haar treinadas com *AdaBoost*. O principal objetivo da utilização de invariância por treinamento é reduzir a quantidade de nós da árvore de classificadores e consequentemente a complexidade da detecção.

Portanto, o objetivo geral desta tese é propor, implementar e avaliar experimentalmente uma nova abordagem para detectar faces em imagens, de forma tolerante a variações de orientação no plano das imagens, utilizando uma combinação de classificadores.

Mais especificamente, os seguintes requisitos foram considerados: detecção de faces frontais com pequenas variações de pose (mudanças de orientação fora do plano da imagem), e detecção de faces frontais e de perfil com grandes variações de orientação no plano da imagem.

Como foi necessário o treinamento de dezenas de cascatas de classificadores usando AdaBoost e cada treinamento tipicamente pode levar semanas, se executado em um único computador, foi proposta e implementada uma estratégia de paralelização do método de treinamento.

Assim, esta tese possui os seguintes objetivos específicos:

- Proposição e implementação de uma abordagem de paralelização do método de treinamento de cascatas de características tipo Haar com AdaBoost;
- Proposição de uma abordagem para obter invariância por treinamento usando características do tipo Haar;

- Proposição de uma árvore de classificadores para a detecção de faces invariante à rotação;
- Realização de experimentos para a avaliação e ajustes do detector proposto;
- Realização de experimentos para comparar os resultados com aqueles obtidos a partir de outras abordagens existentes.

1.4 Relevância

Apesar de já haver na literatura especializada e na indústria várias propostas e aplicações de software para a detecção de faces em imagens digitais, ainda há muito trabalho a ser feito. Em sua maioria, os detectores de faces que clamam a detecção em tempo real realizam tal detecção em imagens de baixa resolução e não possuem robustez a todas as variações que uma imagem de face pode ter (pose, orientação, oclusão, iluminação, etc). Essa ausência de robustez total pode ser evidenciada pelas bases usadas para avaliar alguns dos detectores mais conhecidos: Rowley, Baluja e Kanade (1998b), Viola e Jones (2004) e Huang et al. (2007). A base usada para testar esses classificadores é a que foi organizada por Rowley, Baluja e Kanade (1998b), a qual não contém um subconjunto de imagens que apresente todas as variações. Por exemplo, o subconjunto que apresenta variações extremas de rotação no plano não apresenta quantidades razoáveis de imagens com oclusão ou problemas de iluminação.

Os detectores que conseguem grande robustez às variações de pose da face (Rowley, Baluja e Kanade (1998b), Jones e Viola (2003), Huang et al. (2007)) o fazem por meio da criação de uma hierarquia de classificadores, sendo cada classificador responsável por uma faixa de variação da face. Em geral, a robustez às outras variações possíveis são obtidas por meio da adição de imagens contendo características semelhantes àquelas que se deseja detectar no conjunto de treinamento.

Um dos melhores detectores conhecidos foi proposto por Huang et al. (2007). Esse detector é capaz de processar imagens de resolução 320×240 pixels a uma taxa de 3 a 4 imagens por segundo, detectando imagens em qualquer pose (obteve taxa de verdadeiro po-

sitivo de 98% para uma quantidade de falsos positivos em torno de 60, quando testado no subconjunto rotacionado da base CMU-MIT). Como será apresentado no Capítulo 5, a complexidade da árvore de classificadores dessa abordagem é bastante alta. O detector proposto nesta tese é capaz de detectar faces com invariância à rotação no plano da imagem com taxas de acerto da ordem de 95% na base CMU-MIT e possui uma árvore de classificadores de menor complexidade.

1.5 Estrutura da Tese

Esta tese está dividida do modo descrito a seguir. No Capítulo 2, serão descritos alguns termos técnicos presentes nesta introdução, tais como invariância por treinamento e abrangência de recorte e de marcação. Um conjunto de trabalhos propostos na área de detecção de faces em imagens é apresentado no Capítulo 3. A abordagem proposta nesta tese é descrita no Capítulo 4. Os resultados experimentais de testes e comparações com outros métodos para validar a abordagem proposta são apresentados no Capítulo 5.

A avaliação da abordagem de paralelização do método de treinamento de classificadores com características do tipo Haar e Adaboost é apresentada no Capítulo 6. Finalmente, no Capítulo 7, são apresentados uma breve discussão do que foi exposto na tese, as conclusões finais e proposições para trabalhos futuros. Além disso, resultados experimentais apresentados no exame de qualificação da proposta de tese são discutidos no Apêndice A.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta alguns conceitos fundamentais para o entendimento da abordagem proposta nesta tese. Na Seção 2.1, é apresentada uma explanação sobre conceitos de processamento de imagens digitais e uma introdução à detecção de faces em imagens. O conceito de extração de características para detecção de faces é apresentado na Seção 2.2. As teorias que justificam a utilização de combinação de classificadores são apresentadas na Seção 2.3. Finalmente, as variações de orientação que uma face pode apresentar e as abordagens usadas para tratar essas variações em detecção de faces são descritas na Seção 2.4. O capítulo é finalizado com a seção de conclusão.

2.1 Introdução à Detecção de Faces

Uma imagem digital geralmente é representada no formato conhecido como *raster*. Nesse formato, a imagem pode ser considerada, de modo simplificado, como uma matriz de pixels. O termo pixel provém do inglês e significa elemento de imagem (*Picture Element*). Se a imagem for representada em tons de cinza, em geral, possuirá apenas uma matriz de pixels. Se for representada em modo colorido no formato RGB, possuirá três matrizes de pixels. Cada matriz corresponde a um dos três canais de cores: vermelho (R), verde (G) e azul (B).

Os pixels das imagens podem ser acessados por meio de coordenadas, sendo que a contagem se inicia no canto superior esquerdo em direção à direita e à parte inferior. Uma das

formas mais comuns de representação de imagens cinza é utilizar 8 bits por pixel o que resulta em valores de pixel variando entre 0 e 255.

A detecção de objetos em imagens pode ser realizada por meio do escrutínio da imagem por uma janela. A cada deslizamento da janela, são extraídas informações da região, que são dadas como entrada para um classificador. Inicialmente, a janela é posicionada no canto superior esquerdo e percorre a imagem até o canto inferior direito. Ao fim de cada escrutínio, pode ser realizada uma das seguintes ações: redimensionar a imagem ou redimensionar a janela. A imagem pode ser redimensionada para menor resolução e a janela para maior resolução. O Algoritmo 1 descreve a abordagem de deslizamento de janela em que a imagem é redimensionada para menor resolução a cada escrutínio completo por uma janela de tamanho fixo.

Após a finalização do escrutínio da imagem, ocorre o processo de verificação da ocorrência de múltiplas detecções. Se houver regiões muito próximas classificadas como face, elas provavelmente correspondem a uma única face, e serão fundidas de modo a gerar apenas uma região que corresponda à média das regiões vizinhas.

No Algoritmo 1, há duas etapas fundamentais: extração de características e classificação de regiões. O modo mais simples de classificar regiões de imagens é utilizar os valores das intensidades dos pixels, mas outras características podem ser extraídas, tais como componentes obtidas por PCA (*Principal Component Analysis*), LBP (*Local Binary Patterns*) e histogramas dos níveis de cinza. Mais detalhes sobre extração de características serão apresentados na Seção 2.2. As características extraídas são usadas como entrada para classificadores previamente treinados. Alguns classificadores bastante usados são: SVM (*Support Vector Machines*), redes neurais e árvores de decisão. Nos últimos anos, é bastante comum o uso de combinação de classificadores (KUNCHEVA, 2004), a Seção 2.3 apresenta alguns conceitos de classificadores e métodos de combinação.

Entrada: imagem, escala, passo, tamanho_da_janela

Dados: array_de_faces, menor_dimensao, largura_img, altura_img, tamanho_img

menor_dimensao = 2×tamanho_da_janela;

enquanto *tamanho_img >= menor_dimensao* **faça**

  para *i = 0* **até** *i < largura_img - tamanho_da_janela* **faça**

    para *j = 0* **até** *j < altura_img - tamanho_da_janela* **faça**

      extraia características;

      classifique a região;

      se *região é face* **então**

        adicione as coordenadas ao array de faces;

      fim

      j = j + passo;

    fim

    i = i + passo;

  fim

  redimensione_a_imagem(imagem, escala);

  se *largura_img < altura_img* **então**

    tamanho_img = largura_img;

  senão

    tamanho_img = altura_img;

  fim

fim

Saída: array_de_faces

Algoritmo 1: Algoritmo de deslizamento de janela

2.2 Extração de Características

Um classificador de padrões de imagens pode ser treinado utilizando as intensidades dos pixels da imagem. Por exemplo, o detector de faces proposto por Rowley, Baluja e Kanade (1998a) utiliza esta abordagem. Porém, dependendo da resolução da imagem a partir da qual os padrões são extraídos, a dimensionalidade dos dados pode afetar o desempenho do classificador, tanto em termos de velocidade de processamento quanto na capacidade discriminatória dos dados usados. Para lidar com esses fatores, muitos autores têm processado os valores dos pixels, com o intuito de reduzir a dimensionalidade e extrair características dos dados. Segundo Bishop (2006), outro fator que justifica a extração de características é que assim as variáveis de entrada do classificador são transformadas para um novo espaço no qual se espera que o problema se torne mais fácil.

Algumas abordagens de sucesso para a extração de características de imagens são: Modelos de Razões de Faces (FRT - *Face Ratios Template*), Padrões Binários Locais (LBP - *Local Binary Patterns*), características tipo Haar e PCA. Os Modelos de Razões de Faces e os Padrões Binários Locais são descritos no Apêndice A, neste capítulo serão descritas apenas as características tipo Haar, pois apenas esse tipo de característica foi usado na abordagem proposta.

As características tipo Haar são semelhantes às wavelets de Haar e consistem simplesmente em subtrações de somas de valores de pixels, as quais são obtidas rapidamente usando a representação de imagem integral que foi proposta por Viola e Jones (2001). As características tipo Haar são exemplificadas na Figura 2.1, as quais são classificadas em três tipos: borda, linha e centro-vizinhaças.

Figura 2.1: Padrões para extração de características tipo Haar.

Como discutido anteriormente, a técnica mais usada para detectar faces é deslizar uma janela de tamanho pré-definido por sobre a imagem, a qual é redimensionada até um certo

limite. A cada passo de deslocamento, as características são extraídas da região da imagem dentro da janela e são usadas como entrada para um classificador previamente treinado para aquele tipo de padrão. O problema com a técnica de janelas deslizantes advém do tempo necessário para computar as características a cada passo.

A representação integral de imagens (VIOLA; JONES, 2001) supera o problema do tempo de processamento pré-computando todas as possíveis somas de valores de pixels antes da passagem da janela deslizante. A cada passo, somente poucos acessos à matriz pré-computada são necessários e a soma é feita em tempo constante para cada escala e posição da janela. Na Figura 2.2, há uma ilustração da abordagem de representação integral de imagens. Os números 1, 2, 3 e 4 rotulam pontos na matriz de somas de valores de pixels. As letras A, B, C e D representam regiões das quais se deseja obter os somatórios dos valores dos pixels. Como cada elemento da matriz, com exceção da primeira linha e da primeira coluna, possui o somatório dos valores acima e à esquerda, é possível obter o somatório das intensidades na região D por meio da seguinte expressão: $Soma(D) = valor_1 + valor_4 - valor_2 - valor_3$.

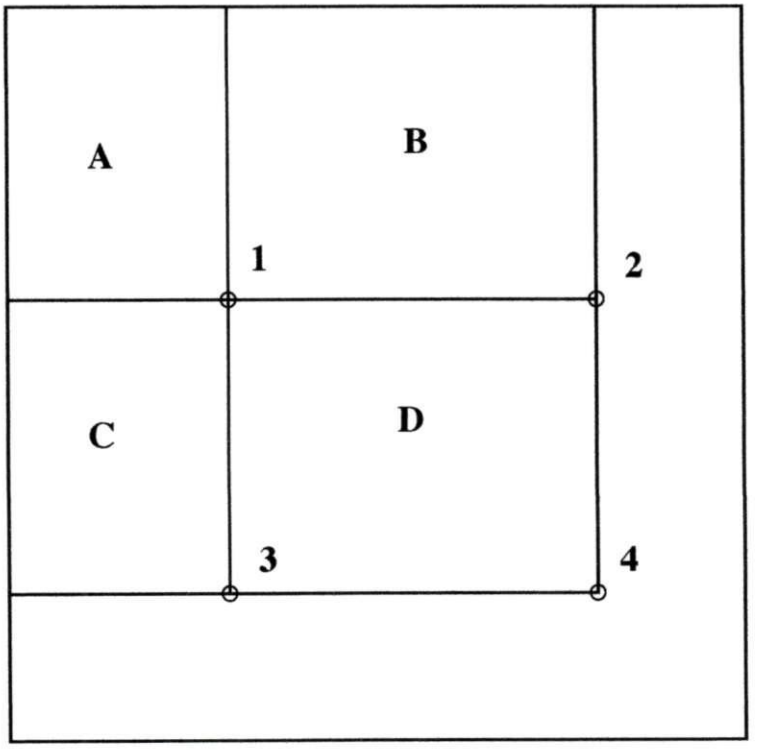


Figura 2.2: Ilustração da representação integral de imagens.

2.3 Classificadores e Métodos de Combinação

Vários classificadores foram propostos e usados em abordagens de detecção de faces, porém aqueles que obtiveram melhores resultados empregaram redes neurais artificiais, máquinas de vetores de suporte ou uma combinação de classificadores simples. Como a abordagem proposta nesta tese utilizará uma combinação de classificadores, não serão detalhados classificadores específicos e sim os métodos de combinação e os motivos pelos quais usá-los. Além disso, pode-se afirmar que as redes neurais e as máquinas de vetores de suporte já foram suficientemente bem comentados e descritos na literatura especializada. Se o leitor julgar necessário, poderá consultar as seguintes referências sobre o assunto: Haykin (1999) e Cristianini e Shawe-Taylor (2000).

Kuncheva (2004) comenta a sugestão de Dietterich (2000) de que há pelo menos três razões pelas quais uma combinação de classificadores pode obter resultados melhores do que aqueles obtidos por um classificador individual:

- Razão estatística: por vários motivos cada classificador terá capacidade de generalização diferente. Portanto, ao invés de escolher o melhor classificador, uma opção mais *segura* seria usar todos e calcular uma saída média;
- Razão computacional: alguns algoritmos de treinamento realizam subida de encosta (*hill-climbing*) ou algum tipo de busca aleatória, que podem levar a diferentes locais ótimos. A combinação de classificadores exploraria o melhor dos ótimos locais;
- Razão representacional: é possível que o espaço de classificadores considerado para o problema não contenha um classificador próximo o suficiente do ótimo. Um conjunto de classificadores simples poderia ser uma melhor opção para esse tipo de situação.

Há várias possibilidades de combinação de classificadores. Alguns classificadores retornam valores que indicam o grau de *confiança* de que a classificação está correta. Esses valores podem ser aproximados matematicamente para probabilidades, as quais podem ser aplicadas a regras de combinação de probabilidades, tais como: regra da soma e regra da multiplicação. Além disso, pode ser realizado um novo treinamento, usando as probabilidades de saída de vários classificadores.

Uma abordagem de combinação de classificadores que tem se destacado em relação a todas as outras é o *boosting* de classificadores, especificamente o *boosting* adaptativo ou *AdaBoost* (FREUND; SCHAPIRE, 1995; FREUND; SCHAPIRE, 1996; FREUND; SCHAPIRE, 1999; FRIEDMAN; HASTIE; TIBSHIRANI, 2000). Segundo Friedman, Hastie e Tibshirani (2000), o *boosting* funciona aplicando sequencialmente um algoritmo de classificação a versões ponderadas dos dados de treinamento e obtendo uma votação ponderada da sequência de classificadores produzida. De modo simplificado, o algoritmo AdaBoost (FREUND; SCHAPIRE, 1996) pode ser descrito como o pseudocódigo apresentado no Algoritmo 2.

Entrada: dados_de_treinamento

; // pesos: w_i

Dados: w_i

para $i = 1$ **até** $i <= N$ **faça**

| $w_i = \frac{1}{N}$; $i = i + 1$;

fim

para $m = 1$ **até** $m <= M$ **faça**

ajuste o classificador $f_m(x)$, usando os pesos w_i, para os dados de treinamento;

$e_m = E_w[1_{(y \neq f_m(x))}]$;

$c_m = log\frac{1-e_m}{e_m}$;

para $i = 1$ **até** $i <= N$ **faça**

| $w_i = w_i e^{c_m \times 1_{(y_i \neq f_m(x_i))}}$

fim

renormalize os pesos de modo que $\sum_i w_i = 1$;

fim

Retorne o classificador $sinal[\sum_{m=1}^{M} c_m f_m(x)]$;

Algoritmo 2: Algoritmo AdaBoost

No Algoritmo 2, são processadas M imagens das quais são extraídas N características. A cada característica é atribuído um peso, w_i, que é normalizado à cada iteração. A função E_w calcula a média de erros sobre o conjunto de treinamento usando os pesos w. O termo c_m é o resultado da normalização logarítmica de E_w. Os pesos w_i são atualizados de modo a refletirem sua importância na próxima iteração.

Da análise do algoritmo *AdaBoost* percebe-se que os classificadores que erram são enfatizados por meio da elevação dos pesos correspondentes. Tal ajuste de pesos é realizado exponencialmente. O que tornou esse método de combinação de classificadores ainda mais popular foi a abordagem proposta por Viola e Jones (2001) para a detecção de faces, na qual cada classificador está associado a uma característica. Os classificadores são *stump classifiers*, ou seja, são bastante simples. Na verdade, os classificadores simples empregados por Viola e Jones (2001) são árvores de decisão de apenas um nível (BRADSKY; KAEHLER, 2008). Esse tipo de classificador também pode ser considerado *1-rule classifier* e é uma implementação bastante elegante do princípio da *Navalha de Occam* (BLUMER et al., 1987). Cada classificador fraco está associado a uma característica, então o processo de combinação de classificadores também realiza seleção de características. Além de características que podem ser obtidas rapidamente por meio da representação de imagens integrais, da combinação de classificadores e da seleção de características usando *boosting*, os autores também acoplaram a seu arcabouço um modo sistemático de realização de *bootstrap*.

O *bootstrap* é uma abordagem de treinamento de classificadores em que versões aprimoradas dos classificadores vão sendo treinadas utilizando padrões que foram classificados incorretamente pelos classificadores treinados anteriormente. Ou seja, padrões cada vez mais difíceis vão sendo adicionados ao conjunto de treinamento. Essa abordagem havia sido empregada previamente na criação de detectores de faces, por exemplo Rowley, Baluja e Kanade (1997) a utilizaram. Mas foram Viola e Jones (2001) que popularizaram uma abordagem automática para treinar classificadores com *bootstrap*.

2.4 Detecção de Faces Multipose

Uma imagem de face pode apresentar variações que dificultam o processo de classificação de imagens. Dentre as variações possíveis, a abordagem proposta no Capítulo 4 lida com as rotações que podem ser agrupadas em: rotações no plano e rotações fora do plano. Na Figura 2.4, são apresentadas as três possibilidades de rotação da face: guinada (*yaw*), arfagem (*pitch*) e rolamento (*roll*). As poses rotuladas como rolamento correspondem a rotações no

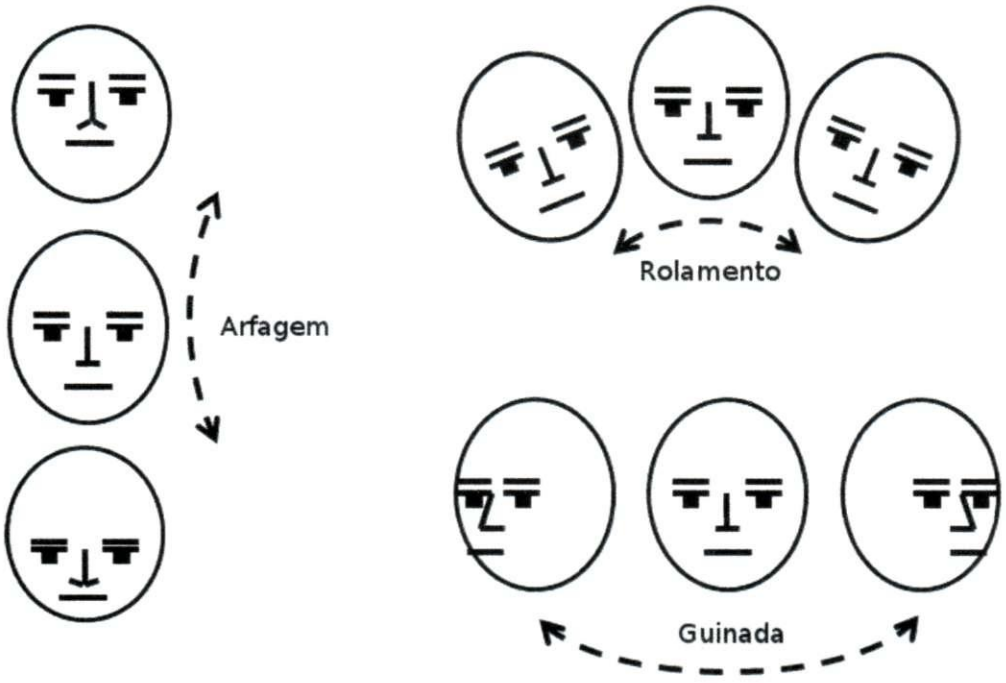


Figura 2.3: Poses da face.

plano e as rotuladas como guinada e arfagem correspondem a rotações fora do plano.

Os detectores de faces que utilizam árvores de classificadores são construídos por meio do treinamento de classificadores especialistas em faixas de ângulos de rotação. Geralmente, as variações de arfagem são consideradas apenas incluindo faces com pequenas variações de tais poses nos conjuntos de faces com variações de rolamento e guinada. Assim, as árvores de classificadores lidam explicitamente com guinada e rolamento.

Um problema enfrentado pela comunidade que trabalha com processamento de imagens de faces é a carência de bases de imagens apropriadas para treinamento e teste. Para treinamento de classificadores para detecção de faces, não há uma base que seja amplamente utilizada. Isso se deve principalmente ao fato de que há muitas possibilidades de variações que as faces podem apresentar e não há uma base que contenha todas as possibilidades. Uma base que tem sido amplamente utilizada para testar detectores de faces é a base CMU-MIT (ROWLEY; BALUJA; KANADE, 1998b), mas é uma base pequena com problemas de rotulamento e que não apresenta muitas variações de pose tipo arfagem. A

base FDDB (JAIN; LEARNED-MILLER, 2010) é uma base mais recente para avaliação de detectores de faces, que apresenta um protocolo de avaliação que permite, dentre outras possibilidades, incluir parte das imagens fornecidas no treinamento dos detectores avaliados. A maior parte dos desafios apresentados pela FDDB para os detectores de faces são relativos a rotações fora do plano. Porém, a base não apresenta quantidade satisfatória de imagens com rotações extremas como ângulos maiores que 90°.

2.5 Considerações Finais

Este capítulo apresentou conceitos básicos e importantes para o entendimento do processamento de imagens de faces, especificamente a detecção de faces. Os conceitos foram apresentados desde os fundamentos da representação de imagens em modo *raster* até a extração de características e treinamento de classificadores. Além disso, foram descritas algumas técnicas utilizadas para agilizar o processo de extração de características que são as características tipo Haar e a representação integral de imagens. Apesar de existirem vários métodos de combinação de classificadores, o método de combinação de classificadores *AdaBoost* foi detalhado neste capítulo devido à sua importância para a área de detecção de faces.

Capítulo 3

Revisão Bibliográfica

Neste capítulo, apresentam-se um levantamento e uma análise de trabalhos relacionados com esta tese, focando em trabalhos nos quais são investigados ou propostos métodos para a detecção de faces em imagens estáticas (*still images*). Yang, Kriegman e Ahuja (2002) identificaram as seguintes categorias de métodos para detecção de faces: métodos baseados em conhecimento, abordagens de características invariantes, métodos de casamento de padrões (*template matching*) e métodos baseados em aparência. Alguns métodos para a detecção de faces podem ser enquadrados em mais de uma dessas categorias.

Por outro lado, após a realização da presente revisão bibliográfica, foi percebida uma separação mais nítida entre os métodos ao se analisar: o tipo de informação passada como entrada para os métodos e a utilização (ou não) de uma estratégia de combinação de classificadores. Conforme ficará evidente a seguir, a maioria dos métodos atuais busca, principalmente, novos tipos de características ou novos métodos para combinar classificadores existentes. Desse modo, decidiu-se agrupar os métodos aqui analisados em dois grandes grupos: detecção de faces por classificador único e detecção de faces por combinação de classificadores.

3.1 Detecção de Faces Utilizando um Único Classificador

Segundo Lienhart, Kuranov e Pisarevsky (2002), o principal propósito da utilização de características, ao invés dos valores dos pixels brutos, como entrada para um algoritmo de aprendizagem, é reduzir a variabilidade intraclasse e aumentar a variabilidade interclasses. Além disso, Lienhart, Kuranov e Pisarevsky (2002) afirmam que geralmente as características extraídas codificam o conhecimento que é difícil de aprender utilizando apenas um conjunto de dados brutos.

Jesorsky, Kirchberg e Frishholz (2001) propuseram um método para a detecção de faces que utiliza a medida de distância conhecida como *Hausdorff Distance* (ALT; BEHRENDS; BLöMER, 1991) para realizar o casamento de um conjunto de pontos extraídos das bordas de uma imagem candidata a face com os pontos de um modelo de face. O sistema de detecção de faces proposto consiste de duas etapas de detecção, uma mais grosseira e outra que consiste num refinamento dos resultados da primeira. Ambas as etapas são constituídas por uma etapa de segmentação e por uma etapa de localização.

Na etapa de segmentação, são extraídas bordas utilizando o operador de Sobel (GONZALEZ; WOODS, 2010). Tais bordas são refinadas para que possam ser extraídos pontos que serão utilizados para o cálculo da medida de distância. Os autores afirmam que o sistema é robusto a mudanças de iluminação, mas a experiência em extração de bordas de imagens tem mostrado que esse processo é extremamente sensível a variações de iluminação. Outra crítica a esse artigo é que são apresentadas poucas imagens da base que foi utilizada para os testes e, além disso, as variações de iluminação, que poderiam dificultar a tarefa do detector, ocorrem primordialmente na região de fundo da imagem e não na região da face.

Aparentemente, não há nenhuma variação de iluminação que deixe apenas parte da face iluminada, diferentemente do que ocorre na base de imagens *Yale Face Database B* (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001). As imagens utilizadas para teste tinham resolução de 384×288 pixels e a área de busca foi restringida a uma região quadrada, centralizada na imagem, cujo lado corresponde à altura da imagem. Portanto, um dos fatores que influenciaram a obtenção de bons resultados (98,4% para a base XM2VTS (MATAS et al., 2000) redimensionada e 91,8% para a base Bi-

oID (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001)) foi a restrição do espaço de busca para uma região que continha praticamente apenas a face. Além disso, as imagens da base XM2VTS contêm fundo muito bem comportado, conforme pode ser observado na Figura 3.1.

Figura 3.1: Amostras de imagens de faces frontais pertencentes à base XM2VTS.

Devido ao fato de serem sensíveis a variações de iluminação, as bordas das imagens não são muito utilizadas como características para detectar faces. Um operador que tem sido amplamente utilizado para representar, detectar e reconhecer faces (HADID, 2008), que é menos sensível a variações de iluminação do que as bordas extraídas da imagem, é o LBP (*Local Binary Patterns*). Originalmente, o operador LBP foi proposto como um histograma de N níveis ($N = 2^P$) (OJALA; PIETIKÄINEN; HARWOOD, 1996). Porém, para que seja possível representar faces utilizando esse operador, é necessário que a disposição espacial dos elementos da face seja codificada no vetor de características obtido.

Para representar faces com o operador LBP, o que tem sido utilizado é a divisão da imagem de face em regiões e extração de histogramas LBP independentes para cada uma dessas regiões. Em seguida, esses vários histogramas são concatenados em apenas um vetor de características para representar a face. Alguns motivos para a popularidade do operador LBP para a detecção de faces são sua simplicidade, robustez a variações de iluminação e orientação (HADID, 2008). A simplicidade no cálculo dos códigos LBP é decorrente do fato de que são realizadas apenas subtrações entre um pixel central e uma vizinhança pré-

determinada. São essas subtrações entre um pixel central e seus vizinhos que favorecem a invariância à iluminação.

A invariância à orientação foi adicionada ao método LBP a partir da extensão do método conhecido como LBP^{ri} (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002), em que **ri** significa invariante à rotação (*rotation invariant*). Com a intenção de obter a invariância à rotação, Ojala, Pietikäinen e Mäenpää (2002) propuseram uma medida de uniformidade do código LBP. Por meio dessa medida de uniformidade, verifica-se a ocorrência de transições entre zeros e uns nas cadeias de códigos LBP. Um código LBP é chamado de uniforme se contiver no máximo duas transições de zero para um ou de um para zero. Por exemplo, os códigos 00000000 e 00001111 são uniformes, enquanto o código 10101010 não é uniforme. A extração de códigos LBP será explicada detalhadamente no Apêndice A. As características LBP podem ser obtidas de modo global, local, ou de ambos os modos. Diz-se que as características foram obtidas globalmente quando são extraídas de toda a imagem, sem que essa seja dividida em regiões de atuação do operador. Quando a imagem é dividida em regiões e o operador é aplicado para cada região individualmente, diz-se que as características foram obtidas localmente. Hadid, Pietikäinen e Ahonen (2004) propuseram um subespaço de características para representar faces para sistemas de detecção e reconhecimento de faces, tal representação incorpora características locais e globais das faces em um histograma de códigos LBP.

Uma contribuição importante da abordagem proposta por Hadid, Pietikäinen e Ahonen (2004) é a possibilidade de aplicá-la tanto para a detecção quanto para o reconhecimento de faces. Isso a diferencia de outros métodos que tratam de extração de características para detecção de faces diferentemente da extração de características para reconhecimento de faces. O vetor de características final é formado pela concatenação de dois vetores intermediários. O primeiro vetor é obtido pela divisão da imagem em 9 regiões. Em seguida, para cada região, é extraído um histograma LBP de 16 bins (quatro vizinhos e raio 1). Os 9 histogramas resultantes são concatenados para formar um histograma de 144 *bins*. O segundo vetor é obtido pela aplicação do operador $LBP^{ri}_{8,1}$ em toda a imagem, o que indica o processamento da imagem para a extração de códigos LBP utilizando operações uniformes com 8 vizinhos e raio 1. Assim, a representação final da face conterá 203 elementos. Essas caracterísi-

cas são utilizadas para treinar uma máquina de vetor de suporte (SVM) e obter um modelo para a classificação de faces. O artigo apresenta algumas estatísticas comparativas de testes realizados com as bases de imagens MIT-CMU (ROWLEY; BALUJA; KANADE, 1998a) e imagens obtidas da *World Wide Web*. As taxas de acerto são maiores que 97% e o número de falsas detecções é menor que 13, ou seja, de todas as imagens processadas apenas 13 regiões foram marcadas incorretamente como sendo face.

Características locais invariantes à escala se afiguram como a melhor opção do que características do tipo Haar devido ao alto grau de invariância a rotações no plano (*in-plane transforms*) (TOEWS; ARBEL, 2009). Foi esse alto grau de invariância que motivou a abordagem proposta por Toews e Arbel (2009), os quais propuseram um método para detectar, localizar e classificar imagens de faces quanto ao gênero (masculino/feminino). A extração de características invariantes é realizada utilizando-se a técnica SIFT.

Os modelos de aparência são aprendidos a partir da criação de modelos probabilísticos que utilizam as distribuições dos dados e regras de inferência bayesianas. Essa estratégia é conhecida como OCI (Object Class Invariant) e foi proposta, inicialmente, como um método para a detecção de faces invariante a pontos de vista (TOEWS; ARBEL, 2006). Devido às taxas de detecção regulares (em torno de 81%) e altas taxas de falsos positivos (em torno de 15%) obtidas nos experimentos de detecção de faces apresentados em Toews e Arbel (2006), os experimentos realizados por Toews e Arbel (2009) deram maior ênfase ao problema de classificação de gênero.

A maior parte das imagens utilizadas nos experimentos foram obtidas da base Color FERET Face Database (PHILIPS; MOON, 2000). Como crítica, percebe-se que, nessa base, as imagens são bem comportadas, já que contêm pouca área de fundo e portanto, poucas distrações para um detector de faces. Apesar das taxas de detecção de faces razoáveis, os resultados na classificação do gênero são superiores àqueles existentes na área, apresentando uma taxa de erro EER (*Equal Error Rate - taxa na qual os erros de aceitação e rejeição são iguais*) de 11,9% para uma variação de pontos de vista de cerca de 180 graus. Uma crítica importante ao método em questão é o fato de que a extração de características utilizando SIFT é computacionalmente cara, quando comparada a técnicas semelhantes àquela proposta por Viola e Jones (2004), que empregam uma combinação de classificadores treinados com

características do tipo Haar. O trabalho de Viola e Jones (2004) é considerado uma das referências na área de detecção de faces e será discutido em maiores detalhes na próxima seção.

Embora o processo de extração de bordas em imagens seja sensível à iluminação, alguns autores propuseram abordagens para detecção de faces que se baseiam em bordas. Anila e Devarajan (2010) propuseram um método simples para a detecção de faces, o qual suaviza a imagem, extrai bordas, normaliza os valores das bordas entre 0 e 1 e passa os valores das bordas como entrada para uma rede neural de 3 camadas do tipo Perceptron de Múltiplas Camadas (MLP - *Multilayer Perceptron*) com treinamento pelo algoritmo *Backpropagation*.

Essa abordagem utiliza um processo de varredura a partir do qual se busca por regiões retangulares contendo bordas. Apenas essas regiões são utilizadas como entrada para a rede neural. Para aumentar a velocidade de processamento, a abordagem integral para a extração de características é utilizada. A abordagem integral de representação de imagens calcula previamente os somatórios dos valores dos pixels, de forma que é possível obter a soma dos valores dos pixels para qualquer região da imagem, em tempo constante. A única informação referente à rede neural utilizada é que ela possui 4 neurônios na camada de entrada, 4 neurônios na camada escondida e uma saída. Nenhum detalhe sobre o treinamento é apresentado.

Os autores realizaram experimentos para testar a qualidade dos resultados e comparar o método proposto com um detector semelhante ao proposto por Viola e Jones (2004), mas não há comentários sobre como esse detector foi treinado. Os resultados obtidos utilizando a base BioID (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001) são razoáveis (taxa de detecção de 95.33% e taxa de falsos positivos igual a 4.5%), embora a base BioID seja muito mais simples do que outras bases utilizadas para testar detectores de faces, tais como a CMU-MIT (ROWLEY; BALUJA; KANADE, 1998b). Além disso, os autores afirmam que seu detector de faces é aproximadamente 200 vezes mais rápido que o método Adaboost, o que se deve ao fato de não serem utilizados vários níveis de classificadores e do classificador ser aplicado apenas nas regiões que contêm bordas.

3.2 Detecção de Faces por Meio de Combinação de Classificadores

Há uma tendência clara, na área de classificação de padrões, que mostra que os resultados de classificação podem ser melhorados por meio da combinação de classificadores (DUIN; TAX, 2000; JAIN; DUIN; MAO, 2000; KUNCHEVA, 2004; HAN; SIM, 2008). Nesta seção, é apresentada, inicialmente, uma visão geral de estratégias para a combinação de classificadores (Subseção 3.2.1). Em seguida, na Subseção 3.2.2, alguns trabalhos que utilizam combinação de classificadores para detecção de faces são discutidos. Finalmente, na Subseção 3.2.3, são apresentados trabalhos que utilizam um método específico para a combinação de classificadores, o AdaBoost e suas variações.

3.2.1 Fundamentos de Combinação de Classificadores

Segundo Kittler et al. (1998), as técnicas para a combinação de classificadores podem ser agrupadas em dois casos. No primeiro caso, todos os classificadores utilizam o mesmo tipo de características como padrões de entrada. No segundo caso, cada classificador utiliza um conjunto de padrões de entrada diferente dos demais. No tocante à forma de combinação de classificadores, Kittler et al. (1998) consideram as seguintes possibilidades:

- Regra do produto: a probabilidade de um determinado padrão testado pertencer a uma classe dada é igual ao produto das probabilidades de saída de todos os classificadores;
- Regra da soma: a probabilidade de um determinado padrão testado pertencer a uma classe dada é igual à média aritmética das probabilidades obtidas pelos classificadores individuais;
- Regra do máximo: a probabilidade de um determinado padrão testado pertencer a uma classe dada é igual à maior probabilidade obtida pelos classificadores;
- Regra do mínimo: a probabilidade de um determinado padrão testado pertencer a uma classe dada é igual à menor probabilidade obtida pelos classificadores;

- Regra da mediana: a probabilidade de um determinado padrão testado pertencer a uma classe dada é igual à mediana resultante da ordenação das probabilidades obtidas pelos classificadores individuais;

- Regra do voto majoritário (ou consenso): a probabilidade de um determinado padrão testado pertencer a uma classe dada é igual à média aritmética das probabilidades dos classificadores que concordarem em maioria.

Em seu artigo, Kittler et al. (1998) apresentam análises das técnicas de combinação de classificadores aplicadas a uma série de experimentos de reconhecimento de faces, voz e caracteres manuscritos. O principal resultado dessa análise é que o desempenho de classificação utilizando a regra da soma ultrapassou todas as outras técnicas. Duin e Tax (2000) realizaram, também, uma série de experimentos com regras de combinação de classificadores. Os resultados apresentados por Duin e Tax (2000) apresentam uma diferença importante em relação aos resultados de Kittler et al. (1998): nenhuma das regras de combinação se mostrou melhor do que as outras, de modo geral. Porém, deve-se ressaltar que as regras de combinação analisadas por Duin e Tax (2000) foram diferentes, a saber:

- Combinação paralela de classificadores do mesmo tipo para diferentes tipos de características;

- Combinação de classificadores diferentes para características do mesmo tipo;

- Combinação de classificadores fracos.

Os experimentos realizados por Duin e Tax (2000) utilizaram, para classificação, 2000 números escritos à mão. Os classificadores testados foram: Normal Bayes, Árvores de Decisão, Redes Neurais e SVMs. Para treinar esses classificadores foram extraídos os seguintes tipos de características: coeficientes de Fourier, correlações de perfil, coeficientes de Karhunen-Loève, médias dos pixels em janelas 2×3, momentos de Zernike e 6 características morfológicas.

Como resultados importantes das análises dos experimentos, pode-se citar o fato de que os classificadores combinados, tendo como entradas características diferentes, obtiveram

melhor resultado do que os classificadores que foram combinados tendo os mesmos tipos de características como entrada. Além disso, os melhores resultados de classificação foram alcançados pelas combinações de classificadores diferentes que tinham como entrada características diferentes. Até o momento, foi realizada uma introdução sobre combinação de classificadores e os principais motivos pelos quais é importante usá-la. Nesta seção, serão abordados trabalhos que empregam combinação de classificadores para detecção de faces, a seguir.

3.2.2 Combinação de Classificadores para Detecção de Faces

Uma heurística conhecida em detecção de faces é que a intensidade da região entre os olhos é mais clara do que a região dos olhos em uma imagem de face. Utilizando esta heurística como etapa de pré-processamento, Ramirez e Fuentes (2005) propuseram um método para a detecção de faces. O sistema de detecção é composto pela combinação de cinco classificadores, a saber: Naive Bayes, SVM, Voted Perceptron, Indução de regra $C4.5$ e Rede Neural Artificial do tipo Perceptron de Múltiplas Camadas (MLP - *Multilayer Perceptron*). Uma contribuição importante desse método é a utilização dos histogramas das intensidades dos pixels das imagens como entrada para os classificadores. Além dos histogramas das imagens, o sistema utiliza, em um segundo estágio, características semelhantes àquelas propostas por Sinha (2002). Foram realizados experimentos para testar a precisão da combinação de classificadores utilizando a base de imagens BioID (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001). O melhor resultado obtido foi 93,23% de taxa de acerto, com uma média de 1,5 falsos positivos por imagem. Apesar da taxa de acerto promissora, o artigo não menciona quantas sub-imagens foram testadas. É necessário saber quantas sub-imagens foram testadas porque as abordagens de detecção de faces geralmente realizam uma varredura de janela deslizante na imagem. A cada passo de deslizamento da janela, são extraídas características que são submetidas a um classificador. Portanto, dependendo do passo de deslocamento e da escala da janela, a quantidade de sub-imagens pode variar bastante.

Dentre os métodos de detecção de faces que utilizam combinação de classificadores treinados com redes neurais, o mais conhecido e que possui os melhores resultados é o do detector de faces invariante à rotação que foi proposto por Rowley, Baluja e Kanade (1998b).

O núcleo desse detector é formado por três redes neurais. A primeira rede neural é chamada de *router* e serve para determinar o ângulo de rotação da imagem candidata a face. A rede *router* possui três camadas, a primeira com 400 neurônios (correspondendo à quantidade de pixels da imagem de resolução 20×20), a camada escondida contém 15 neurônios e a camada de saída possui 36 neurônios (correspondendo a variações de $0°$ a $360°$ com incremento de $10°$).

Após a determinação do ângulo de rotação, a região candidata a face é rotacionada para que fique na posição vertical sendo, em seguida, repassada como entrada para duas redes neurais treinadas independentemente (ou seja, com inicializações de pesos diferentes) para classificar esse tipo de imagem. O resultado das duas redes neurais é combinado por um operador lógico **E**, ou seja, se as duas obtiverem o mesmo resultado de classificação a imagem é considerada face. Os autores ressaltam um problema comum durante o treinamento de classificadores para a tarefa de detecção de faces, que é encontrar uma amostra representativa de imagens que não contém faces. Para resolver esse problema, os autores sugerem o uso de *bootstrapping* (fortalecimento do classificador por meio de inclusão no conjunto de treinamento de imagens classificadas incorretamente durante a etapa de testes) e explicam como o aplicaram.

Para validar a abordagem proposta, são realizados experimentos com duas bases de imagens chamadas de *upright test* (contendo 130 imagens e 511 faces com rotação de $+ - 10°$) e *rotated test* (contendo 50 imagens com 223 faces com ângulos de rotação maiores que $10°$), respectivamente. A base de imagens *upright test* é a mesma utilizada no artigo Rowley, Baluja e Kanade (1998a), o qual propõe um detector de faces frontais que usa um método semelhante ao proposto em Rowley, Baluja e Kanade (1998b). Os resultados dos experimentos são mostrados em uma tabela destacando as melhores taxas de verdadeiros positivos seguidas das respectivas quantidades de falsos positivos. Segundo os autores, o melhor par verdadeiros positivos / falsos positivos é 85,7% e 15, respectivamente. No Capítulo 5, a base de imagens composta pelas duas comentadas anteriormente que é chamada de CMU-MIT será usada para avaliar a abordagem proposta.

A utilização de SVM tem se mostrado uma boa alternativa para a detecção de faces. Porém, as imagens de faces apresentam uma grande variabilidade devido a mudanças de ilumi-

nação, posições e, principalmente, por possuírem flexibilidade em relação ao posicionamento e forma de alguns de seus elementos constituintes (por exemplo, boca e olhos). Devido à grande variabilidade das imagens de faces, é necessária uma grande quantidade de amostras para que seja possível treinar adequadamente uma SVM. Além disso, as SVM sofrem de um problema de complexidade computacional: o treinamento de uma SVM frequentemente requer a resolução de um problema de otimização que varia quadraticamente em relação ao número de amostras de treinamento (MEYNET et al., 2005). Para tratar esse problema de complexidade computacional durante o treinamento de SVM, Meynet, Popovici e Thiran (2005), Meynet et al. (2005) propuseram dois métodos que dividem o conjunto de amostras e treinam SVM separados combinando suas saídas.

O método proposto por Meynet, Popovici e Thiran (2005) analisa o comportamento dos resultados de classificação de imagens de faces, utilizando SVM treinados com conjuntos de imagens divididos de dois modos distintos: aleatoriamente e por meio de agrupamento (*clustering*). As características utilizadas para treinar as SVM são os componentes principais extraídos por meio de PCA (SMITH, 2002) (*Principal Component Analysis*) e que mantém 85% da variância total. Os experimentos utilizaram 8256 imagens de faces e 14000 de não-faces para treinamento e 7822 imagens de faces e 900000 imagens de não-faces para validação, todas as imagens tinham resolução de 20×15 pixels.

As imagens utilizadas foram extraídas das bases XM2VTS (MATAS et al., 2000) e BANCA (BAILLY-BAILLIÉRE et al., 2003). O conjunto de treinamento foi dividido em 5 subconjuntos e esses subconjuntos foram utilizados para treinar 5 SVM diferentes. As saídas dessas SVM foram utilizadas como entrada para uma SVM final. Os resultados experimentais mostraram que a divisão do conjunto de treinamento aumentou a velocidade de processamento durante o treinamento e, também, aumentou a precisão dos resultados. Outro resultado importante é que a quantidade de vetores de suporte total das SVM treinadas com subconjuntos é menor do que a quantidade de vetores de suporte de uma única SVM treinada com todo o conjunto. Um fator importante que deve ser observado quanto aos resultados experimentais é que, em nenhum dos experimentos, a taxa de falsos positivos se aproximou da taxa considerada boa pela comunidade de detecção de faces: 1×10^{-6}. A melhor taxa de verdadeiros positivos com a combinação dos classificadores ficou em torno de 96%.

Um estudo experimental semelhante ao anterior foi apresentado por Meynet et al. (2005). As principais diferenças entre o método proposto por Meynet, Popovici e Thiran (2005) e aquele proposto por Meynet et al. (2005) são as técnicas utilizadas para combinar os classificadores e as bases de imagens utilizadas. Meynet et al. (2005) analisaram o comportamento da combinação de SVM utilizando regras de combinação de classificadores, a saber: regra do produto, regra da soma, regra do máximo, regra do mínimo, regra da mediana e regra do voto majoritário. Para realizar os experimentos, foram extraídas cerca de 8000 imagens de faces para treinamento e testes das seguintes bases: BANCA (BAILLY-BAILLIÉRE et al., 2003), XM2VTS (MATAS et al., 2000), BioID (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001) e FERET (PHILIPS; MOON, 2000).

As imagens de faces foram redimensionadas para 19 $\times$ 19 pixels e passaram por um processo de análise de componentes principais (PCA) tendo como resultado a redução da dimensionalidade para 15 componentes dos 361 iniciais. O conjunto de imagens de treinamento foi particionado aleatoriamente e submetido a um processo de treinamento utilizando validação cruzada tendo como função de *kernel* uma RBF (*Radial Basis Function*). Os resultados foram semelhantes aos apresentados por Meynet, Popovici e Thiran (2005) e enfatizaram os benefícios de se dividir os dados de treinamento. Essa divisão reduz a complexidade da etapa de treinamento e obtém modelos com menos vetores de suporte. A importância dos modelos esparsos está relacionada a sua maior capacidade de generalização.

A abordagem de decomposição do conjunto de treinamento para SVM em subconjuntos menores já havia sido proposta por Osuna, Freund e Girosi (1997). Nessa abordagem, o espaço de vetores é dividido em regiões e várias etapas de processamento são realizadas nessas regiões separadamente. Contudo, o método proposto por Osuna, Freund e Girosi (1997) não utiliza combinação de classificadores e utiliza um vetor de características contendo os valores dos pixels das imagens de faces em tons de cinza de resolução 19 $\times$ 19 pixels. Mesmo utilizando tantos elementos nos vetores de características, o tempo necessário para gerar os modelos utilizando mais de 50000 imagens foi de 5 horas, uma velocidade de processamento rápida, considerando-se que foi utilizada uma *SPARC Station 20*. O detector foi testado tendo como entrada imagens da base CMU (SCHNEIDERMAN; KANADE, 2000b; ROWLEY; BALUJA; KANADE, 1997; ROWLEY; BALUJA; KANADE, 1998b). Para o

conjunto de teste *Test Set A*, o método proposto obteve 97,1% de acerto e apenas 4 falsos positivos, enquanto para o conjunto *Test Set B* a taxa de acerto foi de 74,2% e 20 falsos positivos. Também é mostrado no artigo uma comparação com a abordagem proposta por Sung e Poggio (1998), deixando evidente a equivalência dos resultados das duas abordagens.

3.2.3 Combinação de Classificadores Usando *Boosting*

Embora a abordagem proposta por Rowley, Baluja e Kanade (1998b) utilize uma combinação simples de classificadores, ela ainda possui um problema relacionado ao fato de que grandes quantidades de janelas que não são faces terem de passar por todos os classificadores. Visando descartar a maior quantidade de não faces possível antes que elas sejam processadas por todos os classificadores, Viola e Jones (2001) propuseram um dos métodos que mais tem despertado o interesse da comunidade que trabalha com detecção de faces. Esse método alia um extrator rápido de características simples (representação integral de imagens para extração de características do tipo Haar (PAPAGEORGIOU; OREN; POGGIO, 1998) a um método de combinação de classificadores fracos que, quando combinados, produzem um classificador bastante preciso.

Esse método de combinação de classificadores é chamado de Adaboost, acrônimo para *boosting* adaptativo (*Adaptive Boosting*) (FREUND; SCHAPIRE, 1999). O algoritmo Adaboost realiza a combinação ponderada de classificadores fracos, na qual os pesos dos classificadores intermediários vão sendo adaptados a cada iteração de modo que os pesos das amostras classificadas incorretamente sejam enfatizados e o algoritmo seja forçado a empregar maior esforço na aprendizagem dos exemplos mais difíceis. Freund e Schapire (1999) afirmam que há várias semelhanças entre o método Adaboost e as SVMs. No entanto, há uma diferença fundamental, relacionada à complexidade computacional.

Enquanto a complexidade computacional nas SVMs é quadrática, no Adaboost a complexidade computacional é linear. A SVM trata sua complexidade quadrática por meio da utilização de métodos de *kernel*, enquanto a abordagem de *boosting* utiliza busca gulosa (*greedy search*). A popularidade desse método, que utiliza o par Adaboost e imagem integral, decorre do poder de generalização proporcionado por métodos de combinação de classifica-

dores. No caso do Adaboost, vários classificadores que possuem uma taxa de acerto baixa para falsos positivos (menor ou igual a 1%) e melhor possível para verdadeiros positivos (maior ou igual a 51%) são combinados em cascata para gerar um classificador forte. Além disso, o método da imagem integral para a representação de imagens aumenta a velocidade de processamento do processo de extração de características.

O método proposto por Viola e Jones (2001) foi expandido por Lienhart, Kuranov e Pisarevsky (2002) em dois aspectos principais: (1) a adição de características do tipo Haar com diferentes orientações, representações diagonais, centro-vizinhaças e (2) a comprovação experimental de que o método de combinação de classificadores *Gentle AdaBoost* (FRIEDMAN; HASTIE; TIBSHIRANI, 2000) é melhor para combinar esses tipos de características do que os métodos *Discrete AdaBoost* (SCHAPIRE; SINGER, 1998) e *Real AdaBoost* (FREUND; SCHAPIRE, 1999). Esses métodos de classificação diferem apenas no algoritmo de aprendizagem. Os experimentos foram realizados utilizando a base de imagens CMU-MIT (ROWLEY; BALUJA; KANADE, 1998b). Duas contribuições importantes dessa análise experimental são a comprovação de que o uso dos novos tipos de características reduziu a taxa de falsos positivos em torno de 10% e a indicação experimental de que 20×20 pixels seria a dimensão mínima para o recorte de regiões de faces empregadas no treinamento de classificadores, sem perdas substanciais nas taxas de classificação correta.

Posteriormente, Jones e Viola (2003) expandiram a abordagem proposta em Viola e Jones (2001) para lidar com faces em múltiplos pontos de vista. A abordagem adotada pelos autores é semelhante a que foi proposta por Rowley, Baluja e Kanade (1998b), ou seja, uma cascata de classificadores é treinada para estimar a pose da face candidata e, em seguida, a imagem é aplicada a uma cascata de classificadores específica para àquela faixa de rotação. Foram treinadas 12 cascatas de classificadores cobrindo intervalos de 30 graus de rotação no plano da imagem.

A primeira cascata, responsável pela determinação do ângulo de rotação no plano, foi treinada usando 4000 imagens de faces. Essa pequena quantidade de imagens foi imposta, segundo os autores, devido ao elevado consumo de memória e tempo de processamento pela estratégia adotada. As cascatas específicas para cada ângulo de rotação no plano foram treinadas usando 8356 imagens de faces e mais de 100 milhões de imagens de não faces.

Todas as imagens utilizadas eram de resolução 24×24 pixels. Cada cascata especialista em uma faixa de rotação atingiu 35 estágios ao final do treinamento.

Em seguida, Jones e Viola (2003) realizaram um treinamento semelhante para imagens de faces em perfil. Foram utilizadas 2868 imagens de faces em perfil e o mesmo conjunto de não faces utilizado para os treinamentos das cascatas especializadas em imagens frontais. Cada cascata especialista em uma faixa de rotação no plano da imagem atingiu 38 níveis ao final do treinamento. Ambos os detectores de faces (frontal e perfil) foram testados sobre a base CMU-MIT. O detector de faces frontais foi testado sobre o subconjunto *tilted* e foram processadas 10.515.781 janelas, resultando em uma taxa de verdadeiros positivos de 95% e 1345 falsos positivos (taxa de falsos positivos: 0,0127903%). O detector de faces em perfil foi testado sobre o subconjunto *profile test set* e processou 48.303.529 janelas.

Foram detectadas corretamente 83,1% das imagens de faces e incorretamente 700 janelas (o que equivale a uma taxa de 0,0014492% de falsos positivos). Os autores ressaltam que seu detector de faces processa uma imagem de resolução 320×240 pixels em 0,12 segundos em um processador Pentium 4 de 2,8GHz. Esse detector de faces é bastante rápido e permite processamento em tempo real, porém suas taxas de falsos positivos ainda estão altas, com relação à taxa objetivo de 1 erro em um milhão de testes, proposta pelos mesmos autores em (VIOLA; JONES, 2004).

Masip, Bressan e Vitrià (2005) apresentaram mais uma variação do método Adaboost para a detecção de faces. O método proposto realiza um *boosting* adaptativo, no qual as características extraídas nas cascatas anteriores não são fixadas, mas sim adaptadas, dependendo dos novos conjuntos de características que vão sendo obtidas a cada cascata. Desse modo, a cada iteração, o algoritmo é focado em amostras mais complexas, dando mais importância às amostras de maior dificuldade de classificação. Isso permite uma maior velocidade de convergência de treinamento, levando o algoritmo a atingir taxas iguais ou maiores do que o método de *boosting* tradicional, porém em um número menor de iterações.

Foram realizados experimentos utilizando tanto conjuntos de características fixas (como no método Adaboost tradicional) quanto conjuntos de características adaptadas. A adaptação das características é realizada por meio da modificação dos pesos de forma que características usadas para classificar amostras mais difíceis obtém pesos mais elevados. Nos experimentos

de teste, foram utilizadas 2000 imagens de faces e 26000 imagens de não faces, obtidas da base XM2VTS (MATAS et al., 2000). O uso de características adaptáveis aumentou a taxa de verdadeiros positivos em 1,4%, porém obteve uma taxa de falsos positivos de 0,2%, considerada alta para problemas de detecção de faces. Além disso, o tempo necessário para processar uma imagem em um processador Pentium 4 de 2,4GHz utilizando Matlab 6.0 foi de 12 segundos (muito lento, se for necessária a detecção em tempo real).

Li e Zhang (2004) propuseram um novo método de aprendizagem baseado em *boosting*, o *FloatBoost*. A principal diferença entre o *FloatBoost* e o *AdaBoost* é que o *FloatBoost* realiza um passo retroativo (*backtrack*), o qual elimina os classificadores fracos, que não são importantes em termos de taxa de erro, do conjunto de classificadores aprendidos. Na etapa de exclusão condicionada, o método *FloatBoost* remove os classificadores mais fracos cuja exclusão implique uma diminuição da taxa de erro geral. O processo de exclusão é repetido, até que um limiar de erro seja atingido.

Uma implicação direta da exclusão dos classificadores mais fracos é a redução da quantidade de características necessárias para classificação. Para tratar o problema da detecção de faces a partir de múltiplos pontos de vista, os autores propõem uma partição do conjunto de imagens em subconjuntos de ponto de vista e um método simples-para-complexo para agilizar o processamento. Esse método simples-para-complexo é implementado por meio de uma pirâmide de classificadores que generaliza a estrutura em cascata proposta por Viola e Jones (2004). Para cada estágio da pirâmide, são treinados classificadores específicos para faixas de rotações específicas da face. Por exemplo, as rotações no plano (*in-plane rotations*) variam de -45 a $+45$ graus, com passo de 30 graus.

Com o intuito de comparar os resultados do método Floatboost com aqueles obtidos a partir do método Adaboost, três tipos de experimentos são realizados. O primeiro experimento, que almejava comparar apenas as taxas de erro quando da fixação da taxa de acerto em 99,5%, foi realizado utilizando 5000 imagens de faces e 5000 de não-faces para treinamento e 1000 imagens de faces e 1000 de não-faces para teste. O primeiro experimento demonstrou claramente a força do Floatboost em atingir taxas iguais ou superiores ao do Adaboost utilizando uma quantidade de características até três vezes menor.

O segundo conjunto de experimentos comparou os métodos Adaboost e Floatboost uti-

lizando as imagens da base MIT+CMU Test Set (ROWLEY; BALUJA; KANADE, 1998b). Mais uma vez, o *FloatBoost* se mostrou superior, com taxas até 10% maiores e utilizando metade da quantidade de características utilizadas pelo Adaboost. O último experimento foi realizado sobre uma base de 25000 imagens de faces em múltiplos pontos de vista, geradas a partir das 6000 imagens de treinamento, por meio de rotações e deslocamentos aleatórios. Alcançou-se uma taxa de acerto de 94% e uma taxa de falsos positivos em torno de 4×10^{-6}. Apesar da afirmação feita pelos autores de que o sistema de detecção seria o primeiro no mundo a realizar a detecção de faces em múltiplos pontos de vista e em tempo real (o mesmo leva $200ms$ para processar uma imagem de 320×240 pixels), há algumas críticas em relação à abordagem subjacente ao sistema, conforme discutido a seguir.

Durante o processo de varredura da imagem, cada janela analisada deve ser re-escalada para o tamanho padrão de treinamento: 20×20 pixels. Além disso, para tratar o problema das orientações das faces, uma série de classificadores teve que ser treinada para cada conjunto de variações de orientação. Neste caso, vale ressaltar a extrema lentidão de treinamento (que pode chegar a semanas de processamento e, segundo os autores, é 5 vezes mais lento do que o método Adaboost convencional) para se obter os conjuntos de características para cada combinação de classificadores fracos.

Outra variação do método Adaboost para a detecção de faces foi proposta por Wu et al. (2004) e foi chamada de *Real Adaboost*. A principal diferença entre esse método e o Adaboost original é que o limiar de ativação dos seus classificadores fracos não são discretos (-1 e 1 no caso do Adaboost). Os autores propuseram uma tabela de consulta(*Look Up Table* - LUT) de classificadores fracos na qual os valores das características de Haar são normalizados entre 0 e 1 e esta escala de valores é dividida uniformemente em n sub-escalas. Desta forma, os limiares de ativação dos classificadores de um estágio anterior podem ser utilizados como fatores de confiança para geração do estágio posterior.

Uma característica importante do detector proposto por Wu et al. (2004) é o agrupamento das possibilidades de variação de pontos de vista das faces no plano e fora do plano. As variações no plano são divididas em 12 possibilidades. Assim, 12 classificadores seriam treinados para faixas de 30 graus de variação no plano. Porém, como as características do tipo Haar podem ser invertidas horizontalmente e rotacionadas de 90 graus, apenas 8 clas-

sificadores tiveram que ser treinados para variações no plano. As variações de pontos de vista fora do plano foram agrupadas em 5 categorias: perfil esquerdo, meio perfil esquerdo, frontal, meio perfil direito e perfil direito.

Como o método utiliza uma abordagem de geração de classificadores para faixas de variações de ângulos no plano semelhante a que foi proposta por Rowley, Baluja e Kanade (1998b), os autores afirmam que sua proposta obtém resultados superiores, devido ao fato de utilizar uma heurística de agrupamento baseada nos limiares de confiança mencionados anteriormente. O detector foi testado sobre a base CMU-MIT. No subconjunto de imagens frontais [1], obteve 96,5% de acerto para uma quantidade de falsos positivos igual a 213. Quando testado no subconjunto de imagens em perfil (208 imagens contendo 441 faces), obteve 91.3% de acerto para uma quantidade de falsos positivos igual a 415.

As abordagens que dividem as possibilidades de variação da pose da face em faixas fixas como a apresentada anteriormente são muito restritivas e podem impor uma carga de complexidade muito alta para os classificadores, especialmente quando estão sendo consideradas imagens de faces que geram dúvida quanto a qual faixa de variação pertencem. Para lidar com tal problema, Huang et al. (2005) propuseram uma abordagem para a detecção de faces que utiliza *Vector Boosting*. A principal diferença do *Vector Boosting*, em relação ao Adaboost, é que ele gera como saída um vetor de valores indicando a quais classes o padrão de entrada pode pertencer. Então, se características extraídas de uma face que não possui pose frontal forem submetidas a um classificador desse tipo, treinado adequadamente, sua saída pode indicar que a classe poderia ser perfil ou meio-perfil.

Assim, em uma árvore de classificadores, uma abordagem simples-para-complexo pode ser utilizada de forma que, a cada nível, os padrões sejam submetidos a classificadores cada vez mais especializados na possível pose a qual a face submetida pertence. Huang et al. (2005) utilizam essa idéia para gerar seu detector de faces. Além disso, os autores utilizam uma LUT similar a que foi proposta por Wu et al. (2004), de forma que os limiares dos classificadores sejam normalizados entre 0 e 1. O detector foi testado usando o subconjunto de imagens de perfil da base CMU-MIT (208 imagens contendo 441 faces) e obteve 95,7%

[1] 130 imagens contendo 507 faces, há divergências entre diversos autores quanto à quantidade exata de faces nessa base.

de acerto para uma quantidade de falsos positivos igual a 470.

Huang et al. (2007) propuseram uma extensão da abordagem proposta por Huang et al. (2005) capaz de detectar faces, com rotações arbitrárias no plano ou fora do plano, em imagens ou vídeos. Para alcançar altas taxas de acerto, a velocidades de processamento aceitáveis, os autores usaram os seguintes métodos inovadores: uma estrutura de detecção em árvore chamada de *Width-First-Search* (WFS), o algoritmo *Vector Boosting*, um algoritmo de aprendizagem fraco baseado em partição de domínio, características esparsas em espaço granular (*Sparse Granular Features*) e uma heurística de busca para características esparsas.

O uso do método WFS é justificado por Huang et al. (2007) a partir do fato de que esse tipo de busca permite que mais de um caminho seja seguido na árvore de busca. Se, ao final do processo de caminhamento na árvore de busca, mais de uma folha for selecionada como candidata, a que obtiver maior grau de confiança[2] será classificada como face. Devido à necessidade de um classificador que permita a geração de saídas não totalmente excludentes, os autores propõem o uso de *Vector Boosting*. Para melhor entendimento do problema, considere-se que o primeiro nó da árvore se ramifica para outros 4 nós, tendo como rótulos da esquerda para a direita as seguintes classes: perfil-esquerda, frontal, perfil-direita e não-face. Então, uma saída possível seria: (0,1,1,0). Esta saída indica que o processo de classificação será passado para os nós inferiores treinados para face frontal e para face em perfil para a direita. Em seguida, esses classificadores determinariam suas saídas e a cada nível da árvore o processamento seria realizado por classificadores mais especializados. Esse tipo de classificação em árvore é, segundo os autores, mais flexível do que outras abordagens propostas na literatura.

Huang et al. (2007) criticam o uso apenas de valores de pixels ou de características do tipo Haar para detecção de faces e sugerem o uso de características granulares. Essas características são obtidas por meio da combinação linear de somas de valores de pixels obtidas de regiões correspondentes de imagens redimensionadas a partir da imagem original. Como a quantidade de possibilidades de características granulares diferentes é muito grande, os autores propõem o uso de uma busca baseada em heurística, a qual favorece as características granulares esparsas que possuam perdas menores no treinamento e complexidade baixa.

[2]probabilidade de a região candidata ser face

Alguns experimentos foram realizados, utilizando a base CMU-MIT (ROWLEY; BALUJA; KANADE, 1998b), para testar e comparar os resultados da abordagem proposta por Huang et al. (2007) com as abordagens propostas por Rowley, Baluja e Kanade (1998b), Schneiderman e Kanade (2000b), Jones e Viola (2003), Wu et al. (2004) e Huang et al. (2005). A abordagem proposta obteve melhores resultados do que todas as demais citadas. Quando o detector proposto foi testado sobre as imagens de perfil, para uma taxa de detecção acima de 95%, a quantidade de falsos positivos ficou em torno de 250 (a quantidade de janelas analisadas não foi mencionada). O detector que obteve os resultados mais próximos desses valores foi o apresentado por Huang et al. (2005) que obteve uma taxa de detecção acima de 95% com uma quantidade de falsos positivos em torno de 475 (a quantidade de janelas analisadas não foi mencionada).

Os resultados obtidos pela abordagem proposta por Huang et al. (2007) sobre o subconjunto da base CMU-MIT que contém imagens de faces frontais também foram melhores do que aqueles obtidos pelos demais. O detector obteve uma taxa de detecção acima de 97% com uma quantidade de falsos positivos menor que 100. Os autores afirmam que seu detector, quando variações extremas de pose estão desabilitadas, atinge uma velocidade média de 10 quadros por segundo se executado em sequências de vídeo contendo imagens de resolução 320 $\times$ 240 pixels. Quando todas as variações de pose estão habilitadas, a velocidade é reduzida para uma média de 4 quadros por segundo para imagens com as mesmas dimensões citadas anteriormente. Além disso, é importante destacar que o classificador completo é composto por 234 nós (cada nó corresponde a uma combinação de classificadores fracos), dispostos em 18 camadas.

Algumas abordagens para a detecção de faces propõem a combinação de outros tipos de características com o método consolidado Adaboost com imagem integral. Esse é o caso da abordagem proposta por Meynet et al. (2007) para o rastreamento de faces e a estimação da posição da cabeça. Um problema na combinação AdaBoost com imagem integral é a necessidade de treinamento com imagens em cada faixa de rotação que deseja detectar. Para tratar esse problema, Meynet et al. (2007) propuseram a criação de uma árvore de classificadores combinados por meio de *AdaBoost*, mas que utilizam tanto características do tipo Haar (VIOLA; JONES, 2001) quanto características obtidas por filtros gaussia-

nos (GONZALEZ; WOODS, 2010).

As características do tipo Haar e os filtros gaussianos são complementares. Enquanto as características do tipo Haar são processadas rapidamente, os filtros gaussianos são processados em menor velocidade. Porém, as do tipo Haar não são seletivas o suficiente para distinguir as posições das faces, enquanto os filtros gaussianos o são. É essa complementariedade que justifica a utilização da combinação em árvore (MEYNET et al., 2007), na qual as características do tipo Haar são extraídas nos estágios iniciais, excluindo cerca de 95% das janelas testadas. Os modelos de faces foram treinados utilizando cerca de 50000 imagens de faces e 500000 imagens de não-faces.

O detector foi testado em um vídeo de resolução 320×240 pixels de dois modos: quadro-a-quadro e por rastreamento (utilizando um algoritmo de condensação - *condensation algorithm* (ISARD; BLAKE, 1998)). O algoritmo de rastreamento detectou faces a 23,45 quadros por segundo, o algoritmo quadro-a-quadro detectou faces a 6,36 quadros por segundo. Um fator que não foi enfatizado no artigo é quais vídeos foram utilizados para realizar os testes, o que impossibilita comparações com outros métodos. Outro fator não mencionado é como a medição da taxa de acerto (93%) foi efetuada. Apesar de a árvore de detecção conter 60 classificadores, o detector obteve bom desempenho em termos de velocidade de processamento.

A detecção de faces é umas das etapas iniciais em processamento de imagens de faces, tais como reconhecimento de faces e reconhecimento de expressões faciais. Chen, Huang e Fu (2007) propuseram um sistema para o reconhecimento de expressões faciais que é capaz de diferenciar entre 7 expressões diferentes: neutro, felicidade, raiva, tristeza, surpresa, desgosto e medo. Nesta revisão bibliográfica, apenas será detalhado o método de detecção de faces proposto como etapa anterior ao reconhecimento de expressões faciais. O algoritmo de detecção de faces proposto é chamado de algoritmo de aprendizagem de *boost* híbrido (*hybrid-boost*) para a detecção de faces em múltiplas poses.

Esse algoritmo é chamado de híbrido por que utiliza dois tipos de características durante o *boosting*: características do tipo Haar e características de Gabor. Uma característica híbrida é definida como $x = (t, x, y, p_1, p_2)$, em que t define o tipo de característica (Gabor ou Haar), x e y definem a posição da característica na imagem e p_1 e p_2 correspondem respectivamente

à largura e à altura, para características do tipo Haar, e à orientação e à frequência, para características de Gabor. A aprendizagem por *boost* híbrido utiliza uma função suave de decisão para classificadores fracos.

A função suave é obtida por associação entre os n intervalos (*bins*) de um histograma construído a partir das respostas de classificadores fracos para as características x definidas anteriormente. Então, por meio de probabilidades *a posteriori*, as características híbridas mais discriminativas são selecionadas. Os experimentos foram realizados utilizando imagens obtidas das bases FERET (PHILIPS; MOON, 2000) (3000 imagens de faces) e fotos pessoais dos autores (87 imagens). A taxa de acerto foi de 99,24% e a taxa de falsos positivos foi de 1,72%. Apesar da alta taxa de acerto, a taxa de falsos positivos está muito alta para o problema de detecção de faces pois, segundo Viola e Jones (2004), para que um detector de faces possa ter uso prático em aplicações reais a taxa de falsos positivos deve estar em torno de 10^{-6}. Além disso o, as imagens da base FERET são muito bem comportadas[3] a fim de poderem ser utilizadas por sistemas de detecção de faces. É importante acrescentar que os autores não apresentaram nenhuma comparação do método proposto com outros métodos existentes.

Rodriguez (2006) propôs, em sua tese de doutorado, um método que utiliza características LBP em conjunto com Adaboost para detectar faces, no qual o operador LBP é utilizado para extrair características de textura de imagens de faces. Essas características são utilizadas para treinar uma cascata de classificadores utilizando o método Adaboost para a tarefa de detecção de faces. Além disso, são treinados classificadores para várias possíveis posições de faces e são utilizadas mais de 20000 imagens de face no processo de treinamento e teste do sistema proposto. Outra inovação no trabalho de Rodriguez (2006) é o método para a avaliação de detectores de faces proposto. O autor critica os métodos de avaliação que utilizam distâncias entre pontos encontrados e pontos marcados manualmente e propõe um método de avaliação voltado para objetivos. Por exemplo, se o objetivo do detector de faces é servir para extrair faces que serão utilizadas por um sistema de reconhecimento, então a melhor forma de avaliar o detector de faces, segundo Rodriguez (2006), é calcular as estatísticas de acerto do sistema de reconhecimento de faces utilizando o detector de faces que está

[3] As imagens possuem possuem fundo simples e não apresentam variações complexas de iluminação.

sendo avaliado.

3.3 Revisões de Literatura e Avaliações de Detectores

Nesta seção discutem-se três artigos nos quais se apresentaram revisões da literatura sobre detecção de faces ou descrições de processos de avaliação de detectores de faces. No primeiro (DEGTYAREV; SEREDIN, 2010), apresenta-se a avaliação de sete abordagens para a detecção de faces. O segundo (ZHANG; ZHANG, 2010) contém uma discussão teórica do impacto da abordagem proposta por Viola e Jones (2001) e das variações desse método. O terceiro (JAIN; LEARNED-MILLER, 2010) é um relatório técnico no qual é descrito um protocolo de avaliação para detectores de faces.

Degtyarev e Seredin (2010) avaliaram sete abordagens para a detecção de faces em nove bases de imagens manualmente rotuladas. Os detectores de faces avaliados foram:

- O detector de faces da biblioteca OpenCV[4], de código aberto e um representante da abordagem proposta por Viola e Jones (2001);
- O detector de faces SIF (KRESININ; SEREDIN, 2009), desenvolvido no Laboratório de Análise de Dados da Universidade Estadual de Tula (Rússia), que utiliza no processo de classificação heurísticas de disparidades claro-escuro entre regiões da face e SVM;
- A biblioteca FDLib (Face Detection Library), desenvolvida por Kienzle et al. (2005), que utiliza ranqueamento e seleção de vetores de suporte em SVM para aumentar a velocidade de processamento, mantendo razoavelmente a precisão dos resultados;
- O detector de faces UniS, desenvolvido na Universidade de Surrey, Reino Unido. Degtyarev e Seredin (2010) não comentam como funciona o detector e a página do projeto não contém detalhes descritivos suficientes;
- O detector de faces presente no SDK FaceOnIt, desenvolvido no Instituto Idiap (SAUQUET; MARCEL; RODRIGUEZ, 2005), o qual se baseia na abordagem

[4]http://opencv.willowgarage.com/wiki/Welcome

de Viola e Jones (2001) e em uma extensão de LBP (Local Binary Patterns);

- Os detectores FaceSDK e VeriLook, produzidos, respectivamente pela Luxand (http://www.luxand.com) e pela Neurotechnology (www.neurotechnology.com).

Para comparar os resultados dos detectores analisados, Degtyarev e Seredin (2010) utilizaram a localização dos olhos das faces detectadas. Nos casos em que o detector retornava apenas as coordenadas e tamanho da face, foi usada uma heurística para calcular a posição dos olhos. O princípio básico da heurística utilizada é o cálculo de duas proporções. A proporção média entre a distância de posições de olhos para o lado superior do retângulo que marca a face e o tamanho do lado da face. A segunda proporção é obtida pela divisão da proporção média da distância entre os olhos pela largura média de cada face.

Todos os algoritmos foram avaliados utilizando as seguintes bases de imagens: Face Place[5], IMM Face Database[6], *Achermann's face collection*[7], BioID, *The Sheffield Face Database*[8], PIE *Database subset* (SIM; BAKER; BSAT, 2003), *Indian Face Database*(JAIN; MUKHERJEE, 2002), *The ORL Database of Faces*[9], e a *Laboratory of Data Analysis Face Database of Tula State University*[10], totalizando 11677 faces. Os parâmetros avaliados foram: taxa de falsa rejeição (FRR - *False Rejection Rate - taxa de negativos classificados incorretamente como positivos*), taxa de falsa aceitação (FAR - *False Acceptance Rate* - taxa de negativos classificados incorretamente como positivos), distância para um algoritmo exemplar (considerado quando FAR e FRR são 0) e parâmetros de velocidade tais como média e mediana do tempo de processamento.

De acordo com os resultados experimentais, o detector que obteve o melhor desempenho e a maior velocidade de processamento foi o Verilook (FRR igual a 0,0523 e FAR igual a 0,0062), o qual processou uma média de 18 imagens por segundo em um processador Core2Duo de 1.66 GHz e 2GB RAM usando o sistema operacional *Windows Vista HP*. O segundo melhor detector foi a implementação da abordagem de Viola e Jones (2001) contida

[5]http://www.face-place.org/

[6]http://www.imm.dtu.dk/ aam/

[7]ftp://ftp.iam.unibe.ch/pub/Images/FaceImages/

[8]http://www.sheffield.ac.uk/eee/research/iel/research/face

[9]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[10]http://lda.tsu.tula.ru/FD/

na biblioteca OpenCV que obteve FRR igual a 0,0628 e FAR igual a 0,0423.

Levando em consideração que foram testadas 48211 imagens de não faces e a melhor taxa de falsa aceitação ficou em torno de 6 para 1000, pode-se afirmar que os detectores de faces existentes ainda não estão aptos a serem usados *de facto* em situações desafiadoras, e.g., no processamento em tempo real de imagens de fluxo de pessoas, em saguões de aeroportos, uma vez que utilizando o método de varredura de janelas, uma imagem de resolução 320×240 pixels geraria mais de 100 mil janelas para serem processadas e, assim, a taxa de falsos positivos seria bastante alta, necessitando constantemente intervenção humana para eliminar falsos positivos.

Ao contrário dos autores anteriores, Zhang e Zhang (2010) não realizaram uma análise teórica do estado da arte em detecção de faces. Os autores afirmam que a abordagem de Viola e Jones (2001) é um marco na história da detecção de faces e que todos os bons detectores de faces que foram propostos subsequentemente contém alguma variação do método que utiliza Adaboost, cascata de classificadores e imagem integral. Eles utilizam essa afirmação para justificar o fato de que sua revisão da literatura trataria apenas de trabalhos que são variações ou extensões da abordagem proposta por Viola e Jones (2001).

O artigo contém três seções principais, cada uma tratando de trabalhos nos quais foram propostas novas abordagens para a extração de características, a aprendizagem por *boosting* e a utilização de outros classificadores diferentes daqueles que usam *boosting*. Dentre as variações de características tipo Haar citadas no trabalho de Zhang e Zhang (2010), estão: características tipo *Haar* com inclinação de 45 graus (LIENHART; KURANOV; PISAREVSKY, 2002), filtros diagonais (JONES; VIOLA, 2003), criação de histogramas de características tipo *Haar* (WU et al., 2004), *Multi-Block LBP* (ZHANG et al., 2007) e características *Locally Assembled Binary* inspiradas em LBP (YAN et al., 2008).

Dentre as variações de Adaboost citadas por Zhang e Zhang (2010), estão o *Gentle Boost*, o *Real Boost*, e o *FloatBoost*. O principal problema no treinamento de cascatas de classificadores tem sido a seleção de características. Quando características tipo Haar são utilizadas, existem centenas de milhares de possibilidades de características. Várias abordagens foram propostas para reduzir o tempo de treinamento nesses casos, dentre os mencionados por Zhang e Zhang (2010), destacam-se: esquema de busca discreta por descida

de encosta (MCCANE; NOVINS, 2003) e seleção aleatória de subconjuntos de características (BRUBAKER et al., 2005).

Um problema associado ao treinamento de classificadores utilizando Adaboost é o aumento da complexidade do treinamento e a possível impossibilidade de encontrar classificadores fracos que sejam capazes, quando combinados, de atingir as taxas desejadas. Para tratar esse problema e melhorar as taxas de acerto, Xiaohua, Lam e Jiliu (XIAOHUA et al., 2009) propuseram uma abordagem que utiliza informações contextuais da imagem da face para determinar como as características serão extraídas.

Na verdade, a informação contextual utilizada é obtida das regiões externas da face: contornos, linhas do cabelo, linhas do queixo, etc. Essa abordagem está de acordo com o que foi sugerido por Sinha et al. (SINHA et al., 2006), os quais afirmam, todavia, que as informações da região externa da face são uteis para o reconhecimento da face e não mencionam sua utilidade em tarefas de detecção de faces.

Além da utilização de informações da região externa da face, Xiaohua et al. (2009) propuseram também um meio de extrair características de Gabor simplificadas que pode ser realizado por meio de representações integrais quando os ângulos da função de Gabor são $0°$, $45°, 90°$ e$135°$. Os autores afirmam que o método Adaboost sofre de um problema conhecido como não monotonicidade, ou seja, a adição de novas características (classificadores fracos) pode levar a melhorias no desempenho atual, mas a uma queda no desempenho geral. Uma solução para esse problema já foi proposta por Li e Zhang (2004) a partir da utilização de uma busca flutuante, ao invés de uma busca sequencial.

Para tratar o problema da não monotonicidade do método AdaBoost, Xiaohua et al. (2009) propõem o aumento gradual da região que contorna a face para que sejam extraídas características também dessas regiões quando o desempenho dos classificadores que utilizam apenas características da região interna não for adequado (estiver abaixo de um limiar pré-determinado). Foram realizados experimentos de teste do detector treinado com a abordagem proposta utilizando as bases FERET (PHILIPS; MOON, 2000), BioID (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001) e CMU-MIT (ROWLEY; BALUJA; KANADE, 1997). Os resultados obtidos utilizando a base CMU-MIT são comparados com os resultados obtidos por outros classificadores,

dentre eles o de Viola e Jones (VIOLA; JONES, 2001) e o de Rowley, Baluja e Kanade (ROWLEY; BALUJA; KANADE, 1998a). Para quantidades de falsos positivos acima de 40, o método proposto obteve melhores taxas de detecção que todos os outros detectores comparados.

Recentemente, um protocolo para a avaliação de detectores de faces foi proposto por meio da criação e disponibilização da base FDDB (*Face Detection Data Set and Benchmark*). Essa base é formada por um conjunto de imagens contendo faces, bem como rótulos relativos às localizações das faces e suas poses, organizadas por Jain e Learned-Miller (2010). Essas imagens foram coletadas da base *Faces in the Wild data set* (BERG et al., 2004). A FDDB contém anotações para 5171 faces em um conjunto de 2845 imagens. Esse conjunto de dados foi usado em uma competição de detectores de faces realizada no *Workshop on Face Detection 2010* do ECCV (*European Conference on Computer Vision*) e nenhum dos detectores testados obteve taxas de acerto acima de 70% para uma quantidade de falsos positivos fixada em 2000 janelas classificadas incorretamente. Esses resultados são importantes porque indicam que os melhores detectores de faces quando testados em imagens que possuem maior variabilidade que as imagens da base CMU-MIT alcançam resultados, ainda, insatisfatórios.

Para treinar um classificador que será usado para detectar faces, de modo de modo a torná-lo obter taxas de falsos positivos inferiores a 1×10^{-6}, é necessário que sejam utilizadas grandes quantidades de amostras de não-faces. No caso dos treinamentos que empregam abordagens de *bootstrap*, como o método de Viola e Jones (2001), podem ser necessárias milhões de amostras. Para tratar essa quantidade de imagens é necessária a utilização de processamento de alto desempenho durante o treinamento. Na Seção 3.4, discutem-se algumas abordagens de paralelização do método AdaBoost.

3.4 A Utilização do Processamento Paralelo no Treinamento de Classificadores

Conforme foi exposto na Seção 3.3, a abordagem que utiliza Adaboost e representação integral de imagens (e suas variações) é a mais popular e que tem obtido os melhores resultados.

Porém, há um problema prático envolvido em sua utilização que raramente é mencionado nos artigos: a necessidade de utilização de milhões de amostras de não face durante o treinamento. Há uma resposta comum para esse problema que afirma que se o processo de treinamento é longo mas o detector gerado é rápido, então o problema não é muito sério.

Contudo, se os treinamentos estão sendo realizados com a intenção de explorar idéias e novas abordagens o tempo de espera por resultados é um fator decisivo. Como será exposto no Capítulo 6, o treinamento de uma cascata de classificadores utilizando Adaboost e representação Integral pode durar mais de duas semanas. Esse tempo de espera pode ser inaceitável se existirem prazos a serem cumpridos.

Os principais fatores que tornam o treinamento de cascatas de classificadores com Adaboost demorado são: a necessidade de ler milhões de imagens do disco rígido no treinamento de cada nível da cascata (dependendo do modo como o algoritmo é implementado) e a busca pela melhor característica que será utilizada para gerar o classificador fraco. É importante ressaltar que nenhum dos artigos mencionados nesta revisão bibliográfica descreve em detalhes o que foi feito para tratar esses problemas.

Jones e Viola (2003) afirmam que utilizaram mais 100 milhões de amostras de imagens de não faces e 8356 imagens de faces para o treinamento do detector frontal. Se considerarmos uma taxa de falsos positivos de 50% para um treinamento, 100 milhões de imagens de não faces disponíveis e 9000 imagens de não faces usadas no conjunto de treinamento a cada nível da cascata, todas as imagens de não faces deveriam ser lidas a partir do treinamento do nível 14 da cascata, pois $\frac{1}{2^{14}} = 6{,}104 \times 10^{-5}$ e $\frac{9000}{(100 \times 10^6)} = 9 \times 10^{-5}$. Se a leitura de uma imagem de não face for realizada em $1ms$ (tempo compatível com computadores pessoais atuais), a leitura de todas as imagens de não face levaria mais de 27 horas.

Então, para treinar uma cascata de 20 estágios, seriam necessários cerca de 7 dias de processamento apenas para os últimos 6 níveis. Isso, sem considerar o tempo para treinamento dos classificadores fracos. Como foi necessário treinar 35 níveis para gerar a cascata apresentada por Jones e Viola (JONES; VIOLA, 2003) e os autores não fazem nenhum comentário sobre a utilização de processamento paralelo, estima-se que o treinamento da cascata de classificadores para faces frontais levou cerca de 3 semanas para ser finalizado. No artigo publicado posteriormente pelos mesmos autores (VIOLA; JONES, 2004), há um parágrafo

de quatro linhas que comenta a utilização de paralelismo reduzindo o tempo de treinamento de uma cascata de 38 estágios de semanas para um dia. Além disso, não há nenhum detalhe da abordagem utilizada para paralelização.

Consideremos outro caso, o treinamento da abordagem proposta por Chang Huang et al. (HUANG et al., 2007) utilizou 30000 imagens de faces frontais, 25000 imagens de faces de meio perfil e 20000 imagens de perfil. Nesse caso, a quantidade de imagens de não faces necessária é muito maior e portanto o tempo necessário para treinamento seria mais de um mês (fazendo uma estimativa análoga a anterior) se não fosse executado em paralelo. Como alguns dos classificadores podem ser treinados independentemente nessa abordagem, os autores afirmam que o tempo total de treinamento foi de cerca de duas semanas utilizando 3 computadores. Porém, os autores não mencionam a utilização de nenhuma técnica ou software específico para paralelização como *threads* [11], MPI (Message Passing Interface)[12] ou TBB (Intel Threading Build Blocks)[13].

Com a popularização dos processadores de múltiplos núcleos, tem surgido uma tendência em direção à implementação de sistemas paralelos híbridos. Esses sistemas utilizam paralelismo de memória compartilhada entre os núcleos do mesmo processador e troca de mensagens entre os diversos processadores de um *cluster*. Assim pode-se afirmar que existem dois modelos básicos de implementação híbrida (memória compartilhada e não compartilhada) de paralelismo: modelo de apenas-mestre (*master-only*) e modelo de sobreposição (*overlap*) (RABENSEIFNER; HAGER; JOST, 2009).

No modelo apenas-mestre, os núcleos de um mesmo processador se comunicam utilizando OpenMP e os processadores de computadores diferentes se comunicam utilizando MPI. Em um modelo híbrido de sobreposição, mais de um núcleo do processador realiza troca de mensagens com núcleos de outro processador. Rabenseifner, Hager e Jost (2009) apresentam as principais características do processamento paralelo híbrido e indicam que uma tendência é a utilização de sistemas híbridos que utilizam MPI e OpenMP. Outra possibilidade que, em geral, não é comentada pelos autores é a utilização de bibliotecas como a *Intel Threading Building Blocks* que facilita a programação de paralelismo de memória

[11] http://tldp.org/FAQ/Threads-FAQ/index.html

[12] http://www.mcs.anl.gov/research/projects/mpi/

[13] http://threadingbuildingblocks.org/

compartilhada.

As seguintes estratégias devem ser implementadas para obter melhor escalabilidade, segundo Rabenseifner, Hager e Jost (2009):

- Reduzir a sobrecarga de sincronização;
- Reduzir o desequilíbrio de carga;
- Reduzir a sobrecarga computacional;
- Reduzir o consumo de memória;
- Minimizar a sobrecarga de comunicação do MPI.

Rabenseifner, Hager e Jost (2009) afirmam também que o principal problema na obtenção de bom desempenho em arquiteturas híbridas é a inexistência de um modelo de programação adequado para o hardware hierárquico (*cluster* contendo processadores de múltiplos núcleos).

O algoritmo Adaboost é um bom caso de uso para ser utilizado em estudos de paralelização híbrida, apresentando desafios tanto para abordagens de memória compartilhada quanto para abordagens de passagem de mensagens. Por exemplo, a implementação disponível na biblioteca OpenCV utiliza uma única matriz para armazenar todas as características que serão utilizadas em cada etapa de treinamento, o que consiste em um desafio para as abordagens de memória compartilhada. Por outro lado, essa mesma implementação necessita que um conjunto de imagens seja avaliado pelos classificadores intermediários centenas de vezes, o que implica em desafios para abordagens de passagens de mensagens.

Zeng, Tang e Liu (2011) propuseram uma abordagem para paralelizar o algoritmo Adaboost utilizando MPI, OpenMP e TSM (*Transactional Memory*) e adotando o modelo apenas-mestre. Nessa abordagem, foram paralelizados as seguintes etapas do Adaboost:

- A avaliação das características para cada uma das amostras de imagens;
- O método de ordenação *Quick Sort* para a ordenação das características por pureza[14];

[14] A pureza é um critério de classificação usado no treinamento de árvores de decisão.

- A determinação do limiar de classificação;
- A seleção da melhor característica.

Foi utilizado como problema de aprendizagem a geração de uma cascata para detecção de faces frontais. Não ficou claro o motivo pelo qual os autores decidiram utilizar quantidades de imagens que não estão de acordo com o que é descrito na literatura especializada em detecção de faces. O treinamento foi realizado utilizando 64328 imagens de faces e 43712 imagens de não faces. É comum a utilização de uma quantidade muito maior de não faces, para que à medida que a aprendizagem for evoluindo para níveis mais extremos da cascata, novas imagens de não faces estejam disponíveis e sejam utilizadas.

Nessa configuração, provavelmente todas as imagens de não faces já teriam sido usadas pelo algoritmo, a partir do segundo nível. Os resultados e os comentários feitos pelos autores explicitam o fato de que utilizar uma abordagem híbrida apenas-mestre obtém maior velocidade de processamento devido à diminuição da comunicação entre processos. Porém, os autores utilizaram quantidades de imagens nos conjuntos de treinamento que sobrecarregam exatamente a etapa de comunicação entre processos. Se uma configuração mais realista tivesse sido utilizada nos experimentos apresentados, o gargalo, provavelmente não seria a comunicação entre processos.

Nesse contexto, uma configuração realista seria a disponibilidade de milhões de imagens de não-face e a utilização de menores quantidades de imagens de faces e não faces nos conjuntos de treinamento, por exemplo 10000 faces e 20000 não faces. Outro fator importante que deve ser ressaltado é que a utilização de TSM não impactou o tempo de processamento muito mais do que a utilização de OpenMP, o principal ganho para o TSM está relacionado à diminuição da complexidade de programação.

Outra possibilidade para paralelizar um algoritmo de *boosting* seria por meio da realização de treinamentos utilizando conjuntos disjuntos de imagens de acordo com a abordagem proposta por Lazarevic e Obradovic (2002). Nesse artigo, os autores descrevem a paralelização do método Adaboost.M2 (FREUND; SCHAPIRE, 1996) e utilizam como classificador as redes neurais artificiais *feedforward* de duas camadas. Três possibilidades de *boosting* são apresentadas: aprendizagem paralela, aprendizagem distribuída em bases homogêneas e

aprendizagem distribuída em bases heterogêneas. Em todas as possibilidades são treinados classificadores fracos em cada processador utilizando parte dos dados disponíveis globalmente.

Uma das observações realizadas em relação ao método de aprendizagem paralelo é que para conjuntos grandes a abordagem paralela obteve melhor precisão nos resultados utilizando uma menor quantidade de ciclos de aprendizagem do que a abordagem não paralelizada. Como o tempo necessário para treinar redes neurais está associado à quantidade de amostras usadas para treinamento, as abordagens distribuídas realizaram o treinamento muito mais rápido do que a abordagem não distribuída.

Como a principal intenção de Lazarevic e Obradovic (2002) foi mostrar a possibilidade de realizar treinamentos com Adaboost em paralelo ou de modo distribuído, os autores não realizaram medições dos tempos de processamento para os classificadores treinados. Outro fator que deve ser levado em consideração para essa abordagem é que o classificador gerado não será igual a um classificador gerado por uma implementação não paralela. Uma observação histórica importante é que o ano da publicação desse artigo é muito próximo do ano de publicação do artigo de Viola e Jones (2001). Isso põe em evidência a complexidade de paralelização do método de Viola e Jones (2001), pois mesmo após uma década de surgimento ainda não existe (dentre as publicações avaliadas) um método que o tenha paralelizado adequadamente.

Merler, Caprile e Furlanello (2007) propuseram uma abordagem para a paralelização da versão do método Adaboost que não utiliza *bootstrapping*, mas afirmam que sua abordagem pode ser estendida para o Adaboost com *bootstrapping*. O enfoque dessa abordagem, chamada de *P-Adaboost*, está voltado para a dinâmica de atualização dos pesos do método Adaboost. A cada passo de treinamento, o método Adaboost realiza uma atualização dos pesos que depende dos resultados do passo anterior, isso caracteriza essa etapa como sequencial. Para paralelizar essa etapa, os autores propuseram a divisão do treinamento em duas fases. Na primeira fase, o algoritmo é executado sequencialmente até um determinado número de etapas de treinamento. Na segunda fase, é realizada uma estimativa da distribuição de frequência assintótica dos pesos permitindo a independência dos passos posteriores em relação aos anteriores.

Assim, é possível treinar instâncias do modelo em paralelo e agregá-las do modo tradicional no fim do processo. Três experimentos, com quantidades crescentes de dados, foram realizados para avaliar a velocidade e a precisão da abordagem proposta. O terceiro experimento, que possuía a maior quantidade de dados, contava com 285409 amostras com 74 atributos numéricos cada amostra. Foi demonstrado, a partir dos experimentos, que os modelos *P-Adaboost* convergem de modo semelhante aos modelos do Adaboost quando a quantidade de ciclos do estágio sequencial inicial aumenta.

A análise experimental realizada por Merler, Caprile e Furlanello (2007) para o método *P-Adaboost* não leva em consideração o tempo de acesso aos discos, um fator que também não é mencionado por nenhum dos outros artigos avaliados nessa revisão bibliográfica. Como a quantidade de dados é pequena, em relação à quantidade necessária para treinar uma cascata para a detecção de faces, os tempos de processamento são dominados pela fase de aprendizagem do método. Porém, conforme será apresentado no Capítulo 6, dependendo da quantidade de computadores e de imagens utilizados, o processo que consome a maior quantidade de tempo pode ser o carregamento das imagens e a avaliação delas pelos classificadores fracos a cada etapa de treinamento.

Há também algumas abordagens que propõem métodos de paralelizar o algoritmo Adaboost de modo distribuído, com uma quantidade reduzida de trocas de mensagens. Por exemplo, Huang e Shi (2010) propuseram uma abordagem na qual se distribui o processamento relacionado à busca pela melhor característica para cada classificador fraco. Cada máquina componente do sistema distribuído retorna a melhor característica para o nó raiz e a melhor dentre elas é utilizada para o treinamento.

Os autores realizaram experimentos para medir a melhoria na velocidade de processamento. Porém, as quantidades de imagens utilizadas são muito pequenas para um problema de treinamento para detecção de faces. Foram usadas 5646 imagens de faces e 13030 imagens de não faces, enquanto Jones e Viola (2003), sete anos antes, utilizaram mais de cem milhões de imagens de não faces. Além disso, não são mencionadas as taxas obtidas pelo classificador, isso serviria para demonstrar a corretude dos resultados obtidos pelo método distribuído.

Embora a biblioteca MPI esteja consolidada como uma das melhores opções para imple-

mentar paralelismo, outras ferramentas têm sido desenvolvidas. Galtier, Pietquin e Vialle (2007) propuseram uma abordagem de paralelização que utiliza *JavaSpace* para paralelizar o algoritmo AdaBoost usando memória compartilhada. Os próprios autores comentam que a utilização de MPI e C/C++ provavelmente levaria a maior velocidade de processamento, mas o seu objetivo principal seria reduzir a complexidade de programação para facilitar a utilização da abordagem por pessoas que não possuam familiaridade com processamento paralelo.

A paralelização é realizada com abordagem mestre-trabalhadores (*master-workers*). A cada treinamento, cada trabalhador treina seus classificadores fracos com todo o conjunto de treinamento. O mestre obtém os melhores classificadores dos trabalhadores e utiliza aquele que possui menor erro geral. Assim, está claro que essa paralelização é focada na divisão do conjunto total de características entre os trabalhadores. Porém, há um detalhe que passa despercebido: todos os trabalhadores deverão ler todo o conjunto de não faces. Conforme será, descrito em detalhes no Capítulo 6, a etapa de leitura das imagens pode ser mais demorada do que a etapa de treinamento e os autores não mencionam o impacto disso em sua abordagem.

Apesar de usarem como exemplo de problema um treinamento para detecção de faces, Galtier, Pietquin e Vialle (2007) afirmam que não tem a intenção de aplicar essa abordagem em bases de dados imensas. De certo modo, isso é uma contradição pois o treinamento de classificadores com AdaBoost para detecção de faces exige que sejam utilizadas milhões de amostras de não face, é o que os artigos relacionados descrevem e este trabalho de doutorado confirmou.

3.5 Discussão das Abordagens

Com o intuito de obter uma visão geral em relação aos artigos descritos nas seções anteriores, esses são sintetizados nas Tabelas 3.2, 3.3 e 3.4. Da análise dos artigos discutidos, observa-se que os melhores resultados para detecção de faces foram obtidos pelos métodos que utilizaram algum modo de combinação de classificadores (ZHANG; ZHANG, 2010). Dentre eles, os mais representativos utilizaram alguma variação de AdaBoost para realizar

tanto a extração de características quanto a combinação de classificadores que obtivesse os melhores resultados de classificação utilizando as características extraídas, como exemplo pode ser citado o trabalho de Huang et al. (2007). Nas Tabelas 3.2 e 3.3, há nove colunas, as quais contêm:

- Referência: indicação do artigo que foi discutido nesta revisão e do qual foram extraídos os dados da linha correspondente na tabela;
- Característica: indicação se houve extração de características e, em caso positivo, quais características foram extraídas;
- Classificador: os nomes, ou abreviações, dos classificadores usados. Caso haja abreviação, seu significado pode ser obtido na lista de abreviaturas no início da tese;
- Iluminação: caso o artigo trate explicitamente de problemas de iluminação nas faces, essa coluna conterá a palavra "sim", caso contrário conterá a palavra "não";
- Oclusão: indicação se a abordagem trata ou não oclusões;
- Rotação/Pose: indicação se a abordagem é ou não invariante à rotação ou pose;
- Base de imagens: apresentação dos nomes das bases de imagens usadas para testar os detectores;
- Resolução das Imagens/Quantidade de Imagens: apresentação da largura e da altura, em pixels, das imagens usadas para teste e a quantidade de imagens usadas;
- Desempenho: apresentação das taxas de erro e de acerto de detecção nas bases mencionadas em coluna anterior. Alguns trabalhos apresentam taxas de verdadeiros positivos (VP) e falsos positivos (FP), outros apresentam taxa de erro igual (EER - Equal Error Rate) ou curvas ROC. Enfim, não há consenso em relação a melhor forma de apresentar os resultados.

Em relação à coluna "Desempenho", há uma tendência em se comparar detectores de faces usando curvas ROC. Um exemplo dessa tendência, é o protocolo de avaliação de detectores FDDB (JAIN; LEARNED-MILLER, 2010). Esse protocolo fornece, além das imagens,

todo o software necessário para gerar as curvas ROC para avaliar um detector de faces. Além disso, há uma página na web [15] que fornece os dados das curvas ROC de vários detectores de faces avaliados com a FDDB.

Um fator crítico tanto para os métodos que utilizam SVM quanto para os que utilizam Adaboost é o grande tempo necessário para treinamento ou geração de modelos. Para os detectores que utilizam Adaboost, esse fator é menos problemático visto que ele ocorre apenas na etapa de treinamento ou busca por melhor combinação. No entanto, os métodos que utilizam SVM podem levar mais tempo para classificação se a quantidade de vetores de suporte utilizada pelo modelo gerado for alta. Apesar de alguns autores já terem apresentado algumas soluções para esse problema relacionado à grande quantidade de vetores de suporte das SVM (DONG; KRZYZAK; SUEN, 2002; DONG; KRZYZAK; SUEN, 2005), ainda não existe uma solução que trate esse problema de modo eficiente. Este trabalho de doutorado tinha como objetivo inicial propor uma abordagem de detecção de faces utilizando SVM como classificador. Porém, o uso de SVM foi desconsiderado devido à baixa velocidade de processamento dos protótipos de detectores gerados durante a pesquisa.

Embora existam inúmeros métodos diferentes para a detecção de faces, ainda há uma lacuna no que diz respeito à comparação entre eles. Nenhum dos trabalhos analisados neste capítulo apresenta um meio convincente e homogêneo de comparação com outros métodos. Não há uma base de imagens que seja extensivamente utilizada por todos ou pela maioria dos métodos. O que se constata é uma tendência a executar o detector proposto sobre uma base que foi utilizada por outro método e comparar os resultados de verdadeiros positivos e falsos positivos. Uma base de imagens que tem sido bastante utilizada para esse tipo de comparação é a CMU-MIT, que foi utilizada para testes do detector proposto por Rowley, Baluja e Kanade (1998a).

Uma solução proposta por Rodriguez (2006) é realizar os testes do detector com orientação a objetivos. Logo, se um detector for utilizado por um reconhecedor de faces, esse detector deverá ser avaliado com relação à quantidade de faces corretamente reconhecidas que foram extraídas pelo detector. Uma base de imagens em conjunto com um protocolo para a avaliação de detectores foram propostos por Jain e Learned-Miller (2010), a FDDB.

[15] http://vis-www.cs.umass.edu/fddb/results.html

A referida base será usada em alguns dos experimentos apresentados no Capítulo 5.

O detector proposto por Xiaohua et al. (2009) utiliza uma combinação de classificadores treinados com características diferentes (Gabor e Haar), tendo obtido resultados bastante elevados a partir da base CMU-MIT, o que constitui em comprovação do que foi apresentado por Kittler et al. (1998) sobre a utilização de características diferentes para obter taxas de acerto elevadas.

Observando as Tabelas 3.2 e 3.3, pode-se chegar à conclusão de que os classificadores mais usados são redes neurais, SVM e classificadores simples (*stump classifiers*) combinados por alguma variação de *boosting* (*AdaBoost*, *FloatBoost*, *Gentle Boost*, *Real AdaBoost*, *Vector Boosting*, etc). As características mais usadas, provavelmente devido ao uso de *boosting*, foram as características do tipo Haar. Além disso, como o detector proposto por Viola e Jones (2001) foi o primeiro a obter resultados com taxas de verdadeiros positivos acima de 90% e muito mais rapidamente que abordagens anteriores, como a de Rowley, Baluja e Kanade (1998a) e de Schneiderman e Kanade (2000b), todos esses detectores usaram a base CMU-MIT, a qual se tornou a mais popular para a avaliação de detectores de faces.

Na Tabela 3.4, é apresentado um resumo da discussão sobre abordagens que propõem a paralelização de métodos de treinamentos de classificadores que usam *boosting* ou paralelizam partes do processo de combinação de classificadores. Duas medidas muito usadas para avaliação de paralelização são o *speedup* e a escalabilidade. Porém, apenas um dos artigos menciona avaliação desses dois itens. Portanto, o motivo pelo qual não foram inseridas colunas na Tabela 3.4 informando os valores obtidos para *speedup* e escalabilidade é a ausência de tais medições na maioria dos artigos.

O artigo de Rabenseifner, Hager e Jost (2009) não apresenta medidas de *speedup* e escalabilidade por que ele trata de uma discussão sobre os modos de obter paralelismo híbrido e apresenta avaliações em termos de *Teraflops* por segundo. Embora Merler, Caprile e Furlanello (2007) também apresentem uma abordagem para paralelizar o método AdaBoost, não mencionam as medidas de *speedup* e escalabilidade.

O único artigo que avalia a escalabilidade e o *speedup* de sua abordagem é o de Galtier, Pietquin e Vialle (2007). Deve-se ressaltar que as medições foram realizadas apenas para o laço de interação da combinação de classificadores. Há dois gráficos que exibem resultados de avaliação da paralelização, o primeiro apresenta os tempos de execução versus a quantidade de computadores usada, o segundo apresenta as medidas de *speedup* versus a quantidade de computadores usada. Há uma incoerência entre os gráficos: os tempos apresentam um decaimento exponencial, mas os valores de *speedup* aumetam de modo praticamente linear. A variação dos tempos nos testes de escalabilidade ficou em torno de 10 segundos com o aumento das quantidades de imagens e de computadores usados.

Outro artigo, dentre os discutidos, que apresenta uma tabela com os cálculos de *speedup* é o de Huang e Shi (2010). Os autores apresentam os resultados de *speedup* para 4 situações resultantes da combinação de quantidades de computadores usados (2 ou 4) e quantidade de características usadas (32 ou 64). Para o caso em que foram usados 4 computadores e 64 características, o *speedup* foi de 2,66. Os autores não realizaram testes para avaliar a escalabilidade da abordagem proposta.

É no artigo de Zeng, Tang e Liu (2011) que realmente se propõe uma forma de paralelização que se assemelha àquela proposta nesta Tese. Apesar de Zeng, Tang e Liu (2011) utilizarem o termo *speedup* em seu texto, o *speedup* não é calculado formalmente, mas sendo inferido da análise de gráficos contendo os tempos de execução. Além disso, os autores não mencionam a escalabilidade de seu algoritmo. Um problema semelhante ocorre com o artigo de Lazarevic e Obradovic (2002), no qual os autores mencionam o *speedup*, apresentam uma tabela de comparação, mas não o calculam de acordo com a Equação 6.1 apresentada no Capítulo 6.

Tabela 3.1: Rótulos para as referências apresentadas nas Tabelas 3.2, 3.3 e 3.4 que contêm o resumo dos trabalhos analisados neste capítulo.

Referência	**Rótulo**
(JESORSKY; KIRCHBERG; FRISHHOLZ, 2001)	1
(HADID; PIETIKÄINEN; AHONEN, 2004)	2
(TOEWS; ARBEL, 2009)	3
(RAMIREZ; FUENTES, 2005)	4
(ROWLEY; BALUJA; KANADE, 1998b)	5
(VIOLA; JONES, 2004)	6
(LIENHART; KURANOV; PISAREVSKY, 2002)	7
(JONES; VIOLA, 2003)	8
(MASIP; BRESSAN; VITRIÀ, 2005)	9
(LI; ZHANG, 2004)	10
(WU et al., 2004)	11
(HUANG et al., 2005)	12
(HUANG et al., 2007)	13
(MEYNET et al., 2007)	14
(CHEN; HUANG; FU, 2007)	15
(RODRIGUEZ, 2006)	16
(MEYNET; POPOVICI; THIRAN, 2005)	17
(MEYNET et al., 2005)	18
(OSUNA; FREUND; GIROSI, 1997)	19
(XIAOHUA et al., 2009)	20
(ANILA; DEVARAJAN, 2010)	21
(RABENSEIFNER; HAGER; JOST, 2009)	22
(ZENG; TANG; LIU, 2011)	23
(LAZAREVIC; OBRADOVIC, 2002)	24
(MERLER; CAPRILE; FURLANELLO, 2007)	25
(HUANG; SHI, 2010)	26
(GALTIER; PIETQUIN; VIALLE, 2007)	27

Tabela 3.2: Resumo dos trabalhos analisados. Na última coluna da tabela, há resultados de desempenho dos detectores avaliados. Como pode ser visto, não há um padrão seguido pelos diversos artigos. Uns apresentam taxas de verdadeiros positivos (VP) e falsos positivos (FP), outros apresentam taxas de precisão e revocação, etc.

Referência	Característica	Classificador	Iluminação	Oclusão	Pose	Base	Resolução Quantidade	Desempenho
1	Bordas	Hausdorff distance	Não	Não	Não	XM2VTS BioID	360 × 288 384 × 286 1180/1521	VP: 98,4%/FP não comentado VP: 91,8%/FP não comentado
2	LBP	SVM	Sim	Sim	Sim	MIT-CMU	Várias 80	VP: 97,8% 13 falsos positivos
3	SIFT	OCI	Sim	Sim	Sim	MIT-CMU	Várias Não definida	Bayes EER: 0,26 SVM EER: 0,33
4	Histogramas	Naive Bayes SVM Voted Perceptron Indução de regra $C4.5$ Rede Neural	Não	Não	Não	BioID	384 × 288 1521	VP: 93,23 2236 falsos positivos
5	Tons de cinza	Comb. de R. Neurais	Não	Não	Sim	MIT-CMU	Várias 180	VP: 92,4% 67 falsos
6	*Haar-like*	Adaboost	Sim	Sim	Sim	MIT-CMU	Várias 130	VP: 93,9% 167 falsos positivos
7	*Haar-like* melhorado	Gentle Adaboost	Sim	Sim	Sim	MIT-CMU	Várias 130	VP: 82,7% 10 falsos positivos
8	*Haar-like* melhorado	Adaboost	Não	Não	Sim	MIT-CMU	Várias	VP: 89,7/221 falsos positivos

Tabela 3.3: Resumo dos trabalhos analisados. Continuação.

9	*Haar-like*	Adaboost	Sim	Sim	Sim	XM2TVS AR	360 × 288 26000 Não Faces 2000 Faces	99,73% VP: FP: 0,2%
10	*Haar-like*	Floatboost	Sim	Sim	Sim	MIT-CMU Proprietária	Várias 125/Não definida	VP: 90,2 31 falsos positivos
11	*Haar-like*	Real Adaboost	Não	Não	Sim	MIT-CMU	Várias	VP: 94,5% 57 falsos positivos
12	*Haar-like*	Vector Boosting	Não	Não	Sim	MIT-CMU	Várias	VP: 88,0% 48 falsos positivos
13	Granular	Vector Boosting	Não	Não	Sim	MIT-CMU	Várias	VP: 97% 75 falsos positivos
14	*Haar-like* e filtros Gaussianos	Adaboost	Sim	Sim	Sim	Vídeo	320 × 240 1500 quadros	VP: 93% FP não mencionada
15	*Haar-like* e *Gabor-like*	*hybrid-boost* e probabilidades	Sim	Sim	Sim	FERET e *World Wide Web*	87 imagens para teste 1128	VP: 94,7% FP não mencionada
16	LBP	Adaboost	Sim	Sim	Sim	CMU/WWW/ Sussex	Várias/573	VP: 94% 743 falsos positivos
17	PCA	Combinação de SVM	Moderado	Não	Não	Banca/ XM2VTS	20 × 15 7822 faces	VP: 93,6% FP: 1,90%
18	PCA	Validação cruzada com SVM	Sim	Não	Não	BANCA XM2VTS	20 × 15/ 907822	VP: 92,2% FP: 1,8%
19	Tons de cinza	Comb. de SVM	Não	Não	Não	MIT	Várias 468	VP: 97,1% 4 falsos positivos
20	Gabor e Haar	Adaboost	Não	Não	Não	FERET BioID CMU-MIT	360 × 288/1762 384 × 286/1521 Várias	VP: 99,77% 123 falsos alarmes VP: 99,41% 143 falsos alarmes VP: 95% 60 falsos positivos
21	Bordas	Redes Neurais	Não	Não	Não	BioID	384 × 286/1521	VP: 95,33% FP: 4,5%

Tabela 3.4: Resumo dos trabalhos analisados que propõem métodos de paralelização.

Referência	Etapa Paralelizada	Bases de Dados	Quantidades de Dados	Modelo	Biblioteca
22	Discussão genérica sobre paralelização	Não	Não	Híbridos	MPI e OpenMP
23	a avaliação das características o método de ordenação a determinação do limiar a seleção da característica	Proprietária	64328 faces 43712 não faces	Híbrido	MPI, OpenMP, TSM
24	Treinamento completo em subconjuntos dos dados	Sintético e Repositório UCI: Covertype, Pen-Based, Waveform, e LED	mais de 581102	Não mencionado	Não mencionado
25	Atualização dos pesos	Sintéticos Repositório UCI e *Protein Homology data set*	mais de 285409	Não mencionado	Não mencionado
26	Escolha de Características	Proprietária	5646 faces e 13030 não faces	Distribuído	Não mencionado
27	Escolha de Características	Proprietária	1000 a 8000	Distribuído	JavaSpace

3.6 Considerações Finais

Algumas conclusões podem ser formuladas a partir da análise dos artigos apresentados para guiar a criação de um novo método de detecção de faces. Em primeiro lugar, deve-se extrair características das imagens que ajudem o classificador em sua tarefa. A utilização dos valores das intensidades dos pixels por si só não agregam valor a qualidade dos resultados de classificação. Além disso, as características extraídas devem ser as mais invariantes possíveis a mudanças de iluminação, orientação, pose, etc. Um exemplo desse tipo de invariância é a invariância à orientação obtida pela utilização de características LBP. O método LBP permite obter tal tipo de invariância devido ao uso de diferenças centro vizinhanças entre os valores dos tons de cinza dos pixels analisados. Conforme pode ser visto no Apêndice A, a aplicação de LBP para obter um detector de faces invariante à rotação foi investigada nesta pesquisa de doutorado. Porém, a versão de LBP que é invariante à rotação (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002) não possui representatividade suficiente para representar faces, a não ser que as características sejam extraídas localmente. Contudo, a extração local das características impossibilita a obtenção de invariância à rotação.

Um fator que influencia a qualidade dos resultados de detecção é a combinação de classificadores, pois, como foi mostrado por Kittler et al. (1998), quando se utilizam em conjunto classificadores treinados com características diferentes, eles se complementam fortalecendo o resultado final. Esse é o motivo pelo qual o método Adaboost se sobressai sobre os demais. No entanto, em sua forma tradicional (VIOLA; JONES, 2004; LIENHART; KURANOV; PISAREVSKY, 2002) esse método utiliza características muito simples e de pouca representatividade, conforme já foi exposto por Balas e Sinha (2003).

Capítulo 4

Abordagem Proposta

Neste capítulo, a abordagem proposta para a detecção de faces invariante à rotação no plano é apresentada. Além disso, também é apresentada, a abordagem para a paralelização do treinamento de cascatas de classificadores.

4.1 Abordagem Proposta para a Detecção de Faces Invariante à Rotação

O principal objetivo desta tese é apresentar uma abordagem de detecção de faces invariante à rotação e que seja o mais robusta possível a variações de iluminação e oclusão. Assim, surgiu a necessidade de implementar modificações na abordagem inicialmente proposta que é detalhada no Apêndice A. As principais alterações na abordagem estão relacionadas às características e ao classificador utilizados. Foram empregadas características do tipo Haar e uma cascata de classificadores treinados de modo semelhante ao que foi proposto por Viola e Jones (2004). O questionamento decorrente do que foi afirmado anteriormente sobre a abordagem proposta seria: onde está a inovação?

O método *JointBoost* proposto por Torralba, Murphy e Freeman (2004), Torralba, Murphy e Freeman (2006), Torralba, Murphy e Freeman (2007) para compartilhar características entre múltiplas classes de imagens, será agora brevemente explicado, visto

que ele serve de inspiração para a abordagem multipose apresentada posteriormente.

Torralba, Murphy e Freeman (2007) argumentam que é possível demonstrar, tanto subjetiva (por meio de inspeção visual) quanto objetivamente, que algumas características de faces frontais estão presentes em faces em perfil ou que características de faces frontais sem rotação (0 graus) estão presentes em faces frontais com outros ângulos de rotação no plano. Essa observação pode ser estendida para categorias de objetos diversas, como automóveis, casas e animais. A partir desta linha de raciocínio, Torralba, Murphy e Freeman (2007) propõem uma abordagem de *boosting* para problemas multiclasse, o *JointBoost*.

No método *JointBoost*, a cada ciclo de obtenção de classificador fraco, a característica escolhida será aquela que obtiver o menor erro de classificação para a maior quantidade de classes diferentes. Assim, pode-se afirmar que essa característica é compartilhada entre as diferentes classes. Segundo os autores (TORRALBA; MURPHY; FREEMAN, 2007), seus experimentos mostram que classificadores de objetos treinados conjuntamente (*jointly*, ou seja usando *JointBoost*) tendem a selecionar características que generalizam bem para classes diversas. Essas características costumam ser bordas, *blobs*, etc.

As características do tipo Haar, como aquelas usadas por Viola e Jones (2004), também podem ser usadas para generalizar entre múltiplas classes. Porém, tal poder de generalização é restrito, ou seja, não é possível obter invariância à rotação por treinamento para imagens de faces utilizando apenas características do tipo Haar. Nesta tese, a invariância à rotação por treinamento refere-se a um treinamento de cascata de haar, com *AdaBoost*, em que as imagens de faces são rotacionadas no plano.

Alguns experimentos foram realizados para verificar a possibilidade de treinar um classificador de faces invariante à rotação simplesmente variando a rotação das faces de treinamento e não obtiveram êxito. Porém, alguns *insights* muito importantes foram tirados desses experimentos. Primeiro, o classificador treinado com características do tipo *Haar* não será capaz de generalizar faces frontais com qualquer ângulo de rotação no plano, mas até atingir uma certa quantidade de estágios o treinamento converge satisfatoriamente. À medida que mais estágios vão sendo treinados, o problema de classificação se torna mais complexo e as características disponíveis não são capazes de generalizar adequadamente. Outra idéia inspirada por esses experimentos é que classificadores treinados com invariância por treinamento,

mas com um número reduzido de estágios, podem ser combinados para gerar uma árvore de classificadores multipose.

Na Figura 4.1, há uma representação simplificada de uma árvore de classificadores treinados com invariância por treinamento e com quantidades reduzidas de estágios, a qual pode ser usada para detectar faces frontais com qualquer ângulo de rotação no plano. A raiz da árvore é uma cascata de no máximo 5 estágios e classifica faces frontais em qualquer orientação no plano. Como a árvore é binária, as faixas de orientação vão sendo divididas por dois à medida que a classificação se propaga pela árvore. A divisão por dois permite que classificadores mais específicos sejam usados nos níveis mais profundos da árvore. Além disso, como em abordagens anteriores, a quantidade de falsos positivos vai sendo reduzida exponencialmente a cada nível da árvore.

O círculo ao lado da árvore de classificadores, no canto superior esquerdo, ilustra o padrão de orientação de faces usado. Deve-se salientar que o padrão usado difere do círculo trigonométrico, que padroniza o ângulo 0° correspondendo ao ângulo 270° do padrão adotado nesta tese. A diferença entre os padrões é simplesmente um deslocamento de 90° para que o ângulo 0° corresponda à imagem com a cabeça voltada para cima (*upright face*).

Outra característica importante da árvore de classificadores apresentada é que cada folha atua numa faixa de 20°. Ou seja, os classificadores das folhas devem ser treinados com imagens de faces que possuem variação de $\pm 10°$ em relação ao ângulo que rotula a folha. Além disso, folhas vizinhas possuem uma interseção de 10° em suas faixas de cobertura. Assim, o ângulo central de uma folha corresponde ao ângulo de *borda* da folha vizinha.

O arranjo de faixas de ângulos das folhas foi projetado para permitir redundância e reforço de classificação para ângulos problemáticos, poucas amostras de treinamento de faces tenham sido usadas. A redundância também permite que uma janela candidata seja classificada por mais de uma folha. A folha que irá ser usada para classificar tal janela será a que obtiver o mais alto grau de confiança (que pode ser um limiar, ou uma probabilidade).

É evidente que a árvore de classificadores apresentada na Figura 4.1 ainda não é totalmente multipose. Estendendo a idéia da árvore de classificadores para faces frontais, uma árvore semelhante à apresentada na Figura 4.2 pode ser construída. Deve-se treinar a árvore

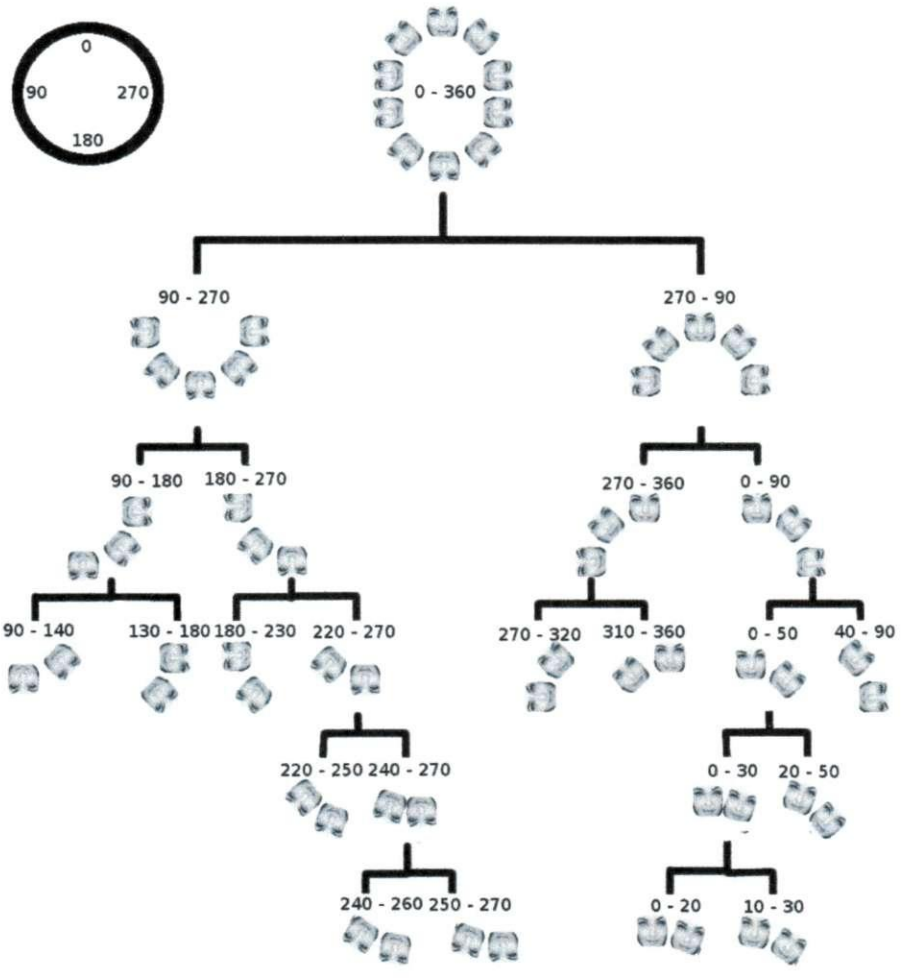


Figura 4.1: Árvore de classificadores para faces frontais com rotação no plano.

para faces frontais utilizando também imagens de faces com pequenas variações de ângulos fora do plano, de forma que a categoria conhecida na literatura especializada como meio perfil seja classificada corretamente.

O objetivo de um classificador que utilize a a abordagem apresentada na Figura 4.1 é apenas detectar faces, pois ele não classificaria adequadamente algumas poses como meio-perfil. Contudo, a extensão dessa abordagem para um classificador capaz de classificar também a pose da face não seria complicado: bastaria adicionar duas novas sub-árvores para as poses meio-perfil-esquerdo e meio-perfil-direito, respectivamente.

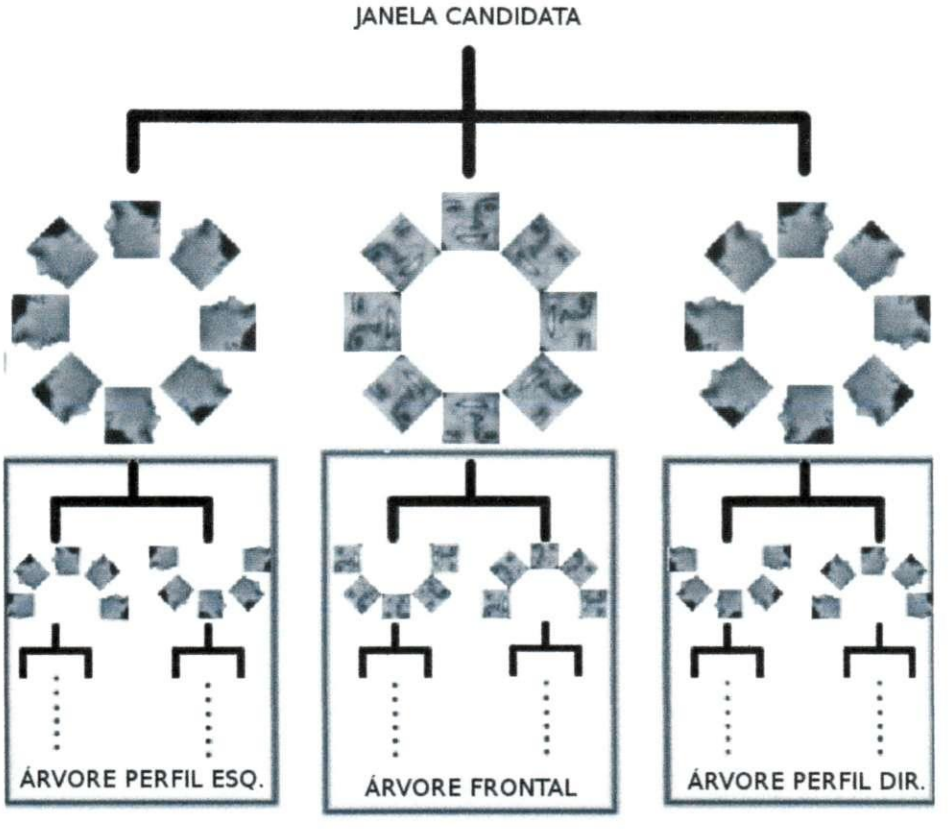


Figura 4.2: Árvore de classificadores multipose.

4.2 Uma Abordagem Híbrida de Paralelização do Método de Viola e Jones de Treinamento de Classificadores para Detecção de Faces

Para entender a abordagem de paralelização proposta, antes é necessário entender a abordagem de Viola e Jones (2001), Viola e Jones (2004). Portanto, esta subseção é dividida em duas partes. A primeira parte explica a abordagem de Viola e Jones (2001), Viola e Jones (2004) e a segunda explica a abordagem de paralelização.

A abordagem de Viola e Jones (2001) é baseada principalmente em classificadores fracos, *boosting* de classificadores e *bootstrapping*. Para criar classificadores fracos, árvores de decisão, possuindo apenas um nó, são treinadas. Como características fracas, Viola e Jones (2001) usaram características do tipo Haar.

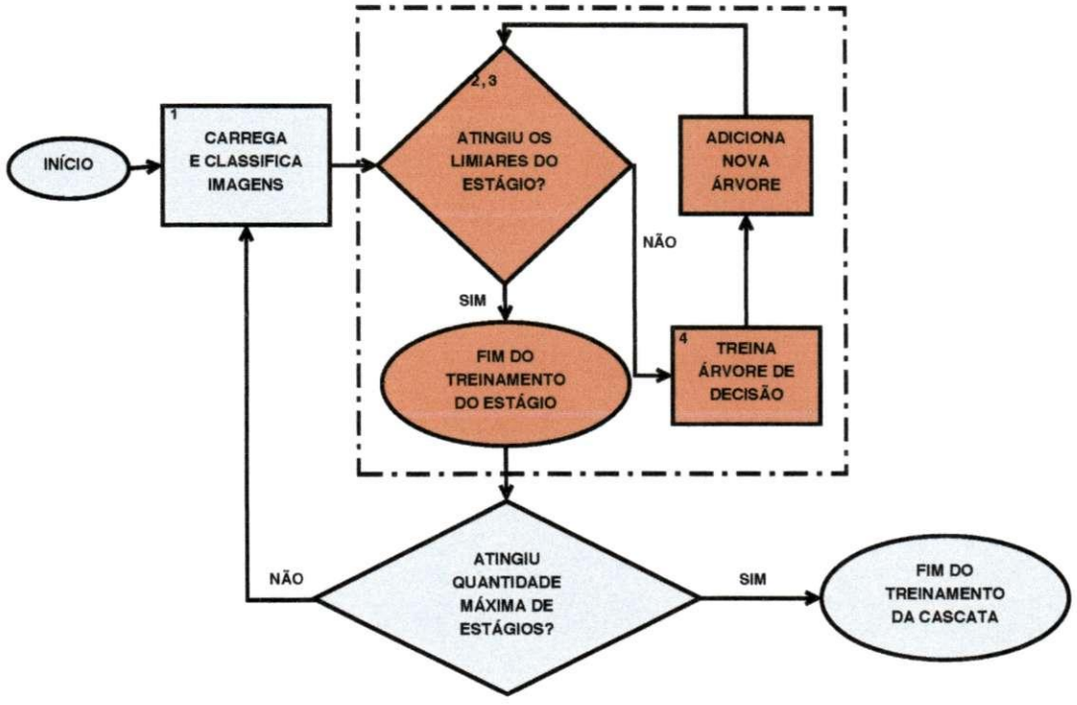


Figura 4.3: Diagrama representando o método de Viola e Jones. As regiões amarelas indicam estágios de treinamento propriamente dito, as regiões cinza indicam processamento geral.

No método de Viola e Jones (2001), as árvores de decisão passam pelo processo de *boosting* Adaptativo (AdaBoost) (FREUND; SCHAPIRE, 1999). Outras variantes de *boosting* podem ser usadas, na abordagem desta tese foi usado o Gentle Boost (FRIEDMAN; HASTIE; TIBSHIRANI, 2000). Outra característica muito importante da abordagem de Viola e Jones (2001) é o método sistemático de *bootstrap*. Um dos primeiros trabalhos a mencionar o uso de *bootstrap* no treinamento de classificadores para detecção de objetos em imagens foi o de Sung e Poggio (1998). O *bootstrap* consiste basicamente no uso de imagens classificadas incorretamente para retreinar o classificador com *padrões difíceis*. A abordagem de Viola e Jones (2001) é representada na Figura 4.3.

O número de cada item na lista a seguir é atribuído aos passos correspondentes na Figura 4.3 que correspondem aos passos principais que são propensos à paralelização:

1. os procedimentos de carregamento e classificação;

2. o cálculo do limiar do estágio;

3. a classificação após a adição de um novo classificador fraco; e

4. o treinamento da árvore de decisão.

Constata-se na Figura 4.3, que há dois passos de classificação. O primeiro passo, correspondendo ao item número 1, usa todos os classificadores de todos os estágios treinados. O segundo passo de classificação, correspondendo ao item número 3, usa somente os classificadores para o estágio que está sendo treinado.

Outra parte do método que pode ser facilmente paralelizada é o cálculo do limiar de cada estágio, marcado com o número 2 na Figura 4.3, o qual é realizado por meio da classificação das imagens de faces. Em seguida, o limiar da árvore de decisão escolhido será aquele que atingir a taxa de verdadeiros positivos mínima determinada no início do treinamento. As imagens de não-faces são classificadas usando esse limiar.

Finalmente, o treinamento da árvore de decisão também pode ser paralelizado. Nesse treinamento, todas as características de todas as imagens devem ser avaliadas. Portanto, uma matriz de dimensões *(número de imagens)* × *(número de características)* deve ser processada. Esse passo foi paralelizado na abordagem proposta nesta tese usando *threads* com TBB (*Intel Threading Build Blocks*) [1]. A seguir, a abordagem de paralelização será explicada em mais detalhes.

À medida que os estágios são treinados, o procedimento de recorte das imagens vai se tornando cada vez mais demorado, devido ao aumento exponencial da quantidade de imagens necessárias para classificação. O procedimento de recorte é paralelizado dividindo-se as imagens disponíveis para recorte igualitariamente entre os computadores. Esse procedimento é ilustrado na Figura 4.4.

Para entender a Figura 4.4, considere-se o caso em que há 5 computadores e 1000 imagens de não faces disponíveis das quais é possível obter 1 milhão de recortes. Os parâmetros de treinamento estabelecem que sejam usados 1000 recortes de imagens de não-faces para o treinamento de cada estágio. Nesse caso, as 1000 imagens disponíveis serão divididas igualmente entre os computadores e cada um deles será responsável pela computação de 200 recortes. Em outras palavras, cada computador deverá carregar e recortar imagens de seu

[1] http://threadingbuildingblocks.org/

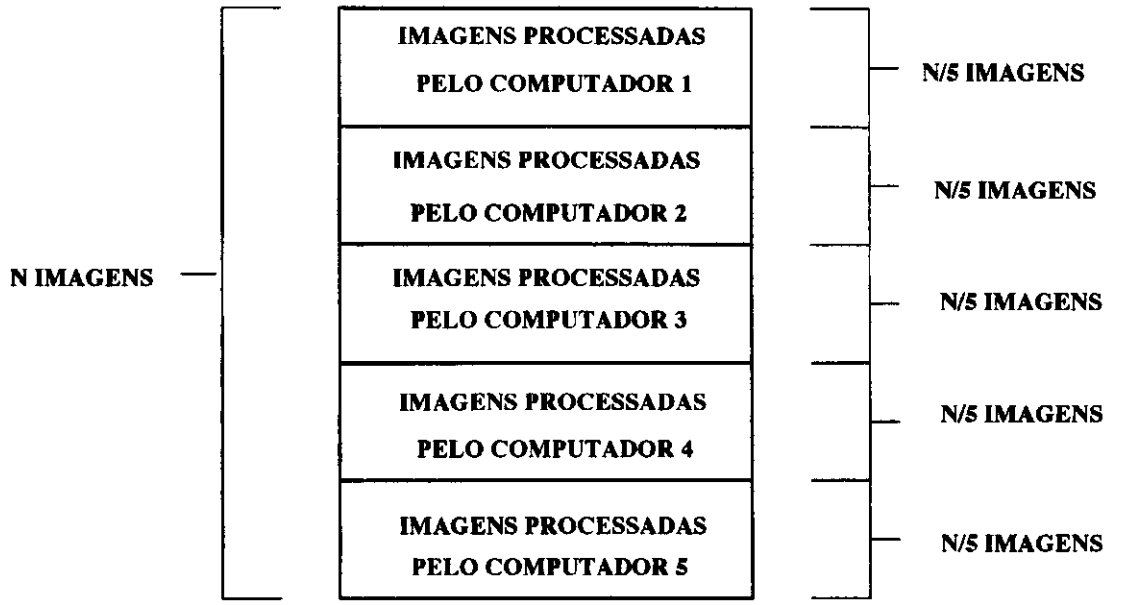


Figura 4.4: Ilustração para a divisão igualitária entre os computadores do conjunto de imagens disponíveis para recorte.

subconjunto, até classificar incorretamente 200 recortes, no primeiro estágio de treinamento.

Após a determinação da característica mais representativa, esta é usada para classificar imagens de faces. As árvores de decisão são classificadores *stump* com apenas um nível. Os limiares de cada árvore são ordenados pela taxa de classificação de modo a determinar o limiar que alcance a menor taxa de falsos positivos. Após a escolha do limiar de classificação de positivos, as amostras negativas são classificadas. Os dois procedimentos de classificação são paralelizados por passagem de mensagens.

As imagens de não faces foram obtidas de dois conjuntos: um contendo pessoas e outro sem pessoas. O conjunto contendo pessoas foi selecionado de imagens da web, de fotos pessoais e fornecidas por amigos e conhecidos do autor desta tese. Todas as faces foram selecionadas manualmente, por meio da utilização de software implementado especificamente para essa atividade, tendo sido suas coordenadas salvas, para uso futuro. Na Figura 4.5, há uma ilustração da interface do software utilizado para seleção manual das regiões de faces.

No processo de recorte, o algoritmo verifica se a região a ser cortada foi marcada previamente como face, senão ela é recortada. O processo de recorte de imagens contendo pessoas é muito importante para adicionar variabilidade e desafios realísticos ao processo de treina-

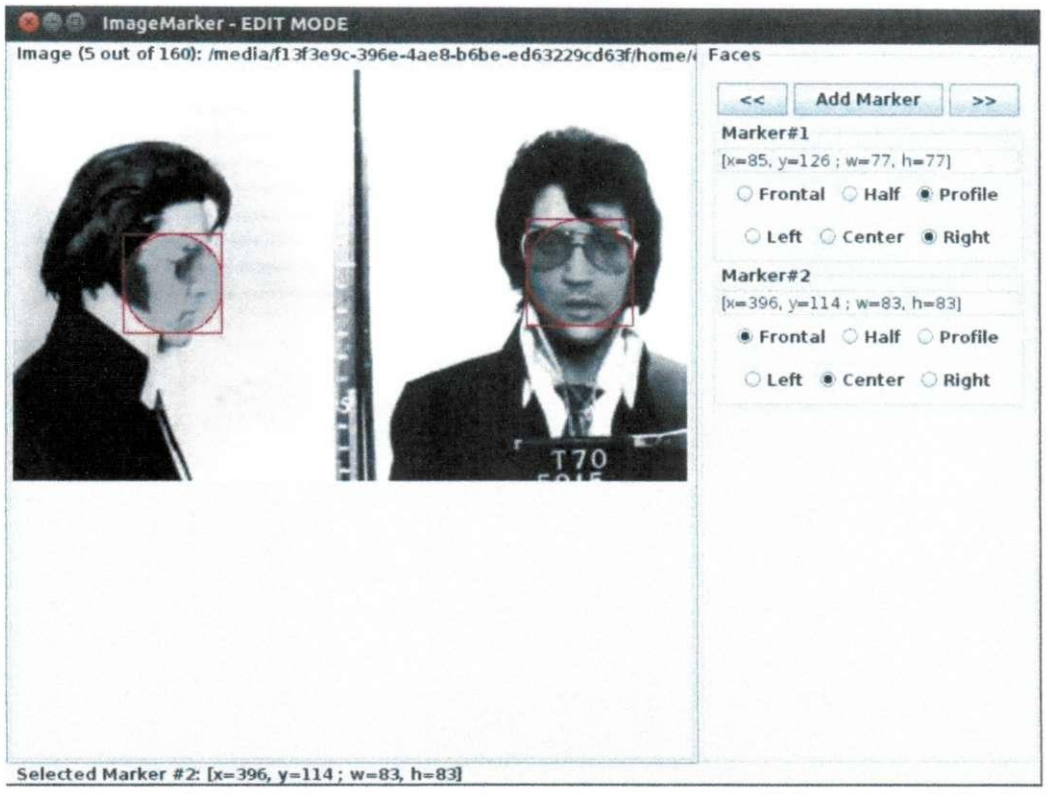


Figura 4.5: Ilustração da tela do programa usado para selecionar imagens de faces.

mento, pois muitas das imagens recortadas nesse caso são regiões contendo cabelo, barba, roupas de diferentes cores e texturas, além de variação de condições de iluminação.

As imagens que não continham pessoas foram obtidas de um conjunto disponível no sítio eletrônico sobre treinamento *AdaBoost* do autor Naotoshi Seo [2]. A referida base contém grande variabilidade de ambientes externos e internos, contendo vegetação, móveis de residências e de escritórios e uma grande variedade de condições de tempo (sol, chuva, neve, etc). Nas Figuras 4.6 e 4.7 algumas amostras de imagens usadas para recorte de faces e de não faces, respectivamente, são apresentadas.

A biblioteca utilizada para adicionar paralelismo ao OpenCV por meio de passagem de mensagens foi a MPICH2 [3]. Segundo a documentação da biblioteca, o MPICH2 é uma implementação do padrão MPI (versão 1 e versão 2) que possui alta portabilidade e alto desempenho e os seus objetivos são:

[2] http://tutorial-haartraining.googlecode.com/svn/trunk/data/negatives/

[3] http://www.mcs.anl.gov/research/projects/mpich2/

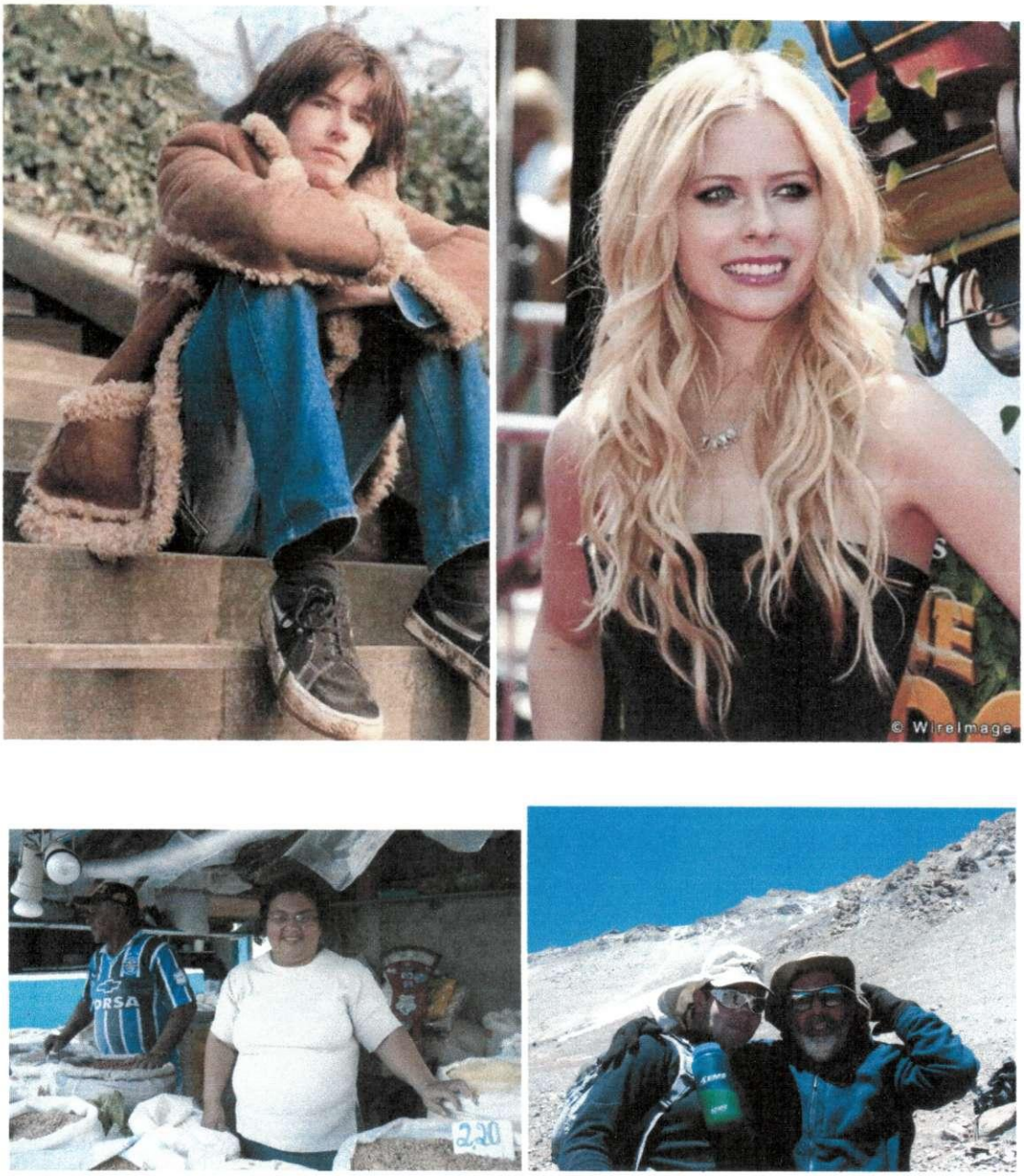


Figura 4.6: Exemplos de imagens contendo pessoas usadas para recortar amostras de faces e de não-faces.

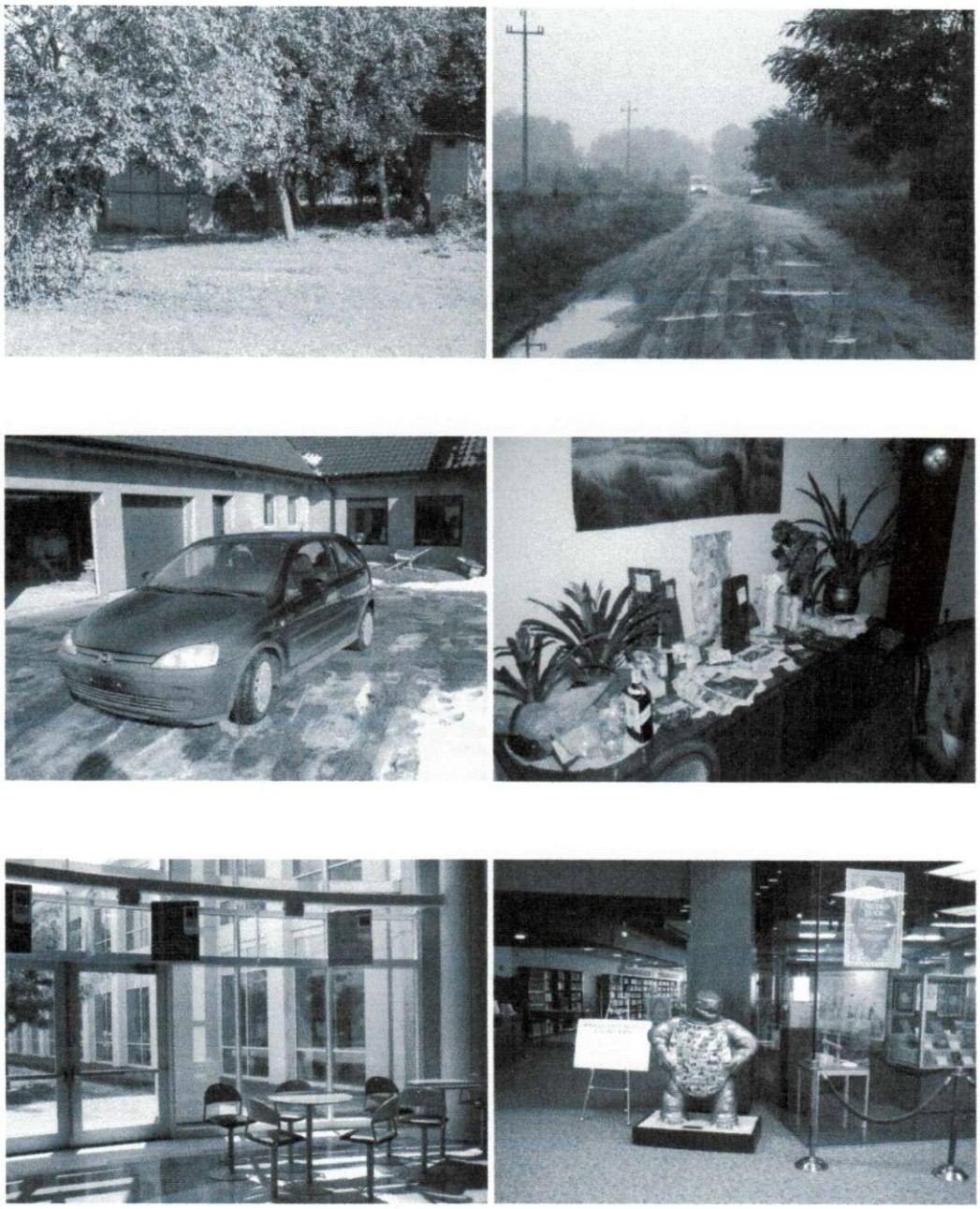


Figura 4.7: Exemplos de imagens sem pessoas utilizadas para gerar recortes de não faces. Obtidas da base fornecida por Naotoshi Seo.

- prover uma implementação do MPI que suporta eficientemente plataformas de comunicação diferentes;
- habilitar pesquisa de alta qualidade em MPI;

A utilização da biblioteca MPICH2 é bastante simples. Após cada computador estar devidamente configurado na rede, deve ser criado um usuário com mesmo nome em todos os computadores que farão parte do *cluster*. Geralmente, as implementações que utilizam MPI trabalham em modo mestre-escravo. Um dos computadores é definido como mestre, é ele que realizará o gerenciamento das tarefas. O gerenciamento de tarefas envolve: dividir adequadamente a carga de processamento para cada computador (quantas imagens cada computador irá processar, por exemplo) e realizar operações de difusão e de redução.

A biblioteca MPICH2 deve ser instalada em cada um dos computadores que farão parte do cluster. A transmissão de informações entre computadores do cluster utiliza ssh e scp. Para que não seja necessário digitar senhas sempre que for necessária alguma comunicação entre os computadores, o serviço de ssh deve ser configurado para não pedir senha. Isso é realizado no Linux utilizando a ferramenta ssh-keygen. O ssh-keygen gera uma chave pública de criptografia RSA a qual é copiada para um diretório *.ssh* na pasta do usuário que criou a chave, localizada no outro computador para o qual deseja-se realizar acesso ssh sem senha. Então, a chave pública do mestre é copiada para todos os escravos e a partir daí o sistema MPI poderá transmitir mensagens entre os computadores de modo seguro e sem a necessidade de senhas.

O MPICH2 fornece um sistema de gerenciamento de processos denominado Hydra, para iniciar tarefas paralelas. O modo mais simples de utilização do Hydra é criar um arquivo de texto contendo os endereços IP de todos os computadores que fazem parte do cluster. Esse arquivo deve estar localizado no mesmo diretório onde o executável paralelo está. Além dos endereços IP dos computadores, o arquivo também pode conter a quantidade de processos paralelos que serão executados naquela máquina. Por exemplo, considere-se um arquivo chamado *hosts.txt* para configurar um cluster de cinco computadores, com conteúdo semelhante ao que é mostrado abaixo:

```
192.168.50.2:1
```

```
192.168.50.3:2
192.168.50.4:2
192.168.50.5:4
192.168.50.6:4
```

Se o primeiro computador listado no arquivo *hosts.txt* contém apenas um núcleo, o segundo e o terceiro computadores possuem dois núcleos cada um e o quarto e quinto computadores possuem quatro núcleos, o modo como o arquivo foi configurado permitirá que cada núcleo de cada computador receba uma parte do processamento. Outro fator importante é que as tarefas são atribuídas a cada computador na ordem em que seus endereços aparecem na lista. Assim, se são requisitados para processamento apenas três computadores, os três primeiros da lista é que receberão tarefas. Para executar um programa que foi implementado usando a MPICH2, basta executar um comando como o apresentado abaixo:

```
mpiexec -f hosts.txt -n 5 seuPrograma argumento1 ... argumentoN
```

Nesta chamada de programa, o termo mpiexec é o programa da biblioteca que gerencia os processos paralelos, o argumento *-f hosts.txt* indica ao gerenciador qual o arquivo que contém os endereços dos computadores do *cluster*, o argumento *-n 5* indica que serão utilizadas cinco linhas de execução paralelas. Os outros argumentos correspondem à chamada normal ao programa paralelo com seus argumentos.

O MPICH2 possui grande versatilidade em relação à quantidade de linhas de execução. Por exemplo, se o usuário possui apenas um computador com quatro núcleos, o *mpiexec* pode ser executado com *-n 4* e o programa será executado em paralelo no computador, sendo distribuído entre os quatros núcleos.

A Intel fornece uma biblioteca de código aberto para implementação de código *multi-thread*, ou seja, paralelismo com memória compartilhada. A TBB [4] (*Threading Building Blocks*) é uma biblioteca de código aberto que permite a implementação de código sem que seja necessário conhecimento profundo do funcionamento das linhas de execução do sistema operacional. Há outras bibliotecas livres para a implementação de código *multi-thread*; por exemplo, a biblioteca *pthread* é a biblioteca POSIX padrão dos sistemas UNIX.

[4] http://threadingbuildingblocks.org/

Amplamente utilizada para implementação de programas em C/C++, porém exige um grau de conhecimento de mais baixo nível, comparada à TBB.

Como a implementação do treinamento de cascata de classificadores *AdaBoost* do OpenCV possui um trecho de código paralelizado com TBB, foi decidido que os trechos do código que necessitassem de paralelismo de memória compartilhada seriam também implementados usando a TBB.

A TBB é muito útil quando se deseja paralelizar *loops*, pois se encarrega de dividir igualitariamente e de acordo com a carga atual as tarefas entre os núcleos do processador. Por exemplo, para paralelizar um *loop for*, define-se uma extensão de trabalho para cada núcleo, chamada de *blocked_range*. O *blocked_range* define o início e o fim da tarefa atribuída a cada núcleo para evitar acesso indevido de memória por processos diferentes. Em seguida, define-se uma estrutura, com rótulo pré-definido pela biblioteca, chamada de *parallel_for*. O *parallel_for* recebe como entrada o *blocked_range*, o índice de início e o de fim do processamento, e o nome do método que será executado em cada núcleo com uma parte dos dados.

No método de treinamento paralelizado, não é necessário que as imagens de não faces sejam recortadas previamente. Os recortes são realizados durante o treinamento. Sempre que é necessária uma nova amostra de não face, ela é recortada da imagem correspondente na memória. Sempre que um novo recorte é obtido, são armazenadas na memória suas coordenadas, largura e altura e a escala do recorte. Portanto, o novo processo de obtenção de recortes de não faces pode ser resumido como segue.

Em primeiro lugar, deve existir um arquivo nos discos de todos os computadores do *cluster* contendo o nome de todas as imagens de não faces disponíveis para recorte. No início do treinamento, o computador mestre divide as imagens de não face entre os computadores do *cluster* e cada computador carrega uma imagem de seu conjunto.

Para permitir o recorte sucessivo das imagens de não faces, é realizado um deslizamento semelhante àquele que é feito no processo de detecção de faces, de modo que cada computador do *cluster* deve manter algumas variáveis atualizadas: nome da imagem que está sendo usada para recorte, coordenadas x e y do último recorte realizado, largura do último

recorte e escala. Sempre que um novo recorte é realizado, essas variáveis são atualizadas. O deslizamento é incrementado inicialmente no eixo x e, quando atinge o limite de colunas, este passa a ter valor zero e o y é incrementado.

A janela de deslizamento é inicializada com o valor mínimo de recorte (por exemplo, resolução 19×19) pixels. Quando a janela de deslizamento atinge o canto inferior direito da imagem, seu tamanho é incrementado por um fator de escala. Para aumentar a variabilidade do conjunto de treinamento, as imagens das quais serão extraídas amostras de não faces contêm faces. Porém, todas as faces foram rotuladas manualmente. Quando uma nova janela está sendo processada, a região de recorte passa por um processo de verificação para avaliar se contém uma face. Se a janela avaliada contiver uma face, essa janela será descartada e uma nova janela é avaliada. Esse processo de verificação permite que regiões contendo cabelo, pele, roupa, barba, olhos, nariz, boca, orelhas ou partes deles façam parte do conjunto de treinamento de não faces. A métrica usada para aceitar ou não um recorte de face é uma verificação de interseção, seguida do cálculo das áreas de retângulos. Um recorte é aceito se não houver interseção com uma região de face ou, caso haja interseção, a área de recorte for inferior a 30% da área da face. Essa porcentagem foi pré-definida a fim de evitar que metade de uma face, por exemplo, seja usada como não face, pois isso poderia confundir o classificador, tendo em vista que algumas características que distinguem uma face frontal de uma não face também distinguem uma face em perfil de uma não face.

4.3 Considerações Finais

Neste capítulo foram, descritos os principais problemas associados a um detector de faces, a saber: oclusão, orientação e iluminação. Adicionalmente, foram apresentadas algumas abordagens que têm sido propostas para tratar esses problemas. Um problema que está se tornando comum em treinamentos de classificadores para a detecção de faces é a necessidade de utilização de grandes quantidades de imagens, na ordem de milhões ou bilhões.

Conforme será apresentado no Capítulo 6, alguns dos treinamentos realizados nos experimentos desta tese utilizaram bilhões de imagens. Seria impraticável a realização de experimentos com tantas imagens sem o uso de um supercomputador ou de um cluster, razão

pela qual nesta tese é proposto um método híbrido de paralelização para o treinamento de cascatas de classificadores fracos.

Uma nova abordagem para a detecção de faces multipose foi apresentada neste capítulo, a qual é ilustrada nas Figuras 4.1 e 4.2. A originalidade da abordagem é a exploração da invariância por treinamento para gerar uma árvore de classificadores para detecção de faces que possui menor complexidade que outras abordagens existentes, tais como as de: Rowley, Baluja e Kanade (1998b), Jones e Viola (2003) e Huang et al. (2007). Além disso, foi proposto um método para paralelização do paradigma de treinamento de classificadores proposto por Viola e Jones (2001).

Capítulo 5

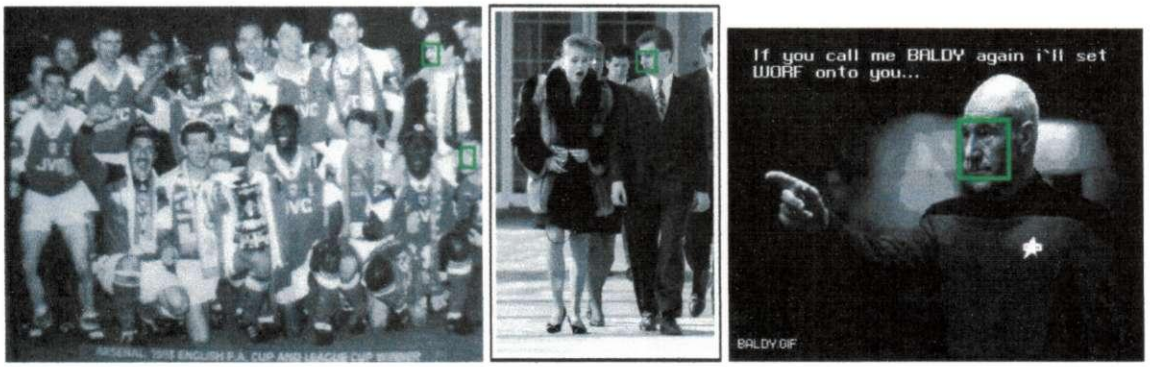
Avaliações Experimentais de Detecção de Faces

Neste capítulo, são apresentados e discutidos os resultados experimentais de avaliação do detector proposto. Além disso, também são realizadas comparações com outros detectores. Os detectores comparados são: o detector proposto por Viola e Jones (2004), o detector proposto por Rowley, Baluja e Kanade (1998a) e o detector disponível para uso online no sítio eletrônico *www.face.com* (esse detector doravante será chamado de *FaceDotCom*). As comparações foram realizadas utilizando a base CMU-MIT para teste. Posteriormente, o detector proposto foi avaliado utilizando a base FDDB (*Face Detection Data Base*). Para essa base, o detector proposto foi comparado apenas com o *FaceDotCom* e com os detectores que forneceram dados para o sítio eletrônico da FDDB.

5.1 Avaliação do Detector na Base CMU-MIT

Foram realizados vários treinamentos de cascatas de classificadores com a intenção de selecionar o conjunto de parâmetros que gerasse uma cascata com o melhor desempenho quando testada sobre a base CMU-MIT, composta por 130 imagens e acompanhada por um arquivo contendo as marcações de pontos importantes das faces contidas na base. A quantidade de faces marcadas é 511, porém no artigo de Viola e Jones (2004), os autores afirmam que são

apenas 507, sem explicar a razão dessa diferença. A partir de uma análise minuciosa das imagens das imagens da base CMU-MIT, verificou-se a existência de marcação para 4 faces de perfil. Para tornar a comparação mais coerente, essas quatro marcações de faces de perfil foram desconsideradas. Na Figura 5.1, são apresentadas as imagens que contêm faces de perfil.



Arsenal.gif *oksana1.gif* *JLpicard.Baldy.gif*

Figura 5.1: Imagens que possuem faces de perfil que são desconsideradas na contagem de acerto.

O objetivo do primeiro conjunto de experimentos foi *verificar a influência do tipo de recorte da imagem de face*, de modo que foram realizados 5 treinamentos em que os parâmetros eram os mesmos, mas o tipo de recorte das imagens de faces eram diferentes. Partindo do recorte original, foram feitas variações adicionando-se ou retirando-se informação de fundo (*background*). Neste trabalho, as diferentes expansões/contrações da região de recorte serão chamadas de abrangência do recorte e, independentemente da abrangência usada, a resolução da imagem recortada é a mesma.

Assim, pode-se chamar o recorte original de abrangência 0 e os outros passos são chamados de abrangência -1, -2, 1 e 2. O recorte de abrangência 0 é obtido simplesmente a partir do recorte da imagem original usando as coordenadas obtidas manualmente no processo de marcação e, em seguida, redimensionando a imagem recortada para o tamanho desejado. Para as avaliações de tamanho de recorte, a resolução das imagem é de 21×21 pixels. Tomando-se como unidade de abrangência a razão entre a largura da face marcada e a largura para a qual a face será redimensionada, os novos recortes são realizados diminuindo-se ou aumentado-se a região de recorte por **n** abrangências. Na Figura 5.2, são apresentados os

cinco tipos de recortes para uma mesma imagem.

Abrangência −2 Abrangência −1 Abrangência 0 Abrangência 1 Abrangência 2

Figura 5.2: Imagens de faces recortadas com diferentes abrangências.

Os treinamentos foram realizados utilizando 1000 imagens de faces e 2000 imagens de não faces por estágio. Foram treinados 20 estágios para cada cascata, extraindo características do tipo *Haar estendido* e usando o método de *boosting GentleBoost*. O termo *haar estendido* refere-se à utilização de características do tipo *Haar* rotacionadas de 45°, conforme proposto por Lienhart, Kuranov e Pisarevsky (2002). Após os treinamentos, as cascatas foram avaliadas com a base CMU-MIT. No procedimento de avaliação dos resultados, foram adotadas as métricas mencionadas por Lienhart, Kuranov e Pisarevsky (2002), segundo as quais uma detecção é considerada correta se:

- a distância euclidiana entre o centro da face detectada e centro de uma face marcada for menor que 30% da largura da face marcada; e
- a largura da face detectada for até, no máximo, 50% maior ou, no mínimo, 50% menor do que a largura da face marcada.

Na Figura 5.3, apresentam-se as curvas ROC dos testes realizados com a base CMU-MIT usando diferentes recortes de faces nos treinamentos. O parâmetro limiarizado para obtenção das curvas foi o limiar do estágio da árvore de decisão usada como classificador fraco. As curvas mostram claramente uma melhoria dos resultados de detecção de faces à medida que o recorte engloba uma região maior do fundo da imagem. Então, tem-se aqui uma validação experimental para a detecção de faces de algo que Sinha et al. (2006) já havia mencionado: a área externa da face é importante para a tarefa de reconhecimento facial. Observando a Figura 5.3 nota-se que o ganho em relação à taxa de verdadeiros positivos do recorte original para o recorte com abrangência de duas unidades a mais foi maior do que 10%.

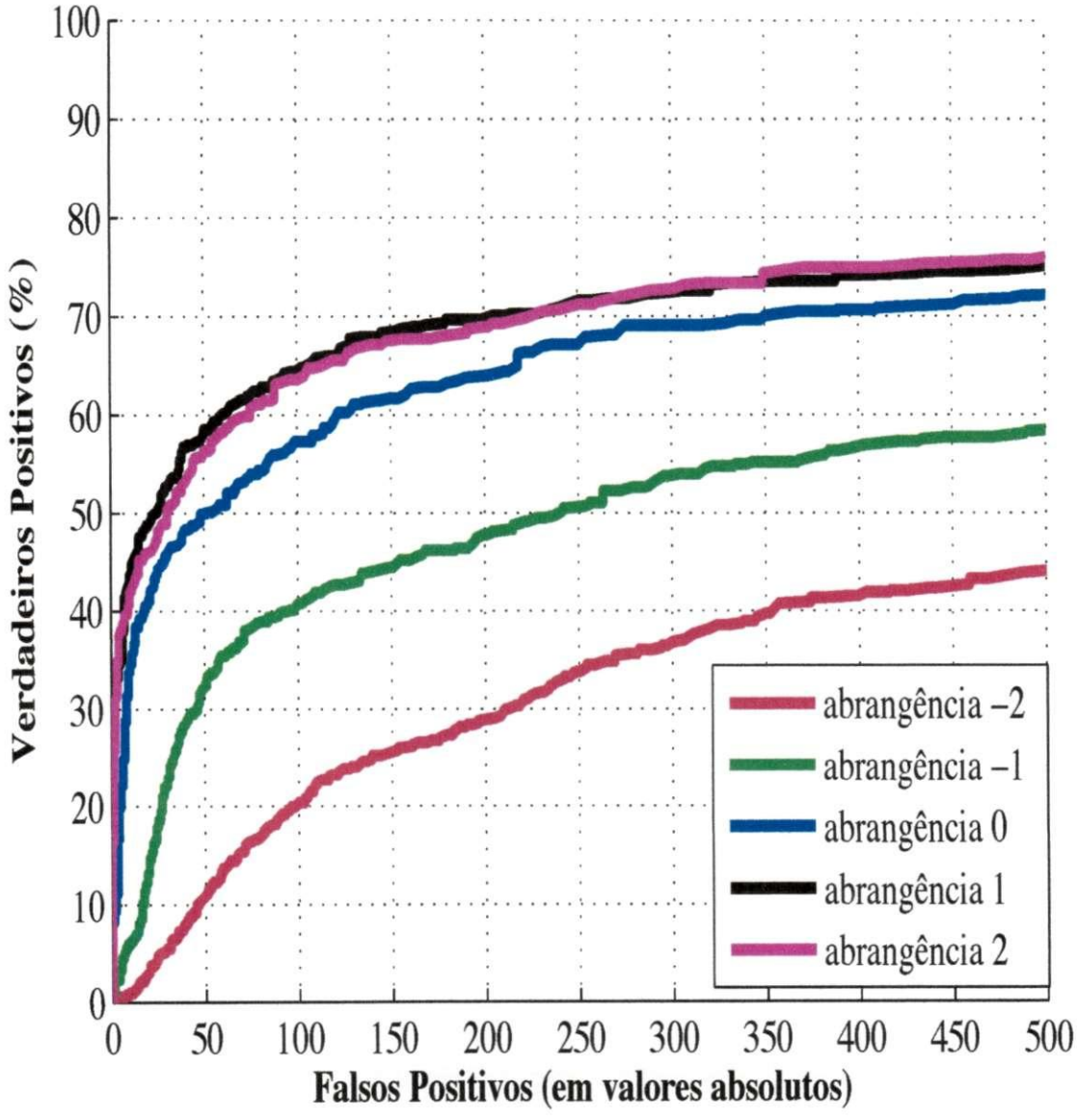


Figura 5.3: Curvas ROC para os testes com treinamentos usando diferentes tipos de recorte de face.

O experimento ora discutido comprova que incluir informação da área externa da face auxilia o classificador na tarefa de detecção. Viola e Jones (2004) mencionam que realizaram treinamentos com imagens de resolução 16×16 mas não obtiveram bons resultados, pois os recortes eram muito ajustados às faces de forma que testa e queixo ficavam de fora, razão pela qual optaram por um recorte que inclui a cabeça inteira e possui resolução 24×24 pixels.

Viola e Jones (2004) não apresentam resultados de experimentos sistemáticos que comprovem que sua escolha de resolução é a melhor. Lienhart, Kuranov e Pisarevsky (2002) fazem alguns comentários sobre as resoluções usadas para treinar classificadores de faces e afirmam que as resoluções presentes na literatura especializada variam (na época em que o artigo foi escrito) de 16×16 a 32×32 pixels.

Constatando a carência de investigações que analisassem o problema da resolução de treinamento, Lienhart, Kuranov e Pisarevsky (2002) realizaram alguns experimentos nesse sentido, constatando que a melhor resolução, dentre aquelas avaliadas para testes com a base CMU-MIT, era de 20×20 pixels. As resoluções avaliadas foram: 18×18, 20×20, 24×24, 28×28 e 32×32 pixels.

Embora os resultados anteriores tenham sido promissores, ainda são inferiores aos apresentados por Viola e Jones (2004) e Rowley, Baluja e Kanade (1998a). Esses autores reportam resultados de verdadeiros positivos acima de 90% e falsos positivos inferiores a 500 (em valores absolutos). Com o intuito de obter melhores resultados do que aqueles apresentados no gráfico da Figura 5.3, foi realizado um novo treinamento. O novo treinamento utilizou 13000 imagens de faces e 26000 imagens de não faces por estágio de treinamento.

Na Figura 5.4, são apresentadas três curvas ROC para os resultados de detecção de faces usando a base CMU-MIT. Os detectores avaliados são: a implementação da abordagem proposta nesta tese, treinada com imagens de resolução 21×21 pixels e abrangência 2 e os detectores propostos por Viola e Jones (2004) e Rowley, Baluja e Kanade (1998a). Deve ser ressaltado que os resultados dos outros detectores foram obtidos de uma tabela publicada no artigo de Viola e Jones (2004). Isso justifica o fato de as curvas ROC desses detectores não estarem completas, há apenas alguns pontos das curvas ROC na tabela do artigo, os outros são obtidos por interpolação.

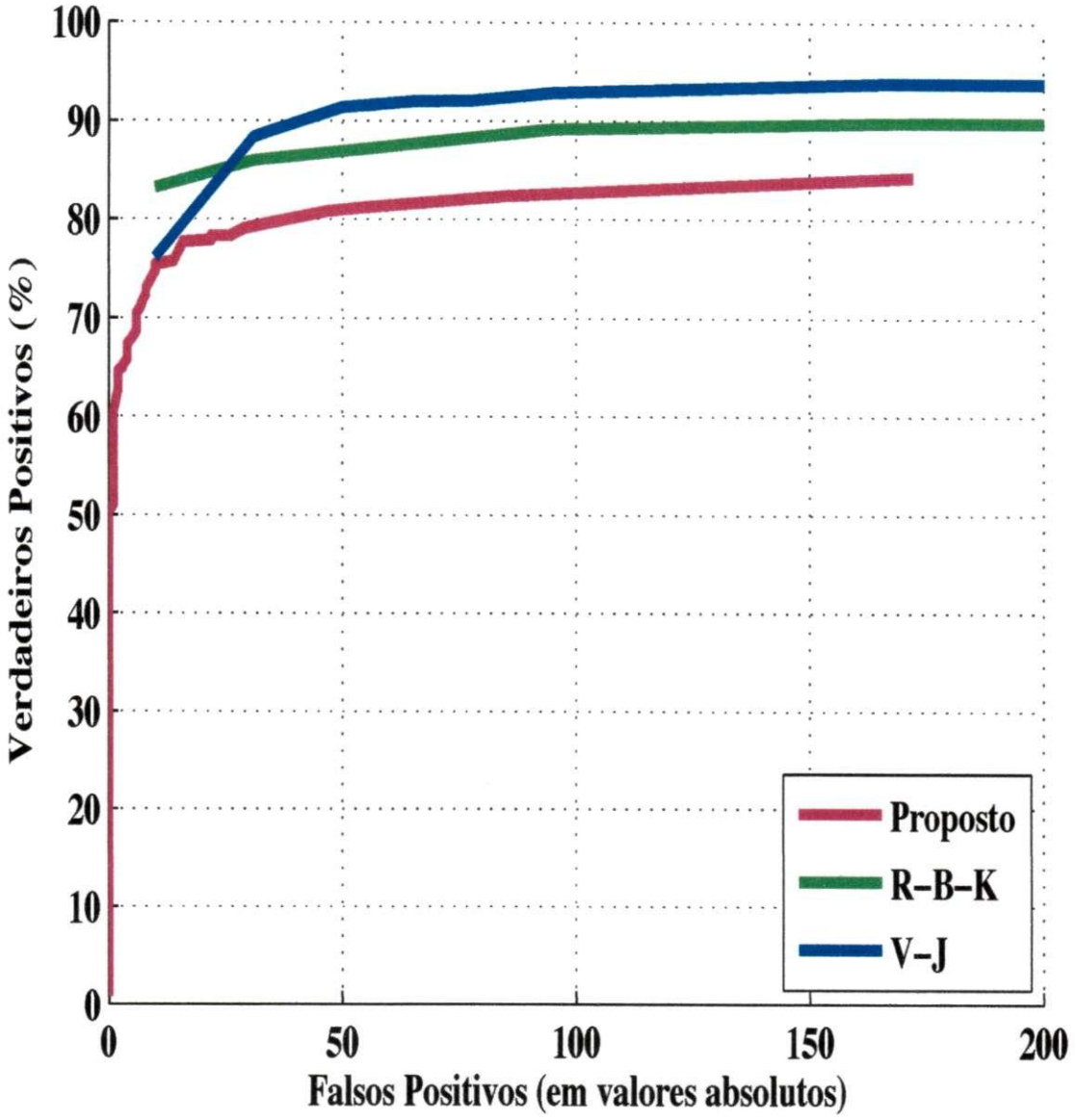


Figura 5.4: Comparações entre a abordagem proposta treinada com 13000 imagens de faces de resolução 21 × 21 pixels e as abordagens de Viola e Jones (2004) e Rowley, Baluja e Kanade (1998a) testadas na base CMU-MIT.

Apesar de o treinamento da cascata ter sido realizado com 13000 imagens de faces, ainda assim o detector obteve resultados inferiores aos da literatura, conforme pode ser visto na Figura 5.4. Tais resultados suscitaram alguns questionamentos, tais como:

- Há faces na base de teste com resolução inferior a 21×21 pixels?
- A qualidade das imagens de treinamento é compatível com a qualidade das imagens de teste?
- Quais os casos em que o detector proposto erra? Existe algum padrão de erro?

Para responder a esses questionamentos as imagens de teste foram analisadas minuciosamente, tendo sido constatados os seguintes fatos:

- Há faces de perfil marcadas no *ground truth*, o que torna a avaliação de detectores de faces frontais inconsistente. Porém, isso já foi mencionado e a solução é desconsiderar as 4 faces de perfil da contagem, conforme deve ter sido feito por Viola e Jones (2004) e Rowley, Baluja e Kanade (1998a);
- A base contém faces frontais que não foram levadas em consideração pelo *ground truth*. A priori, isso não constitui um problema para a taxa de verdadeiros positivos, apenas para a taxa de falsos positivos;
- Há faces incluídas no *ground truth* que não são completamente visíveis na imagem, por estarem cortadas. A argumentação para mantê-las na contagem é semelhante à argumentação do tópico anterior;
- Há uma grande quantidade de desenhos de faces considerados pelo *ground truth*. Esse seria apenas um fator complicador para a detecção, não afetando a consistência da avaliação dos detectores; e
- Por fim, a qualidade das imagens é muito baixa. Em sua maioria, as imagens são fotografias de jornais e revistas antigas ou imagens capturadas de telas de TV. Em alguns casos, como nas fotografias de times de futebol, há imagens em que as faces apresentam alto grau de degradação, caso em que seriam consideradas faces com oclusão.

Os treinamentos dos classificadores que geraram as curvas ROC mostradas na Figura 5.4 foram realizados usando imagens de resolução 21 × 21 pixels. Uma análise minuciosa da base CMU-MIT indicou que há várias faces com resolução inferior a 21 × 21 pixels. Esse fator foi decisivo para que fossem realizados treinamentos com imagens de faces em resolução menor, a resolução escolhida foi 19 × 19 pixels. Essa resolução foi escolhida por vários motivos. Em primeiro lugar, outros autores (LIENHART; KURANOV; PISAREVSKY, 2002) já haviam testado outras resoluções, conforme já foi mencionado anteriormente. Dentre as resoluções testadas por esses autores, aquela que obteve melhor resultado foi a resolução 20 × 20 pixels e a menor resolução testada foi 18 × 18 pixels. Como esta tese tem como um dos objetivos estudar invariância à rotação, deveria ser usada uma resolução com valores ímpares para que fosse possível fazer uma rotação a partir do centro da imagem. Como outros autores (LIENHART; KURANOV; PISAREVSKY, 2002) já haviam mencionado que resoluções superiores a 20 × 20 pixels não obtiveram resultados melhores que resoluções inferiores, a escolha de resolução de valor ímpar, inferior a 20 × 20 pixels, que não havia sido testada pelos autores supramencionados levou à escolha da resolução 19 × 19 pixels.

Inicialmente, foram treinadas duas cascatas com imagens de resolução 19 × 19 pixels. As curvas ROC geradas da utilização dessas cascatas para detectar faces da base CMU-MIT são mostradas na Figura 5.5. Além disso, também são apresentadas nessa figura duas curvas treinadas com imagens de resolução 21 × 21 pixels, um treinamento com 1000 imagens de faces e outro com 2000 imagens de faces.

O que se observa nas curvas ROC da Figura 5.5 é que há uma melhoria expressiva dos resultados obtidos com a cascata resultante do treinamento com imagens de resolução menor, em relação aos resultados obtidos com a cascata treinada com a mesma quantidade de imagens, porém com resolução mais alta. No gráfico da Figura 5.5, os resultados são rotulados como $19 \times 19 - 1kpos$ e $21 \times 21 - 1kpos$ (o rótulo *1kpos* indica que foram empregadas 1000 imagens de faces por estágio de treinamento), respectivamente (a curva azul apresenta resultados melhores que a curva vermelha). O melhoramento do desempenho com o aumento da quantidade de imagens é esperado, desde que as imagens sejam representativas das classes. A melhoria dos resultados evidencia a qualidade da base usada no treinamento e ocorre porque as imagens não são redundantes ou dependentes.

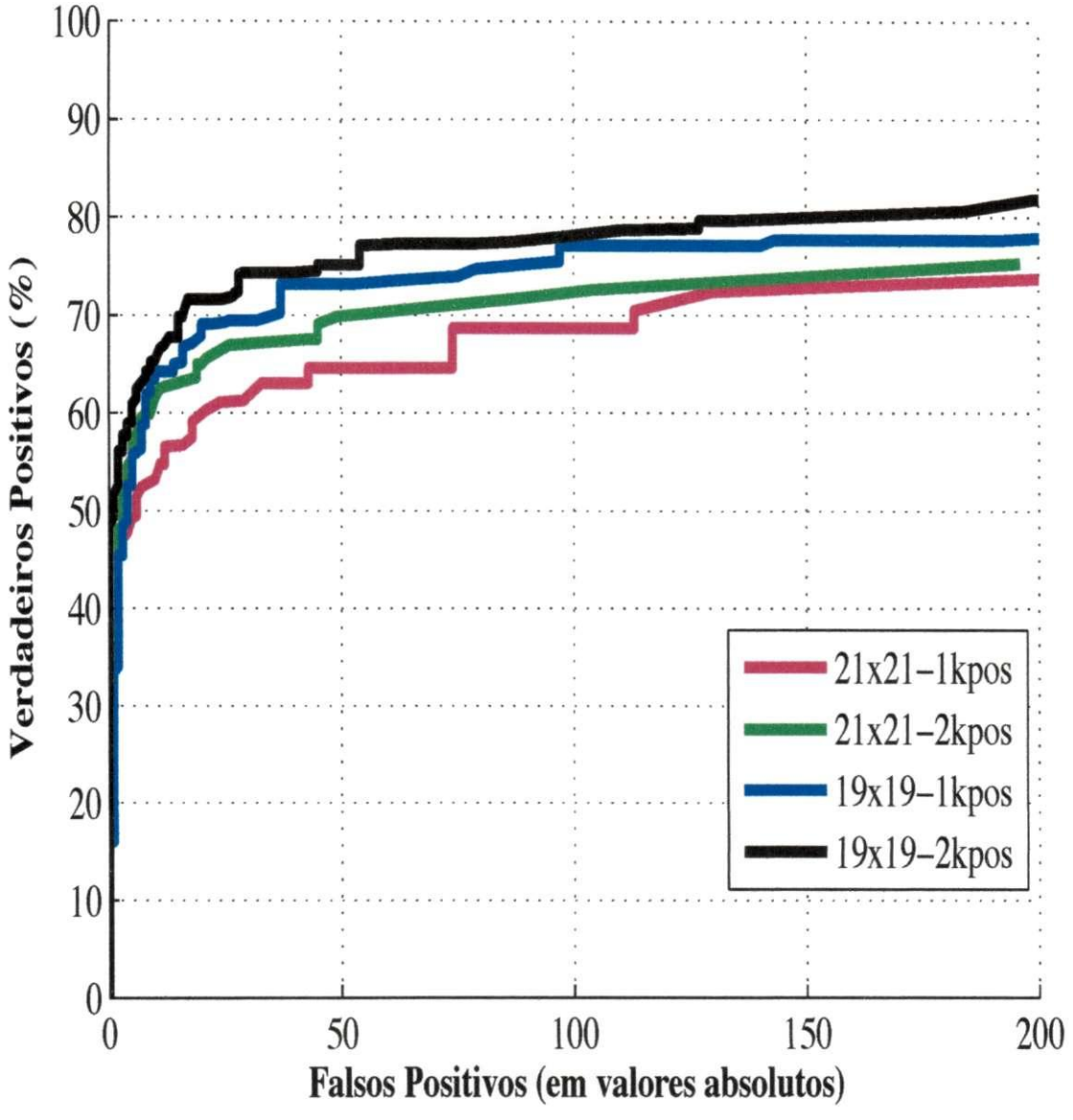


Figura 5.5: Comparações entre cascatas treinadas com resoluções 19×19 e 21×21 para a técnica proposta treinada para faces frontais.

Para averiguar a hipótese de que o aumento da quantidade de imagens de treinamento melhora os resultados, foram realizados quatro treinamentos com quantidades de imagens diferentes e com todos os outros parâmetros iguais. Na Figura 5.6, são apresentadas quatro curvas ROC, cada uma das quais resultante do teste de uma das cascatas tendo como entrada a base CMU-MIT. A observação das curvas explicita o fato de que o aumento da quantidade de imagens realmente melhora os resultados de detecção.

Há mais um fator a ser discutido em relação aos resultados apresentados na Figura 5.6: a curva ROC da cascata treinada com 4000 imagens de faces não atingiu resultados superiores a 90% de verdadeiros positivos. No artigo de Viola e Jones (2004), os autores apresentam resultados superiores a 91% após 50 falsos positivos. Além disso, os autores afirmam terem treinado suas cascatas usando apenas 4916 imagens de faces com resolução 24×24 pixels. Portanto, o questionamento decorrente dessa discussão seria: por que os resultados da cascata treinada com 4000 imagens de faces (a curva preta da Figura 5.6) são inferiores aos resultados de Viola e Jones (2004)?

Como já foi discutido anteriormente, as imagens de face da base CMU-MIT possuem características peculiares, como baixa qualidade e baixa resolução. Então, para lidar com esses fatores, foram coletadas 249 novas imagens que contém faces, diferentes das anteriormente utilizadas nos treinamentos. Foram selecionadas imagens que contivessem faces com pelo menos uma das seguintes características:

- Apresentar baixa resolução;
- A face ser desenhada ou pintada;
- A face estar contida em uma fotografia escaneada ou muito antiga.

Na Figura 5.7, há alguns exemplos de imagens contendo faces consideradas de baixa qualidade. É necessário usar esse tipo de imagens no treinamento, porque há imagens de face na base CMU-MIT, as quais na verdade, são desenhos feitos à mão, ou imagens que possuem um sombreamento tão forte que a região de face sombreada torna-se na verdade um *borrão* sem padrão, o que as faz serem classificadas mais como oclusão do que como sombra propriamente dita.

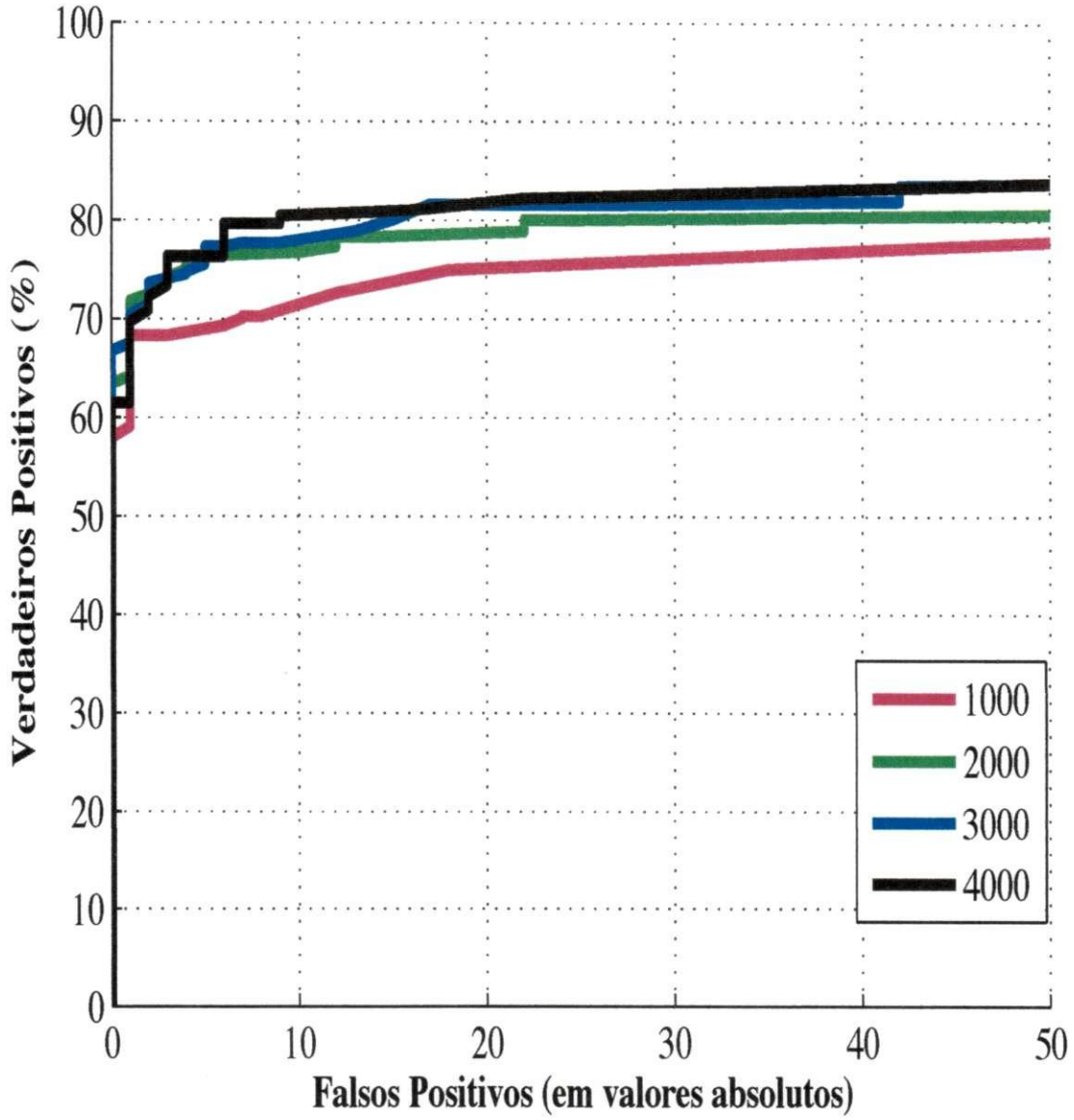


Figura 5.6: Comparações entre cascatas treinadas com diferentes quantidades de imagens e resolução 21 × 21. As quantidades de imagens de faces foram: 1000, 2000, 3000 e 4000. As quantidades de imagens de não faces são o dobro da quantidade de imagens de faces para cada estágio de treinamento.

Figura 5.7: Amostras de imagens de baixa qualidade incorporadas ao conjunto de treinamento.

Para auxiliar o entendimento de quão baixa é a qualidade de algumas imagens da base CMU-MIT, são apresentados alguns exemplos de imagens da referida base na Figura 5.8. Na primeira imagem há a foto de um quadro branco contendo nove desenhos de faces feitos à mão. As próximas três imagens não fazem parte da base CMU-MIT, são apenas recortes da primeira imagem com maior *zoom* para facilitar a visualização. Em seguida, há uma imagem do time da Colômbia, na copa do mundo de futebol de 1993. Nessa imagem, foram marcadas com retângulo verde duas faces problemáticas que são apresentadas, posteriormente recortadas em escala aumentada. Essas duas faces apresentam o problema denominado anteriormente de *borrão*: a iluminação é tão forte que se torna praticamente uma oclusão.

Na Figura 5.8, há mais duas imagens com desenhos feitos em quadro branco, uma imagem contendo um *klingon*[1], com a face destacada com retângulo verde e, posteriormente, recortada para melhor visualização. Além disso, há um desenho de uma suposta face, bastante complicado para um detector de faces classificar corretamente se não tiver sido treinado

[1]Raça guerreira da série *Star Trek* da década de 1960.

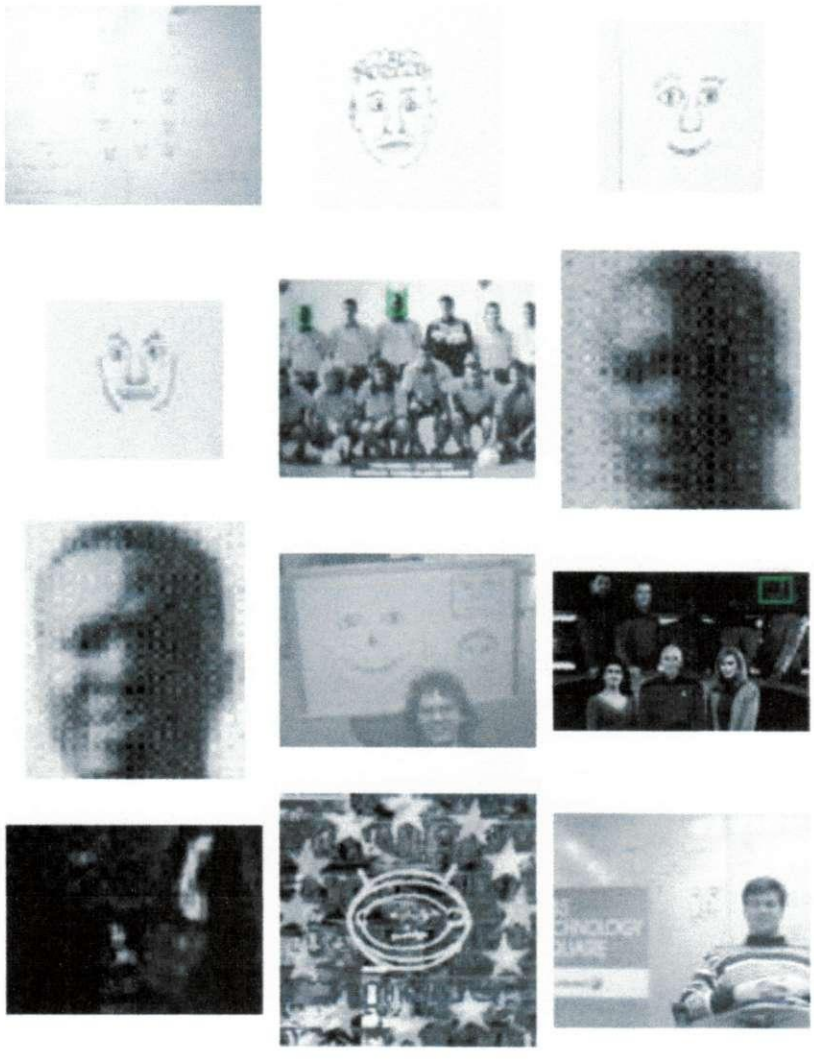


Figura 5.8: Amostra de imagens de baixa qualidade da base CMU-MIT.

com padrões dessa natureza.

Um total de 657 novas imagens de faces foram recortadas das imagens obtidas da Web contendo imagens de baixa qualidade e de padrão mais próximo das imagens da base CMU-MIT. Levando em consideração que as imagens de faces são rotacionadas em ± 10 graus, foi produzido um total de 1971 novas imagens. Um novo treinamento foi realizado utilizando essas imagens, além daquelas que já estavam sendo utilizadas. O novo treinamento empregou imagens redimensionadas para a resolução 19×19 pixels. Os resultados da avaliação são mostrados na Figura 5.9.

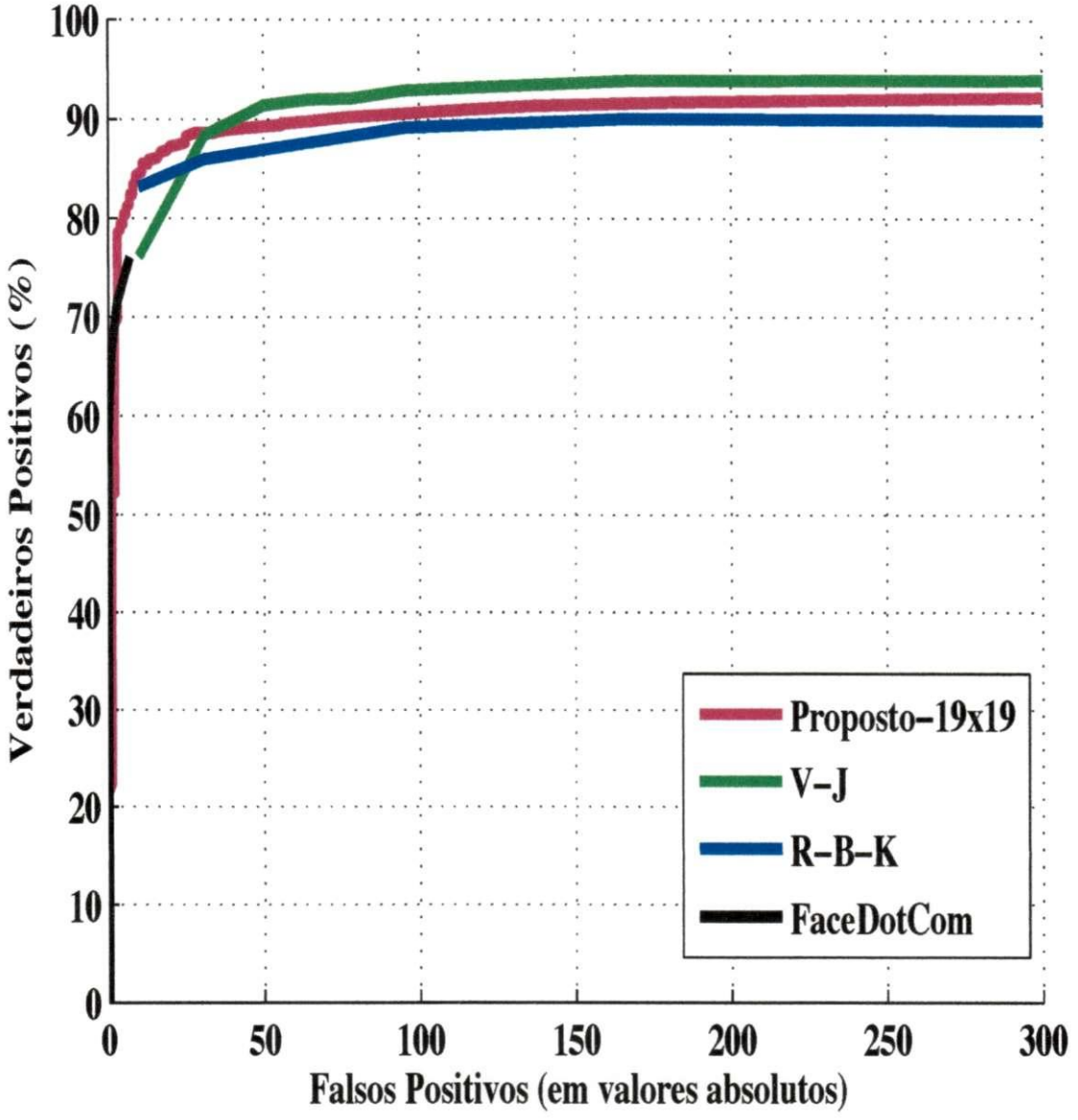


Figura 5.9: ROC para classificador treinado com imagens de resolução 19 × 19 e baixa qualidade testado na base CMU-MIT.

A região de interseção das curvas apresentadas na Figura 5.9 é apresentada na Figura 5.11. As curvas com resultados do detector *faceDotCom* não se expandem para uma maior quantidade de falsos positivos devido à pouca flexibilidade de variação de parâmetros da API (Interface de Programação de Aplicação - *Application Programming Interface*). A configuração de parâmetros usada na aplicação do *faceDotCom* foi a fornecida como padrão, que produz baixas taxas de falsos positivos e, consequentemente, uma taxa inferior de verdadeiros positivos.

Página inicial

Figura 5.10: Página do sítio eletrônico *http://face.com*.

No gráfico da Figura 5.9, há resultados de um detector que não havia sido descrito até o momento, o detector *FaceDotCom*. Esse detector de faces é, na verdade, uma página web hospedada no endereço *http://face.com* que fornece uma API (*Application Programming Interface*) para processamento de imagens de faces online. Podem-se usar os recursos do *FaceDotCom* de duas formas: submetendo imagens para processamento na página ou criando um script PHP para o envio de imagens e recebimento dos dados processados. Na Figura 5.10, são apresentadas a página inicial e a página que contém as ferramentas para processamento de imagens de faces.

O *FaceDotCom* vai muito além de detectar faces, sendo capaz de reconhecê-las, localizar

seus pontos fiduciais, fornecer informações sobre gênero, humor, expressão facial, idade, uso de óculos, situação dos lábios (fechados ou abertos) e ângulo de orientação da face nas três direções (*roll*, *yaw* e *pitch*). Nesta tese, será avaliada apenas sua funcionalidade de detecção de faces. O sítio eletrônico não fornece detalhes de como funciona o processo de detecção de faces no *FaceDotCom*, apenas apresenta algumas informações sobre como é realizado o reconhecimento de faces em um relatório técnico (TAIGMAN; WOLF, 2011). Segundo o relatório, são gerados modelos 3D baseados em forma (*shape*) das faces que serão reconhecidas e modelos discriminativos gerados a partir de bilhões de imagens de faces.

Pode-se observar, nas Figuras 5.9 e 5.11, que o detector treinado com imagens de menor resolução obteve melhores resultados do que todos os outros detectores avaliados, para a região das curvas em que a quantidade de falsos positivos é inferior a 25. Além disso, os resultados do treinamento com imagens de resolução 19×19 pixels são superiores aos resultados obtidos com o classificador treinado com imagens de resolução 21×21 pixels mostrados na Figura 5.5.

Nesta seção, vários fatores importantes para a implementação de um detector de faces foram explorados e as seguintes diretrizes podem ser sumarizadas. Em primeiro lugar, a análise criteriosa dos erros que o detector de faces comete pode auxiliar na melhoria da base de treinamento se imagens contendo as variações observadas na análise forem adicionadas. Além disso, a forma como as amostras de faces são recortadas interfere na precisão do classificador treinado e a adição de elementos externos da face (linhas do queixo e do cabelo, por exemplo) nas imagens recortadas para treinamento pode melhorar os resultados de detecção. Na próxima seção, são apresentados os resultados de detecção de faces avaliados com a base FDDB.

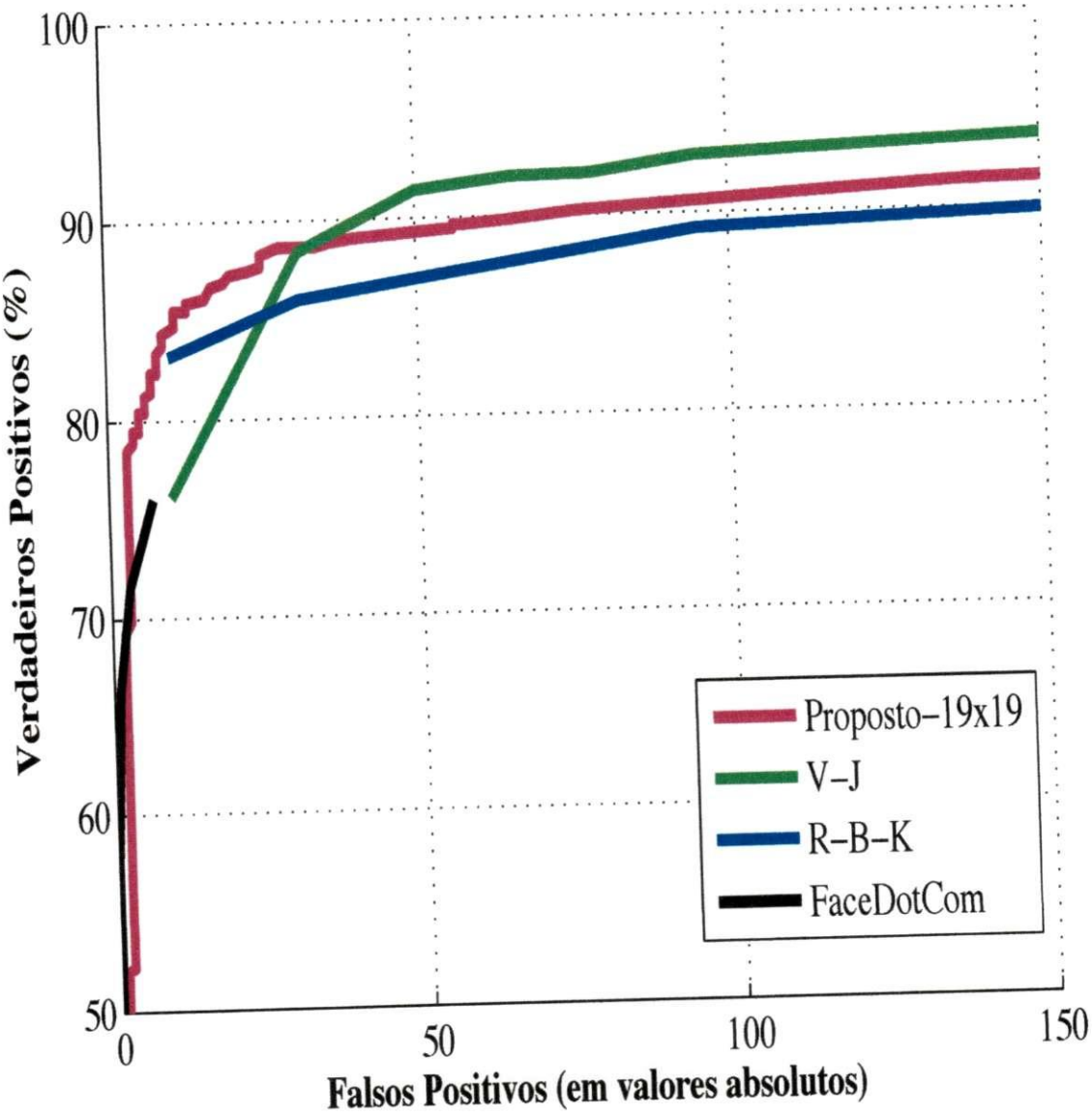


Figura 5.11: ROC para classificador treinado com imagens de resolução 19 × 19 e baixa qualidade, zoom na região de interseção das curvas.

5.2 Avaliação do Detector na Base FDDB

A base de imagens FDDB (Face Detection Data Base) (JAIN; LEARNED-MILLER, 2010), é um *benchmark* para a avaliação de detectores de faces, sem restrição de condições. Essa base vem acompanhada de um protocolo para a avaliação dos resultados obtidos pelos detectores aplicados em suas imagens. A principal motivação para a criação da FDDB foi a ausência de métodos coerentes para comparação de detectores de faces. Uma base de imagens tradicionalmente utilizada para a avaliação de detectores de faces é a CMU-MIT, que foi utilizada em sua versão final por Schneiderman e Kanade (2000a).

No entanto, até a criação da FDDB, as bases de imagens para a avaliação de detectores não exigiam, nem propunham, um método de avaliação. Essas bases eram compostas, simplesmente, por um conjunto de imagens acompanhadas de arquivos contendo as coordenadas das faces ou de alguns pontos fiduciais. Deve-se também levar em conta que alguns detectores de faces não são disponibilizados para que outros pesquisadores possam avaliá-los. Por exemplo, não existe código, nem mesmo o executável binário, disponível para a avaliação de um dos detectores mais populares da literatura especializada em detecção de faces, o detector que foi proposto por Viola e Jones (2004).

O autor desta tese entrou em contato com os autores Viola e Jones (2004) questionando a existência de alguma versão do programa para download e eles afirmaram que não há nenhuma versão disponível. A versão mais próxima do que é proposto por eles é a que foi implementada na biblioteca OpenCV (BRADSKY; KAEHLER, 2008). A biblioteca mencionada vem acompanhada de alguns arquivos xml para a detecção de faces. Porém, os resultados desses detectores são muito inferiores àqueles publicados por Viola e Jones (2004).

Além de dispor das imagens para a avaliação e do *groundtruth*, a FDDB fornece o código que será usado para contar os acertos e os erros. A medida usada para contar uma detecção como acerto é a razão entre a área de interseção e área de união entre a região detectada e a região anotada (JAIN; LEARNED-MILLER, 2010). As faces são anotadas na base como elipses, mas o código fornecido realiza as conversões necessárias para compatibilizar resultados de detecção que foram marcados como retângulos. Na Figura 5.12, são mostrados exemplos de imagens da base FDDB. Os autores apresentam como características da base:

- A grande quantidade de imagens e de faces: 2845 e 5171, respectivamente;
- Uma grande faixa de dificuldades: oclusões, poses, baixa resolução e faces fora de foco;
- A especificação de regiões de faces como elipses;
- Imagens coloridas e em tons de cinza.

Há duas formas de avaliar um detector com a base FDDB: validação cruzada com 10 dobras (*10-fold cross-validation*) e treinamento sem restrições. No primeiro caso, o desempenho cumulativo é relatado como a curva média das 10 curvas ROC. No segundo caso, é permitido o uso de imagens que não fazem parte da base para treinar os classificadores, mas neste caso o conjunto também é dividido em 10 partes e a curva ROC resultante é obtida a partir da média das curvas. Os resultados obtidos pelo detector proposto nesta tese, apresentados a seguir, foram obtidos utilizando o modo experimental sem restrições visto que todos os resultados apresentados no site da FDDB utilizaram tal modo de experimentação.

Outra peculiaridade da FDDB é que há duas métricas de avaliação: discreta e contínua. A métrica discreta conta como acerto toda detecção que obtiver o valor da razão entre área de interseção e área de união maior do que 0,5. A métrica contínua atribui um escore à detecção equivalente ao valor da razão entre as áreas de interseção e união. Nas Figuras 5.13 e 5.16 são mostrados resultados obtidos com as métricas contínua e discreta, respectivamente. Nas Figuras 5.14, 5.15, 5.17 e 5.18 são apresentadas com resolução mais alta algumas regiões dos gráficos dos resultados das avaliações contínua e discreta.

Um fato inesperado ocorreu durante a avaliação do detector proposto pela base FDDB. Os resultados foram bastante inferiores àqueles apresentados por outros detectores fornecidos na página web da base. No entanto, uma inspeção visual das imagens com faces marcadas pelo detector contradizia os resultados numéricos, o que levou à hipótese de que a marcação das faces detectadas não estava coerente com o tipo de avaliação pela qual os resultados do processo de detecção deveriam passar. Ou seja, como as faces estavam sendo definidas por regiões muito ajustadas às faces e a avaliação do protocolo FDDB leva em consideração a área da face detectada, foram realizados experimentos aumentando a região das faces após a detecção.

Figura 5.12: Exemplos de imagens da base FDDB.

Na página web de resultados da FDDB, há curvas ROC para detectores de faces que possuem artigos publicados e resultados sem publicações relacionadas. Os gráficos das curvas ROC são apresentados separadamente, um par de gráficos (com métricas contínua e discreta, respectivamente) para os detectores cujos métodos foram publicados e um par de gráficos para os detectores cujas abordagens não foram publicadas. Dentre os detectores cujas abordagens foram publicadas, aquele que obteve os melhores resultados foi o que foi proposto por Jain e Learned-Miller (2011).

Nas Figuras 5.19 e 5.20, os referidos resultados são rotulados como "jain", e são apresentados resultados de detecção de faces para as duas métricas de avaliação para os quais a única diferença é o aumento da área da região da face por uma unidade, aqui chamada de abrangência. Observa-se que a abrangência que apresenta melhores resultados é a abrangência de 5 unidades. Além disso, aqui está uma das contribuições desta tese. A comprovação experimental de que antes de avaliar um detector de faces usando como métrica a razão entre área de interseção e área de união das regiões detectada e de *groundtruth* deve-se verificar se a marcação das faces detectadas está coerente com a marcação de *groundtruth*. Nesta seção

as abrangências são calculadas de modo similar ao que foi apresentado na Seção 5.1, o que deve ficar claro é que naquela seção as abrangências são usadas na fase de treinamento, nesta seção são usadas na fase de teste. As curvas rotuladas como "jain" referem-se aos resultados apresentados no sítio eletrônico da FDDB [2] que obtiveram os melhores resultados.

Os resultados mostrados nas Figuras 5.20 e 5.21 foram obtidos pelo melhor detector frontal *upright* (com variação de ângulos de ± 10 graus), treinado segundo a abordagem proposta, pelo detector *FaceDotCom* e pelo detector de Jain e Learned-Miller (2011) (que são os autores da FDDB). Esses resultados podem parecer incoerentes ou muito baixos. Contudo, a base é composta por imagens de faces em diferentes poses e, com exceção do *FaceDotCom*, os detectores avaliados foram treinados apenas com imagens de faces frontais sem rotações extremas.

A comparação com os resultados do *FaceDotCom* não é muito justa, uma vez que este foi desenvolvido para detectar poses diferentes da frontal. Além do mais, não se sabe com qual base os outros dois detectores foram treinados, os quais podem ter usado algumas faces contidas na FDDB para treinamento, enquanto o detector proposto não usou nenhuma imagem da base de teste para treinamento.

[2] http://vis-www.cs.umass.edu/fddb/results.html

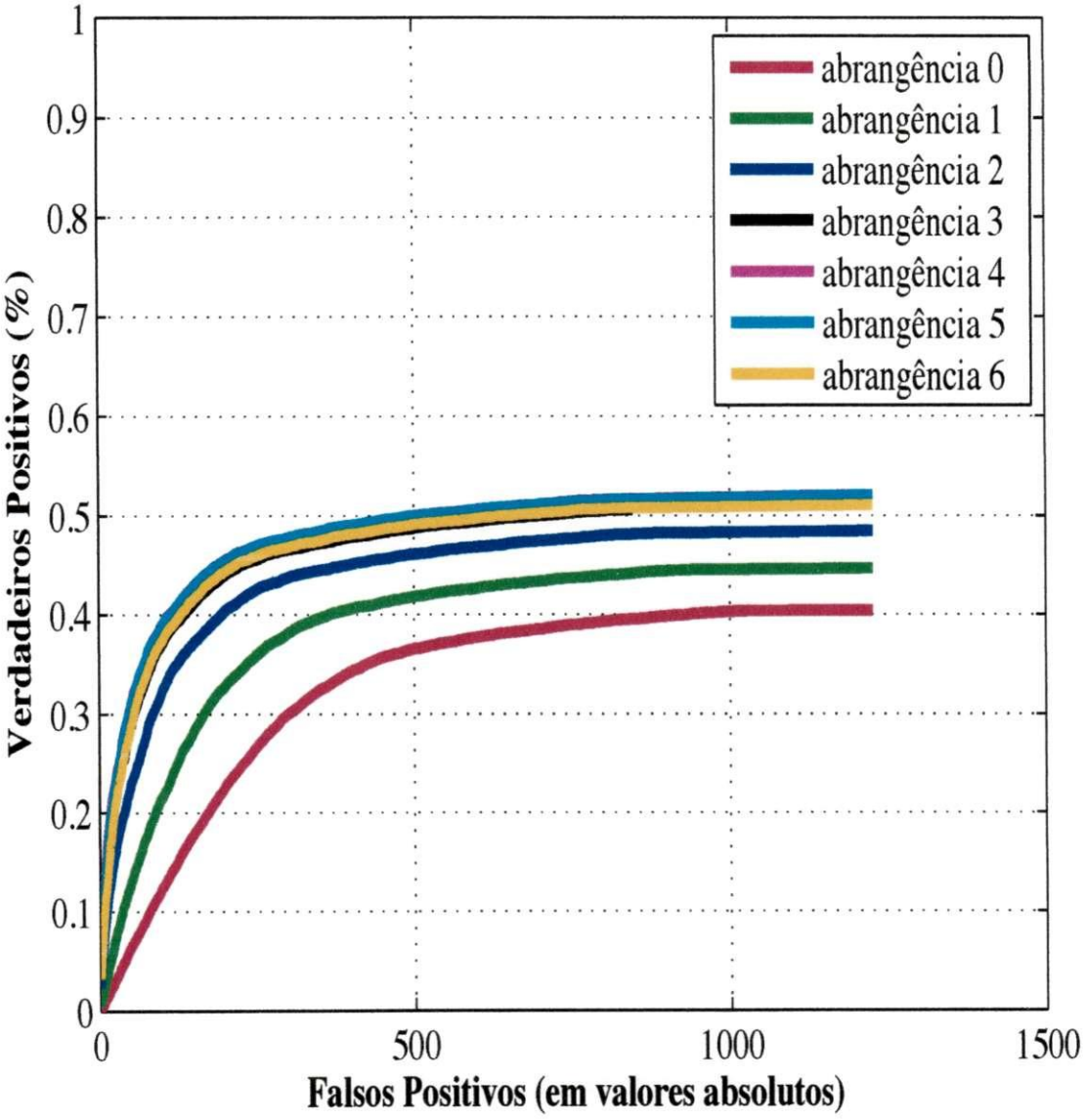


Figura 5.13: Gráfico em visão completa da comparação de várias abrangências de marcação da face detectada na base FDDB com métrica contínua.

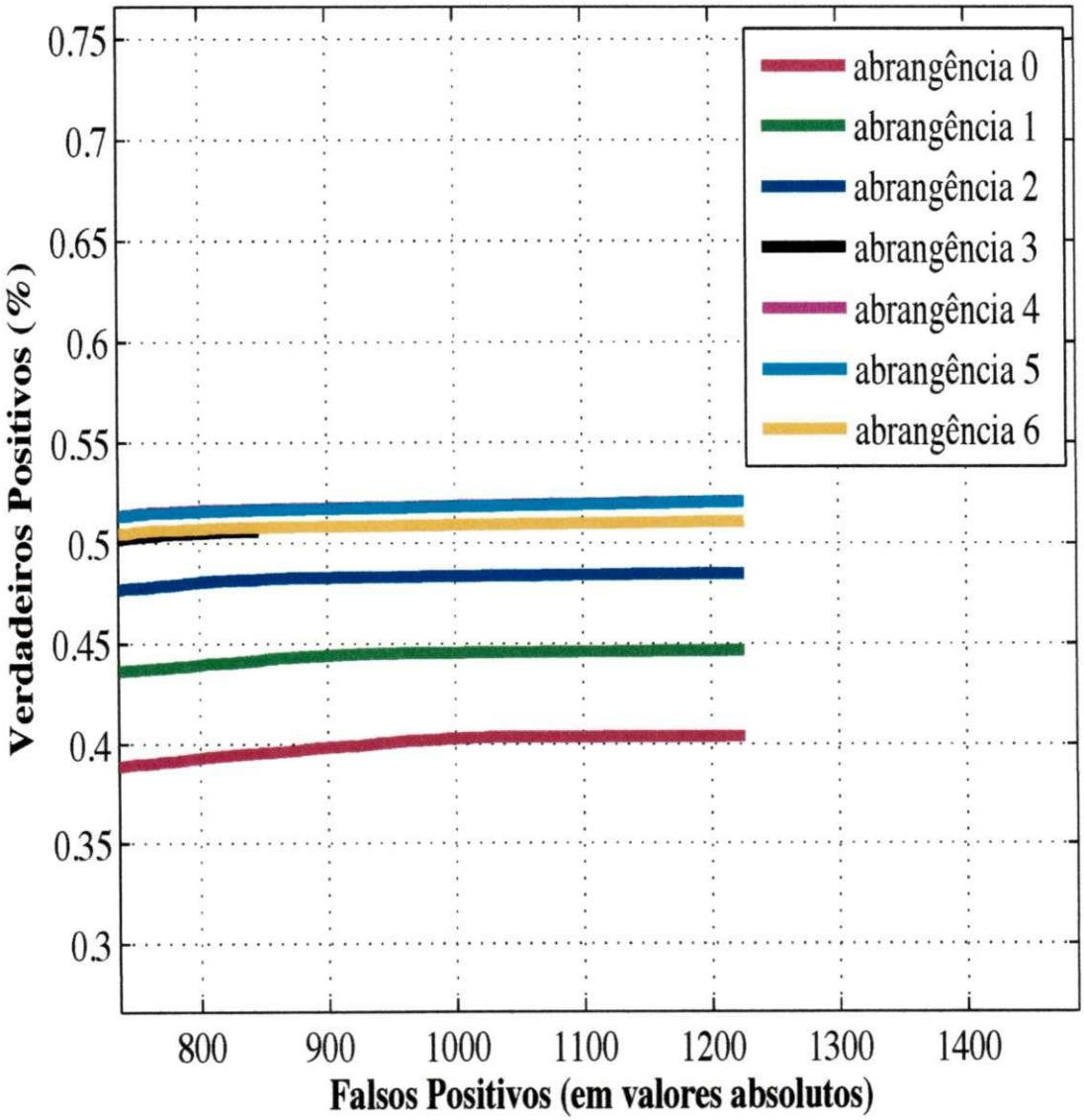


Figura 5.14: Gráfico com *zoom* na região de estabilização das curvas da comparação de várias abrangências de marcação da face detectada na base FDDB com métrica contínua.

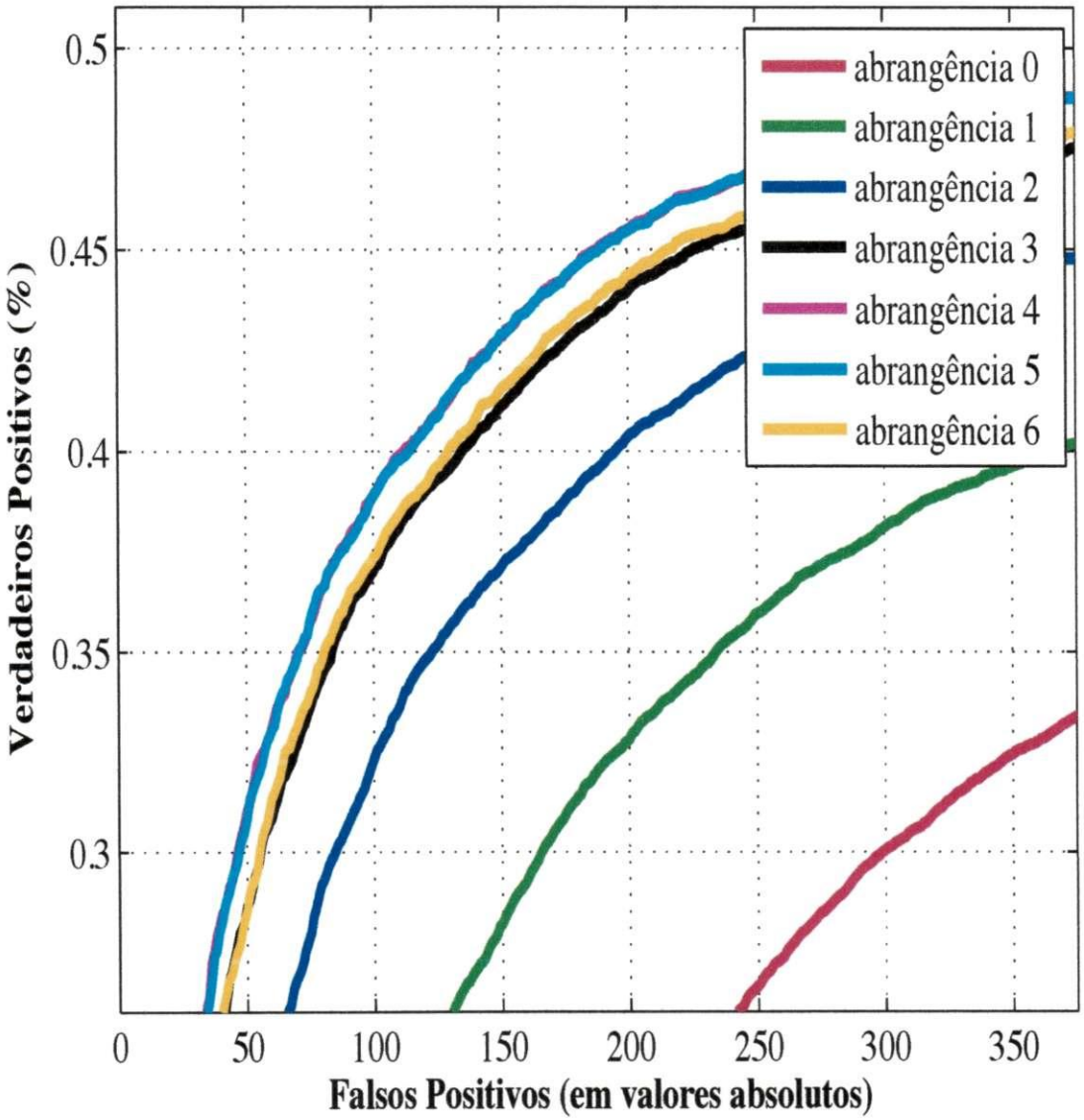


Figura 5.15: Gráfico com *zoom* na região de início das curvas da comparação de várias abrangências de marcação da face detectada na base FDDB com métrica contínua.

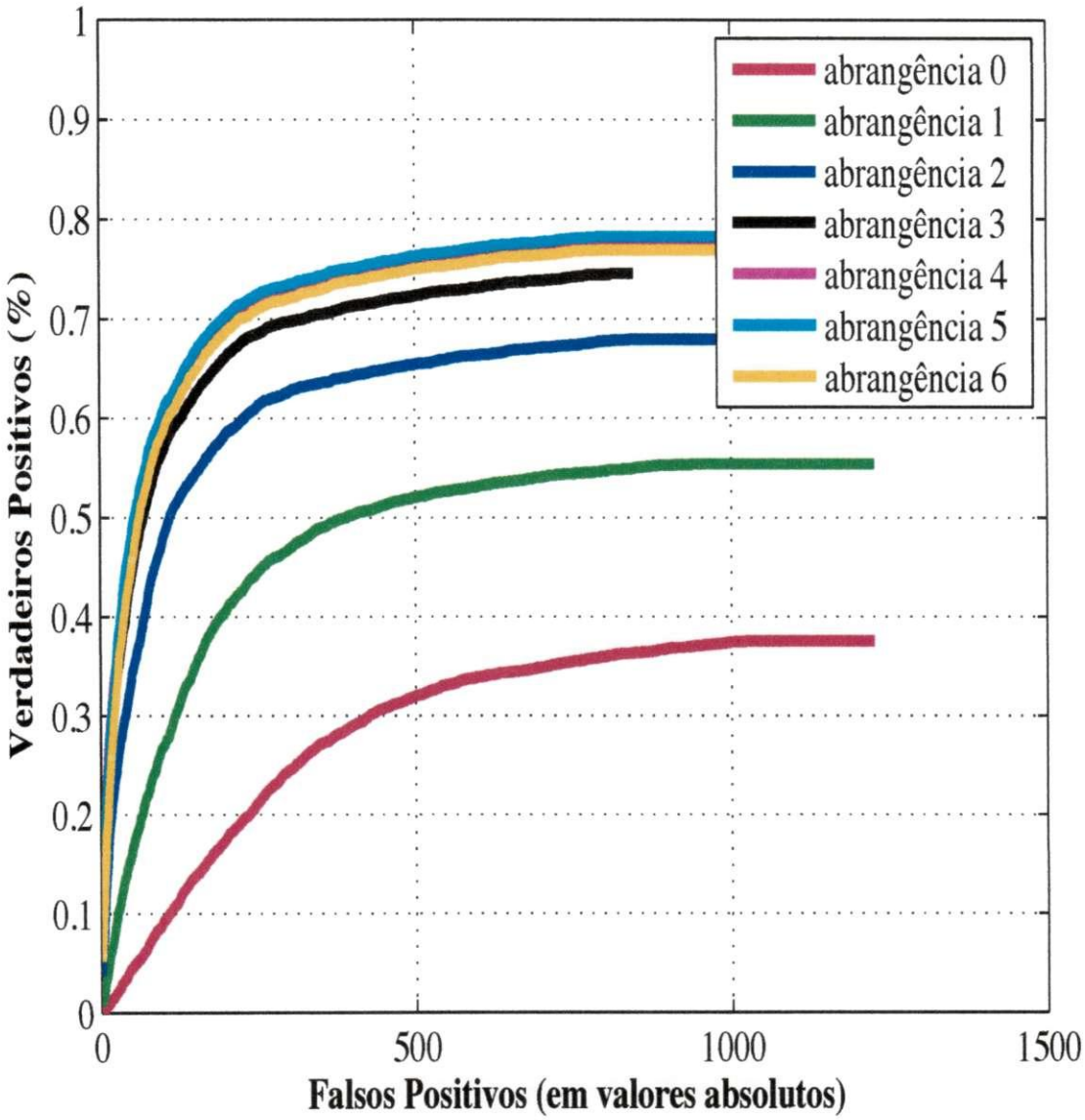


Figura 5.16: Gráfico em visão completa da comparação de várias abrangências de marcação da face detectada na base FDDB com métrica discreta.

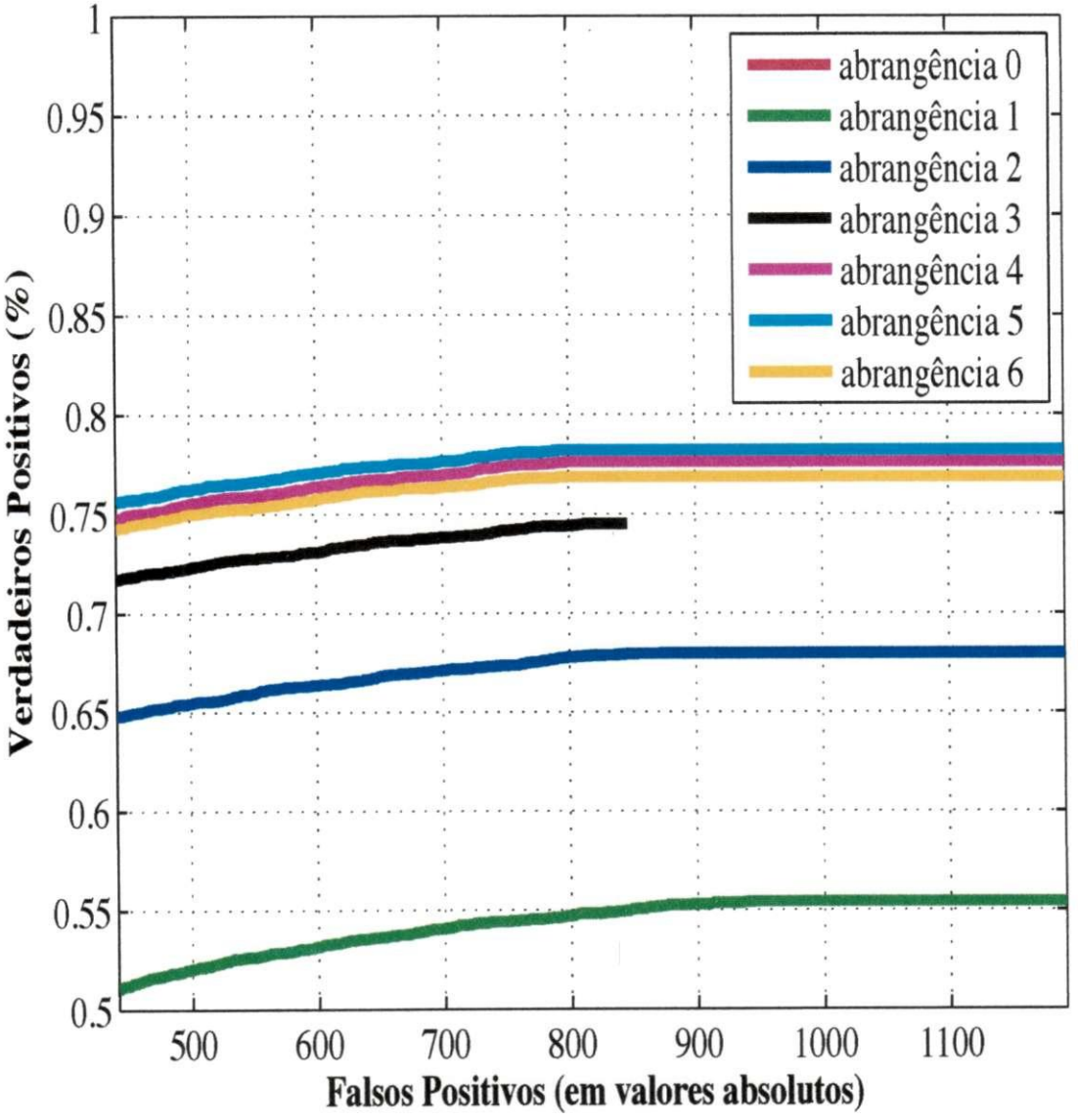


Figura 5.17: Gráfico com *zoom* na região de estabilização das curvas da comparação de várias abrangências de marcação da face detectada na base FDDB com métrica discreta.

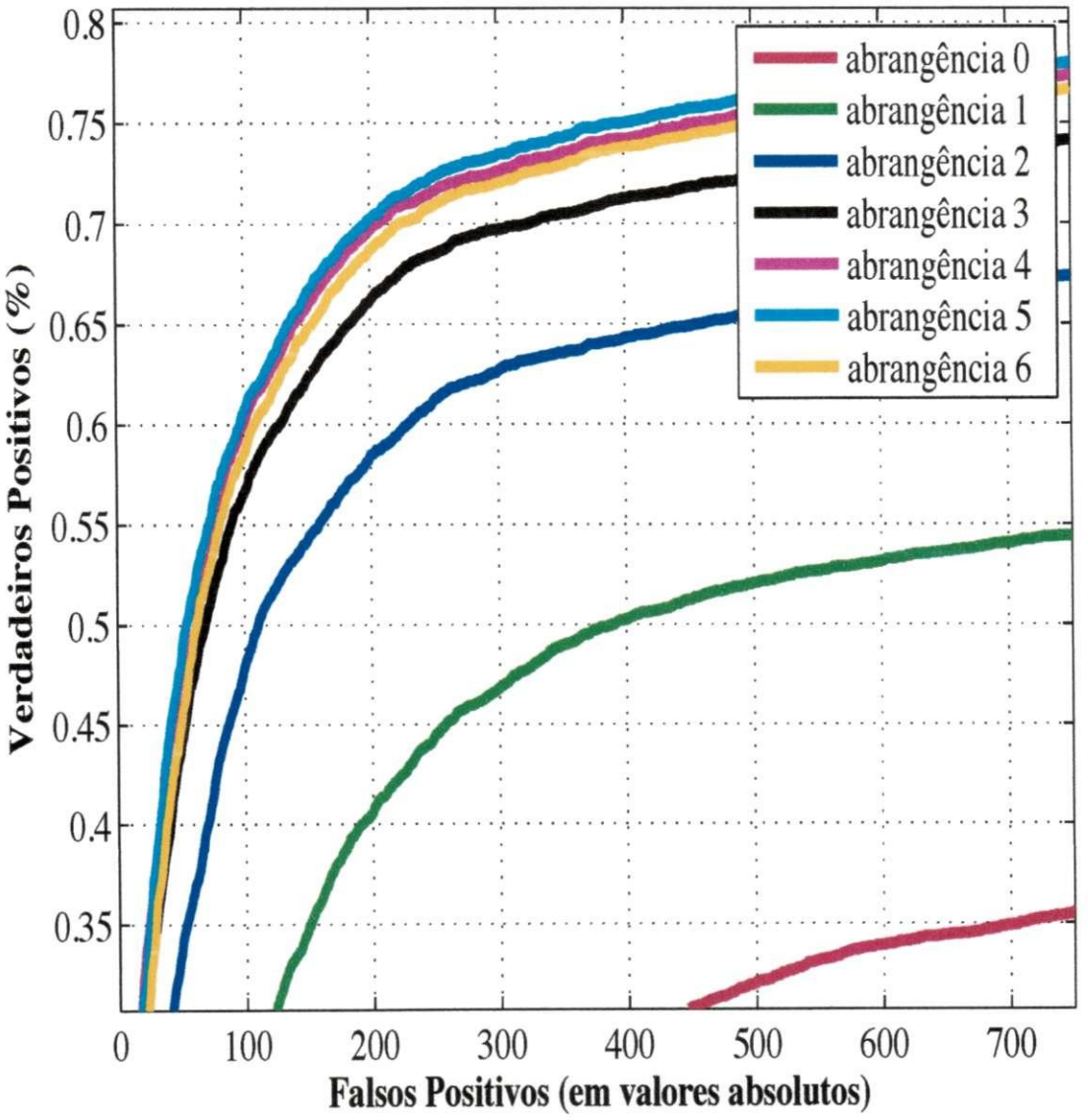


Figura 5.18: Gráfico com *zoom* na região de início das curvas da comparação de várias abrangências de marcação da face detectada na base FDDB com métrica discreta.

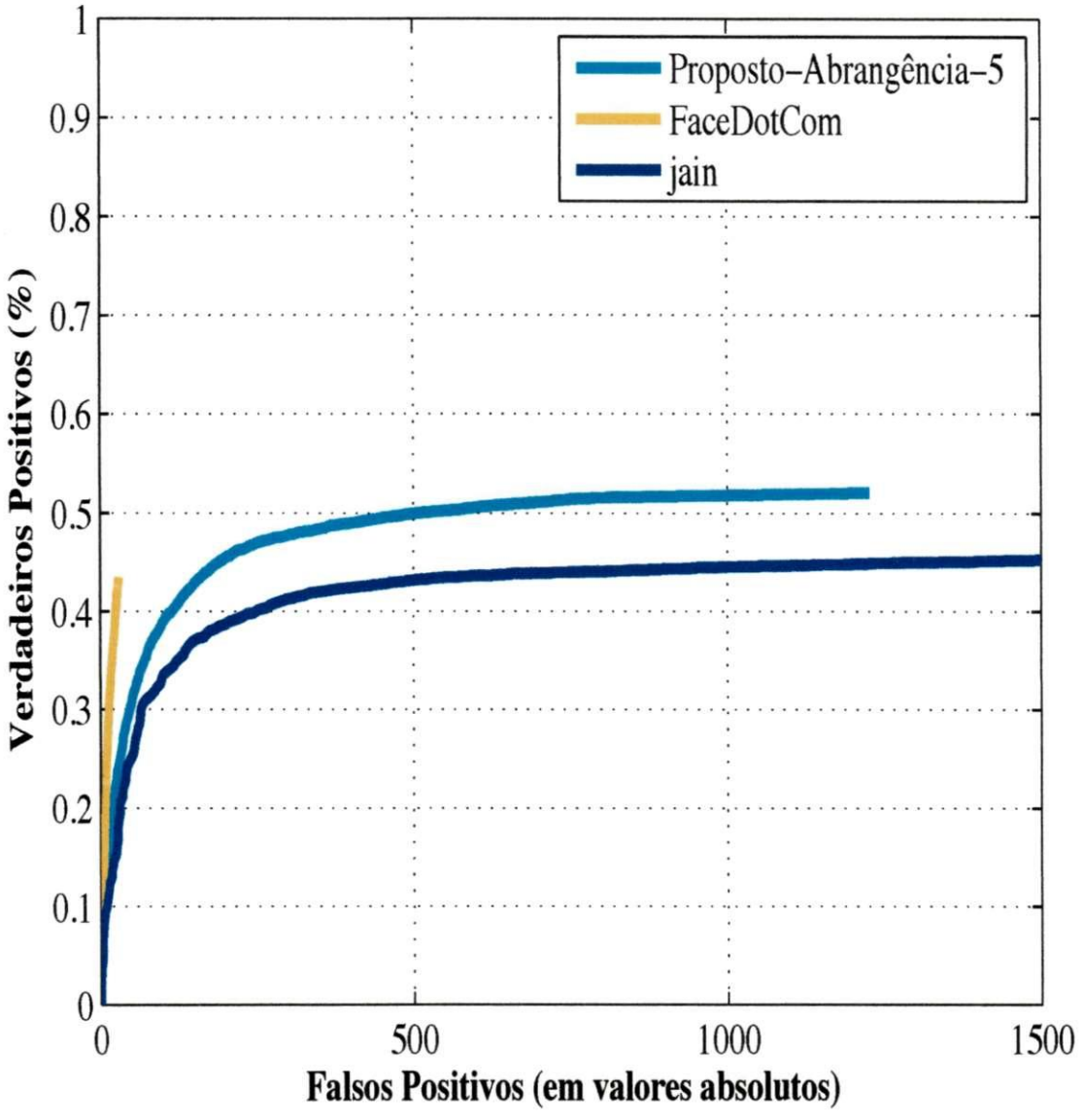


Figura 5.19: Comparação dos resultados da detecção de faces nas imagens da base FDDB, com avaliação em modo contínuo. Os resultados para o detector proposto são marcados com abrangência 5.

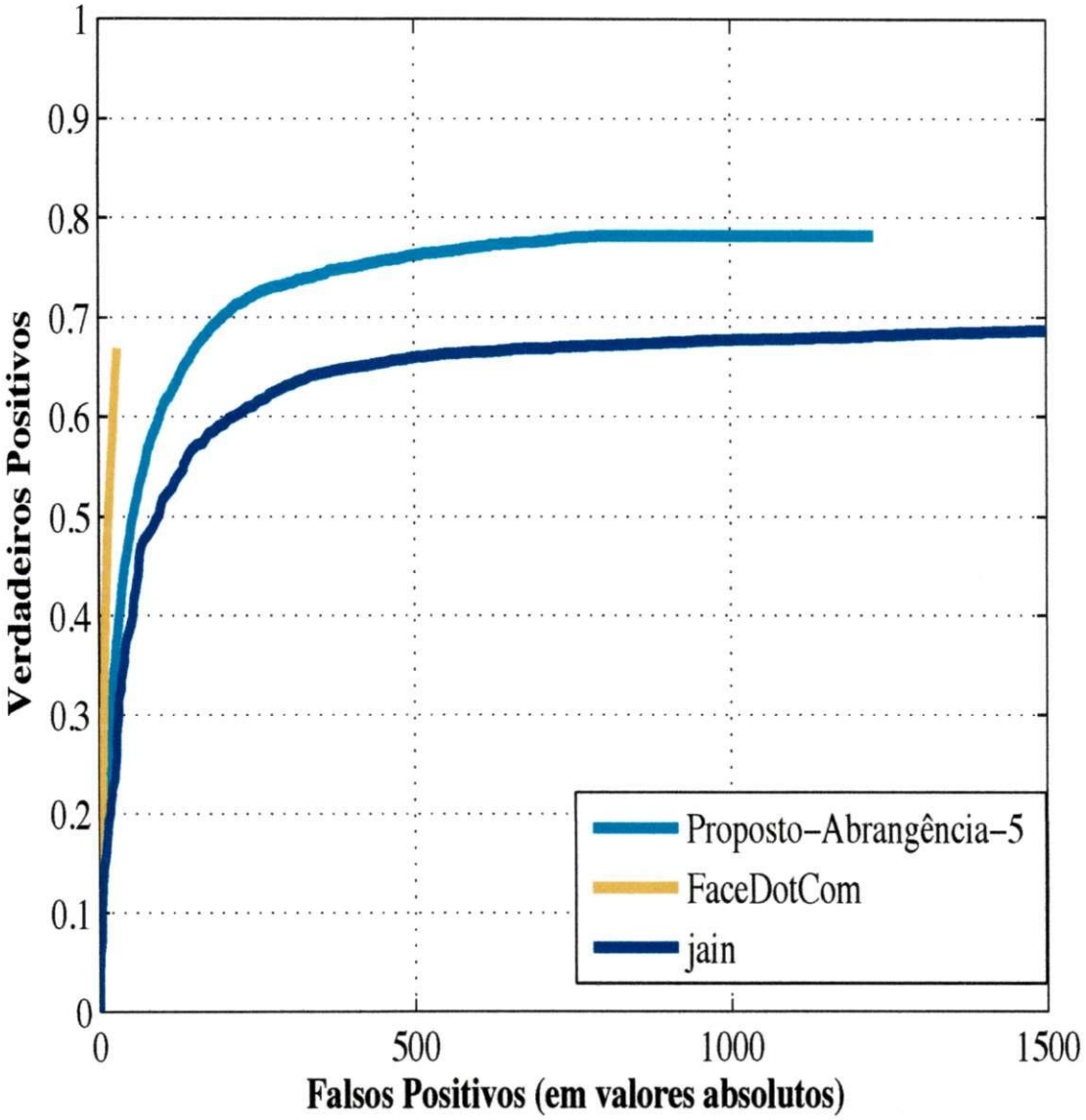


Figura 5.20: Comparação dos resultados da detecção de faces nas imagens da base FDDB, com avaliação em modo discreto. Os resultados para o detector proposto são marcados com abrangência 5.

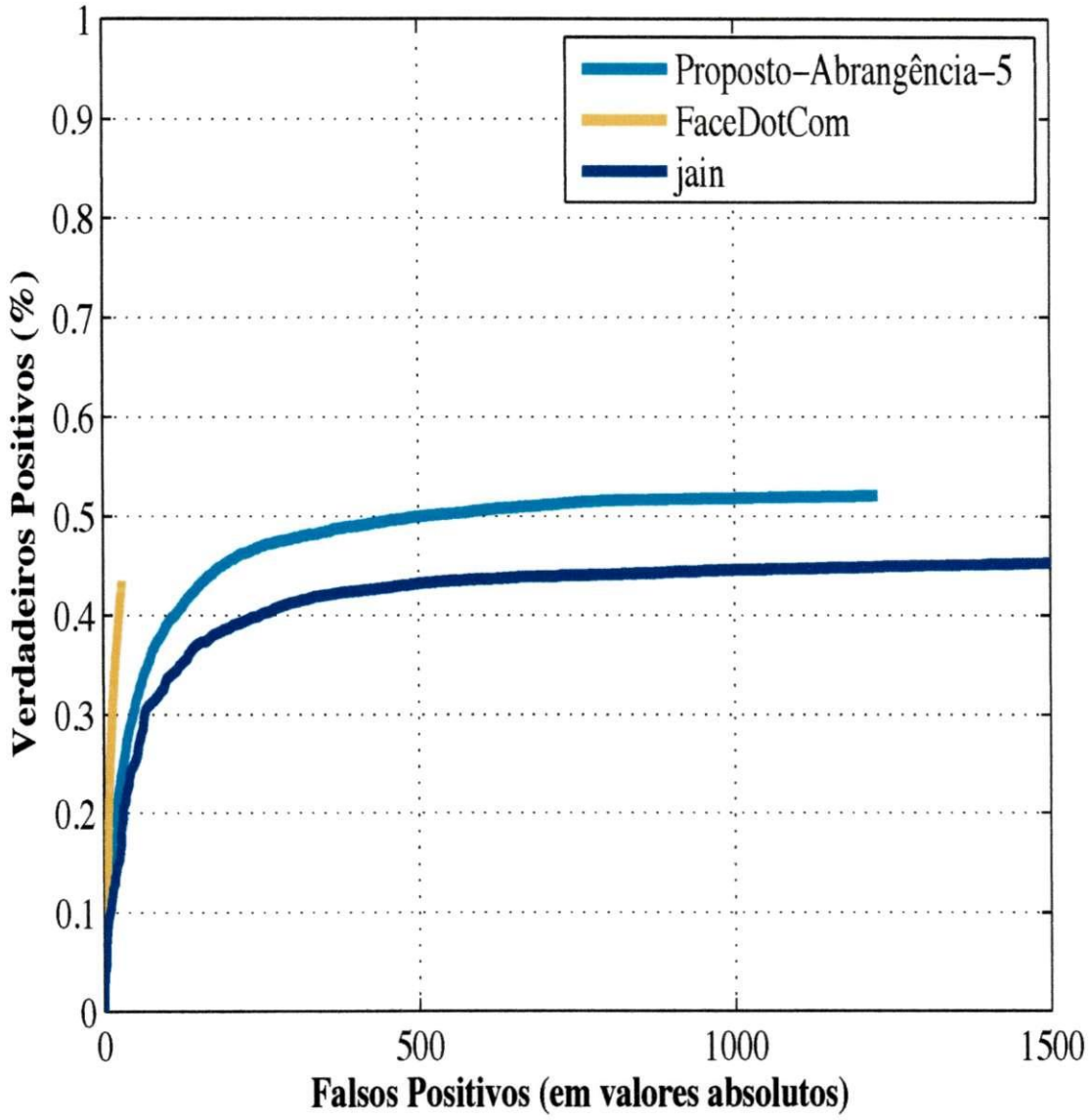


Figura 5.21: Resultados da detecção de faces nas imagens da base FDDB, com avaliação em modo contínuo.

5.3 Avaliação do Detector Invariante à Rotação no Plano

A única base de imagens conhecida e que tem sido amplamente utilizada para a avaliação de detectores de faces invariantes à rotação é a base CMU-MIT (ROWLEY; BALUJA; KANADE, 1998b). A base FDDB possui variações de rotação no plano e fora do plano, mas não possui variações extremas de rotação. Adicionalmente, a CMU-MIT tem sido bastante utilizada para a comparação de resultados de diversas abordagens.

Na Figura 5.22, são apresentados resultados de detecção de faces em imagens da base CMU-MIT, com o conjunto rotacionado, obtidos por quatro detectores: o detector proposto nesta tese, o detector de Jones e Viola (2003), o detector de Rowley, Baluja e Kanade (1998b) e o detector de Huang et al. (2007). As curvas que representam os resultados de Jones e Viola (2003) e de Rowley, Baluja e Kanade (1998b) foram construídas a partir de resultados de tabelas reportadas nos artigos correspondentes.

A curva que representa o resultado de Rowley, Baluja e Kanade (1998b) possui uma forma de *dente-de-serra* porque foi interpolada usando apenas 4 pontos. A métrica de verificação de acerto usada na abordagem proposta é semelhante a mencionada por Lienhart, Kuranov e Pisarevsky (2002), conforme foi descrito na Seção 5.1.

Os resultados obtidos por Huang et al. (2007) são muito bons, com taxas de verdadeiros positivos acima de 90% sem a ocorrência de nenhum falso positivo. Um crítica que pode ser feita à curva referente aos resultados do detector de Huang et al. (2007) está relacionada à sua cobertura: por que os autores não variaram suficientemente os parâmetros de geração da curva, afim de expandi-la mais? Outro fator a ser questionado é que os autores não mencionam a métrica de acerto utilizada, não fazendo nenhuma afirmação sobre a utilização de distância entre os centros da face detectada e da face de *groundtruth*, nem mencionando se levaram em consideração a área das regiões de face.

Em termos de complexidade, os algoritmos que usam árvores de classificadores podem ser comparados quanto à quantidade de estágios. O detector de Jones e Viola (2003) é composto por duas cascatas, uma para estimar a rotação (com 11 estágios) e outra para classificar entre face e não face (35 estágios). Assim sendo, uma janela deve passar por 46 estágios de

avaliação, a fim de poder ser classificada como face.

O detector de Huang et al. (2007) possui 234 nós (cada nó corresponde a um classificador fraco) e 18 estágios. O detector proposto completo possui 192 nós e 6 níveis de altura da árvore de classificação, sendo 64 nós para cada uma das visões: frontal, perfil esquerdo e perfil direito. Assim, uma janela candidata teria de passar por menos nós e por menos estágios de classificação quando submetida ao detector proposto neste tese. Além disso, a faixa de ângulos de rotação que o detector proposto utiliza é mais restrita ($\pm 10°$) em relação à faixa do detector de Huang et al. (2007) ($\pm 15°$), isso permite maior precisão na estimativa do ângulo de rotação.

Os autores (HUANG et al., 2007) não explicam por que não avaliaram seu detector usando o conjunto de imagens da base CMU-MIT sem variações extremas de rotação (*Test Sets A, B* e *C*). Possivelmente, o referido detector teria um resultado inferior, pois foi treinado com faces de resolução 24×24 e a base mencionada possui muitas imagens com resoluções mais baixas.

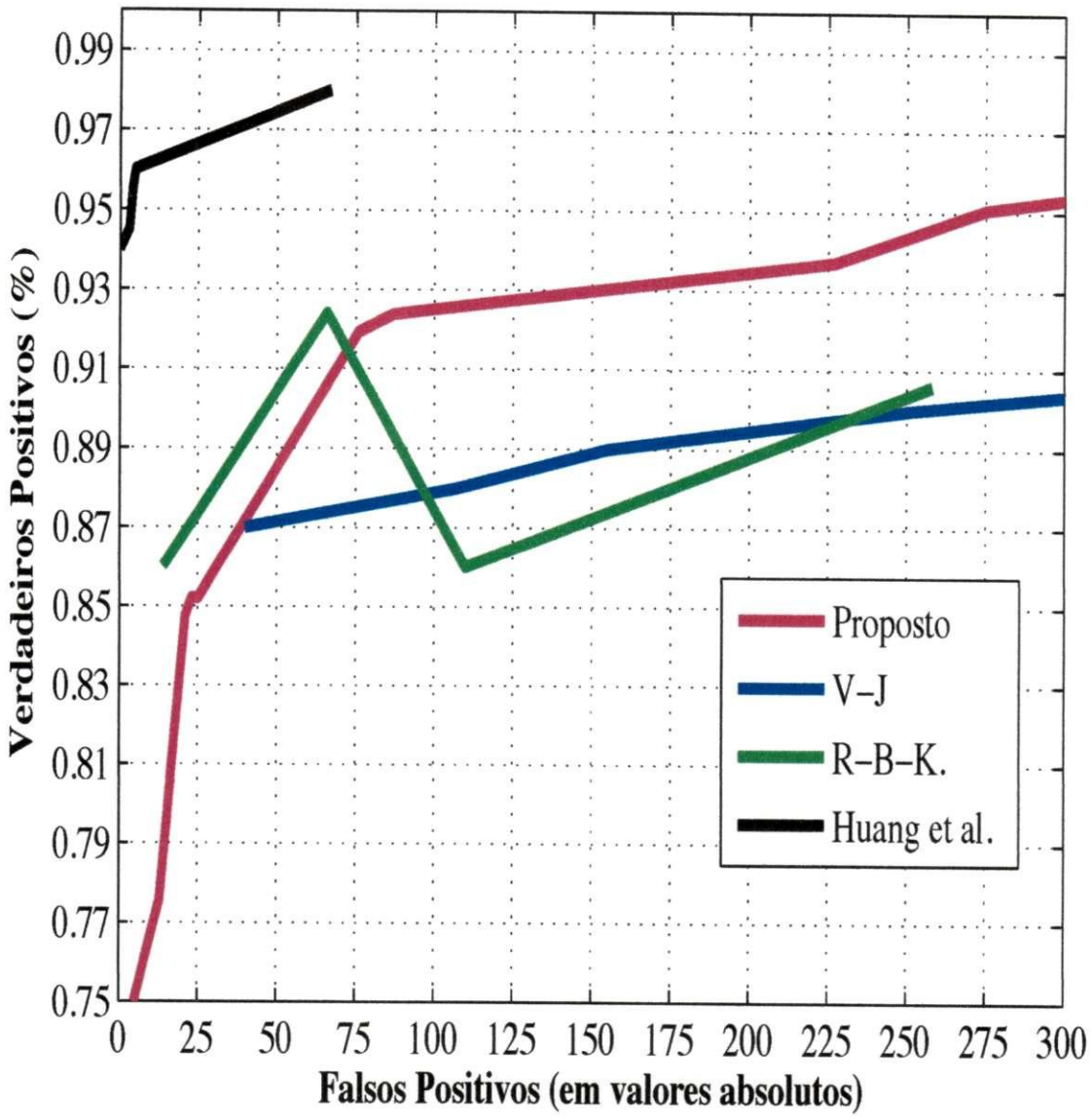


Figura 5.22: Resultados da detecção de faces nas imagens da base CMU-MIT rotacionadas (*rotated set*).

5.4 Considerações Finais

Neste capítulo, foram apresentadas avaliações experimentais do detector de faces proposto nesta tese, bem como comparações com resultados obtidos por outros detectores publicados na literatura especializada ou disponíveis na web. Foram usadas duas bases de imagens distintas para avaliação: a CMU-MIT e a FDDB. Os resultados do detector proposto foram descritos de maneira a evidenciar a evolução da pesquisa e apresentar diretrizes que podem ser seguidas por pesquisadores que estejam desenvolvendo detectores de faces.

As principais diretrizes para a criação de detectores de faces que foram evidenciadas por esta pesquisa, podem ser sumarizadas da seguinte forma:

- É importante que os erros cometidos pelo detector de faces sejam analisados criteriosamente;
- Da análise dos erros, novas imagens podem ser adicionadas à base de treinamento para melhorar sua qualidade;
- As imagens de faces recortadas para o treinamento devem conter regiões externas à face para melhorar a precisão dos classificadores;
- As faces detectadas devem ser marcadas na imagem de modo consistente com a métrica que será usada para avaliação das detecções.

Outro fator importante é a utilização de uma grande quantidade de imagens para o treinamento, visto que a classe negativa (não-faces) possui tamanho praticamente infinito. Para tratar o problema de treinar classificadores com milhões de imagens em tempo factível, foi proposta no Capítulo 4 uma abordagem de paralelização para o treinamento de cascatas de classificadores, cuja avaliação experimental é apresentada no Capítulo 6.

Capítulo 6

Avaliação da Paralelização da Abordagem de Treinamento de Cascatas de Classificadores

Neste capítulo são apresentados os resultados dos experimentos realizados para avaliar o desempenho da abordagem de paralelização proposta nesta tese. Na Seção 6.1, são apresentados os procedimentos experimentais realizados e os tempos de processamento obtidos. Na Seção 6.2 são apresentadas as análises dos tempos de processamento. Há duas medidas bastante utilizadas para avaliar o desempenho de abordagens paralelas, o *speedup* e a escalabilidade, ambas serão explicadas na Seção 6.2.

6.1 Tempos de Processamento

O método de treinamento de cascatas de classificadores é sensível ao tipo de imagem que lhe é fornecido para treinamento. Isso implica no fato de que se dois treinamentos diferem entre si apenas pelas imagens de treinamento, os tempos totais necessários para conclusão podem ser diferentes para cada experimento. Assim, foram realizados dois conjuntos de experimentos com 10 configurações diferentes. Os conjuntos de experimentos foram usados para avaliar o *speedup* e a escalabilidade da paralelização. A diferença entre as configurações de

experimentos é o modo como as imagens de não faces são recortadas para obtenção de amostras de treinamento. Para obter a variação dos conjuntos de experimentos, foram atribuídos sistematicamente valores diferentes a dois parâmetros: o passo de deslocamento e a escala de redimensionamento. A partir dos tempos de processamento de cada uma das configurações foi calculado o tempo médio de processamento e usando tais tempos foram obtidas as médidas de *speedup* e escalabilidade. Na Tabela 6.1, são apresentadas as configurações de passo e escala utilizadas.

Tabela 6.1: Configurações de escala e passo para os experimentos de avaliação de desempenho da paralelização.

Experimento	**Escala**	**Passo**
1		2
2		3
3	1,1	4
4		5
5		6
6		2
7		3
8	1,2	4
9		5
10		6

Para cada uma das configurações apresentadas na Tabela 6.1, foram realizados 50 experimentos para avaliar o *speedup* e 50 para avaliar a escalabilidade sendo 10 experimentos para cada quantidade de computadores usados, que varia de 1 a 5. Devido a grande quantidade de dados resultante dos experimentos, os dados serão resumidos utilizando diagramas de extremos e quartis (*boxplots*). Nas Figuras 6.1 e 6.2 são apresentados os diagramas de extremos e quartis utilizando os dados de todos os experimentos realizados.

Cada um dos gráficos das Figuras 6.1 e 6.2 apresenta um resumo de cinco experimentos utilizando quantidades diferentes de imagens ou computadores. Os experimentos para avaliação de *speedup* foram realizados utilizando quantidades iguais de imagens (1000 amostras de faces e 2000 de não faces) e quantidades de computadores variando de 1 a 5. Nos experi-

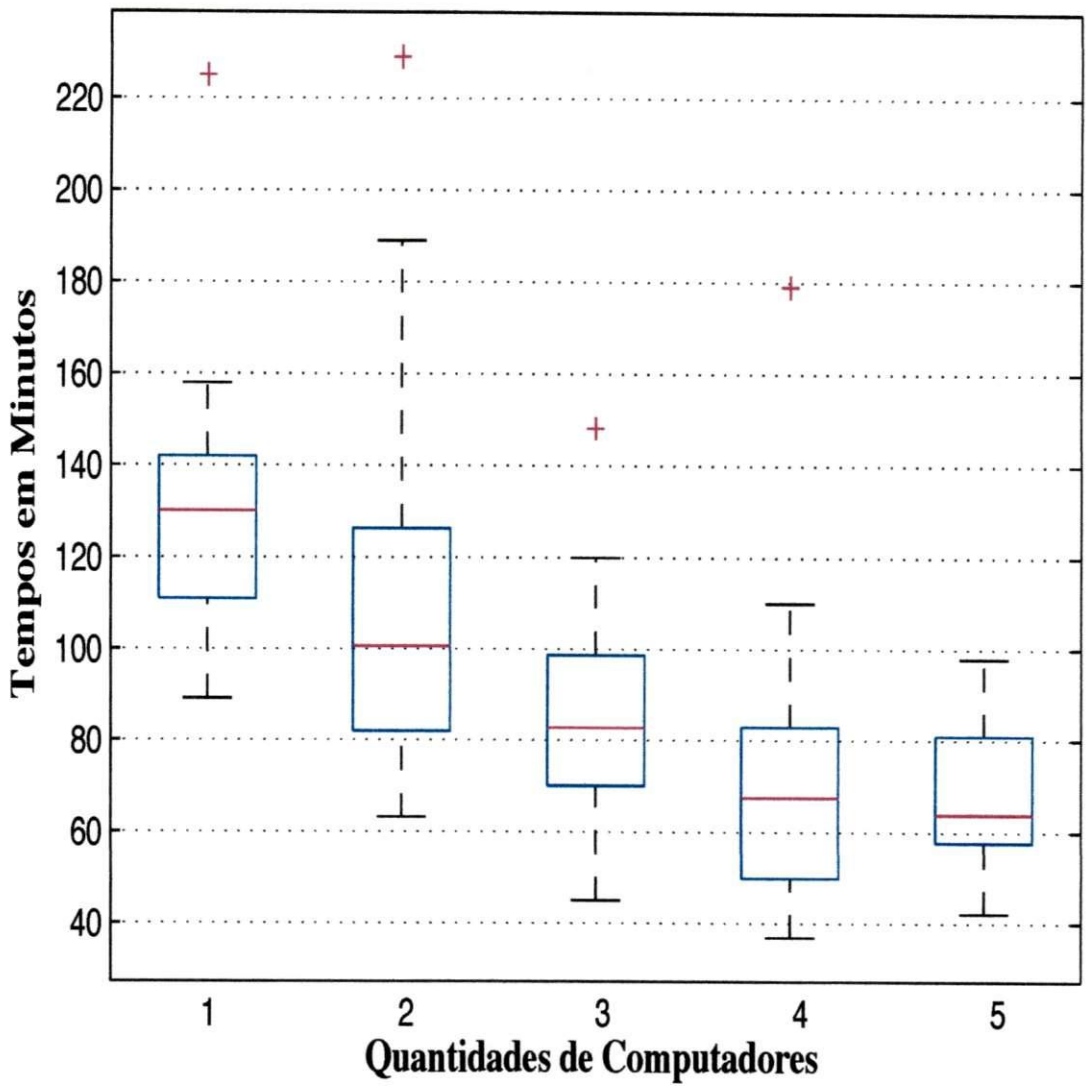


Figura 6.1: Diagrama de extremos e quartis para os experimentos utilizados para avaliar o *speedup*.

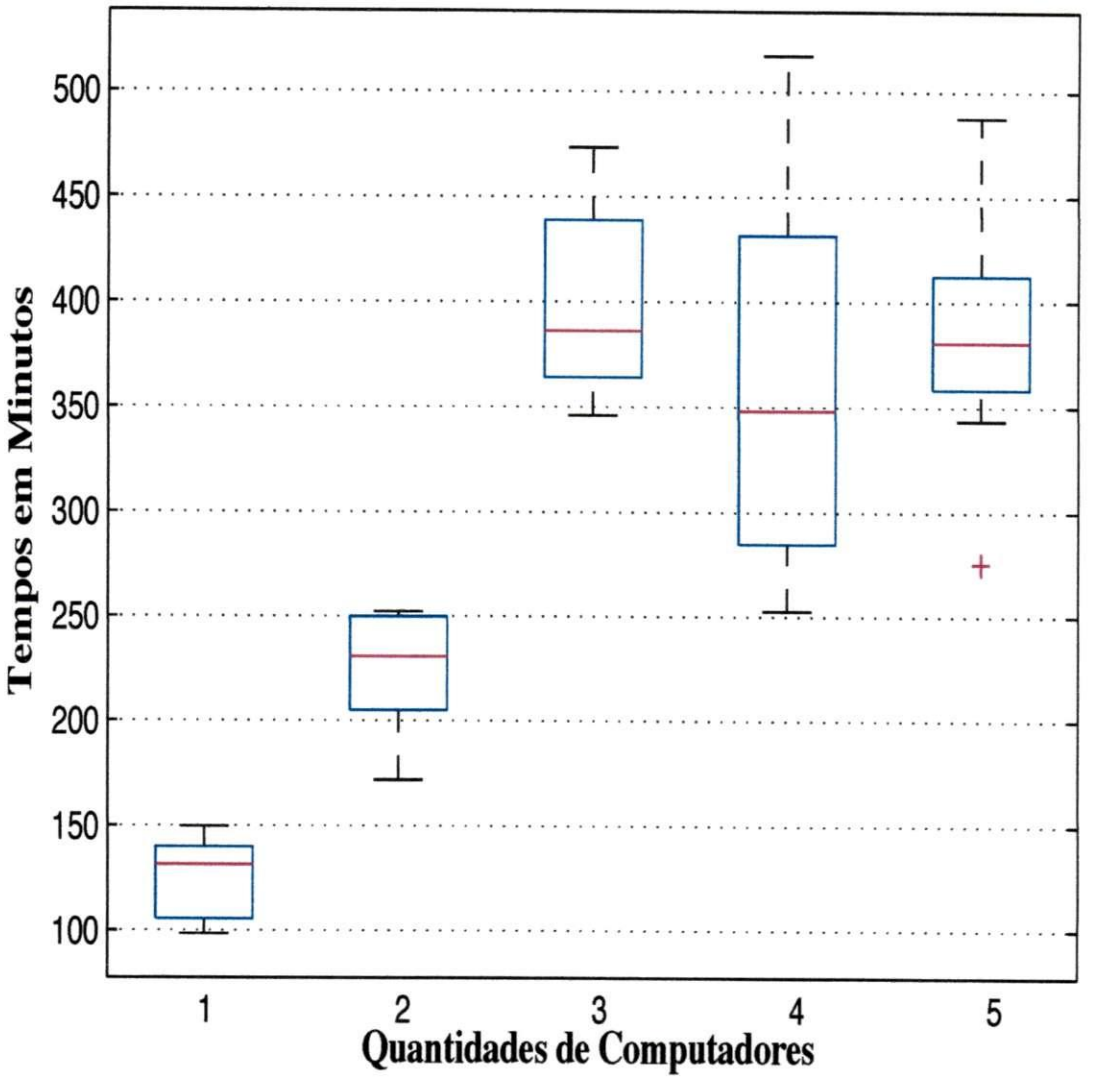


Figura 6.2: Diagrama de extremos e quartis para os experimentos utilizados para avaliar a *escalabilidade*.

mentos para avaliação de escalabilidade, a razão entre quantidades de imagens e quantidades de computadores é mantida constante. Por exemplo, o primeiro experimento usou 1 computador e 3000 imagens (1000 amostras de faces e 2000 de não face) e o experimento 5 usou 15000 imagens (5000 amostras de faces e 10000 amostras de não faces). Nos diagramas de extremos e quartis, as linhas vermelhas centrais das caixas representam as medianas dos valores de tempos e os extremos inferior e superior de cada caixa representam os primeiros e terceiros quartis, respectivamente. As linhas pretas horizontais fora das caixas representam os valores extremos não considerados *outliers*, as cruzes vermelhas representam representam *outliers*.

A partir do gráfico da Figura 6.1, observa-se que a medida que mais computadores são adicionados, os valores das medianas dos tempos de processamento diminuem, isso indica que a velocidade de processamento aumenta. Em relação às distribuições dos tempos de processamento para a escalabilidade, é possível verificar que os tempos começam a se estabelecer em um patamar de estável a partir do uso de três computadores. O fato de que os tempos de processamento usando 2 computadores são inferiores aos tempos usando 3, 4 e 5 computadores será melhor explicado na próxima seção quando a medida de eficiência for explicada.

6.2 Análise dos Resultados

Nesta seção serão apresentadas as avaliações de *speedup* e escalabilidade. Uma das medidas mais usadas para verificar o aumento do desempenho em sistemas de computação paralela é o *speedup* (BUZBEE, 1983; SHI, 1996; HILL; MARTY, 2008). O *speedup* pode ser calculado conforme a Equação 6.1.

$$S(p) = \frac{T_s}{T_p(p)} \tag{6.1}$$

Na Equação 6.1, T_s indica o tempo de execução do algoritmo sem paralelização (geralmente é utilizado o tempo de processamento com apenas um computador como tempo do algoritmo serial) e $T_p(p)$ indica o tempo de processamento do algoritmo paralelo com p com-

putadores. A partir da avaliação dessa equação, observa-se que o *speedup* ideal corresponde à quantidade de computadores utilizados. Por exemplo, se forem necessários 100 segundos para executar determinado algoritmo com 1 computador, idealmente deveriam ser necessários 50 segundos de processamento se 2 dois computadores fossem usados. Desta forma, o *speedup* seria 2 (100/50).

Tabela 6.2: Tempos médios, desvios padrão e valores de *speedup* para as diferentes quantidades de computadores utilizadas.

Computadores	**Tempo Médio (Min.)**	**Desvio Padrão**	***Speedup***
1	128,67	28,75	1
2	109,39	38,43	1,18
3	84	25,17	1,53
4	71,56	31,98	1,80
5	67,01	14,77	1,92

A partir dos valores apresentados na Tabela 6.2, observa-se o aumento da velocidade de processamento à medida que mais computadores são utilizados no processamento. Segundo a Lei de Amdahl, se a porcentagem de um programa que pode ser paralelizada for P, então o *speedup* máximo que pode ser obtido executando a versão paralelizada do programa em N computadores será dada pela Equação 6.2.

$$SU = \frac{1}{(1-P) + \frac{P}{N}} \tag{6.2}$$

O valor de P pode ser estimado utilizando-se a Equação 6.3 (SHI, 1996), na qual SU indica o *speedup* medido para N computadores.

$$P_{estimado} = \frac{\frac{1}{SU} - 1}{\frac{1}{N} - 1} \tag{6.3}$$

Então, se o valor de P for estimado utilizando o *speedup* total obtido para cinco computadores, e apresentado na Tabela 6.2, o resultado é 0,60. A partir desse valor de P, os valores estimados para o *speedup* máximo que pode ser obtido pelo sistema aqui descrito usando diferentes quantidades de computadores são apresentados na Figura 6.3.

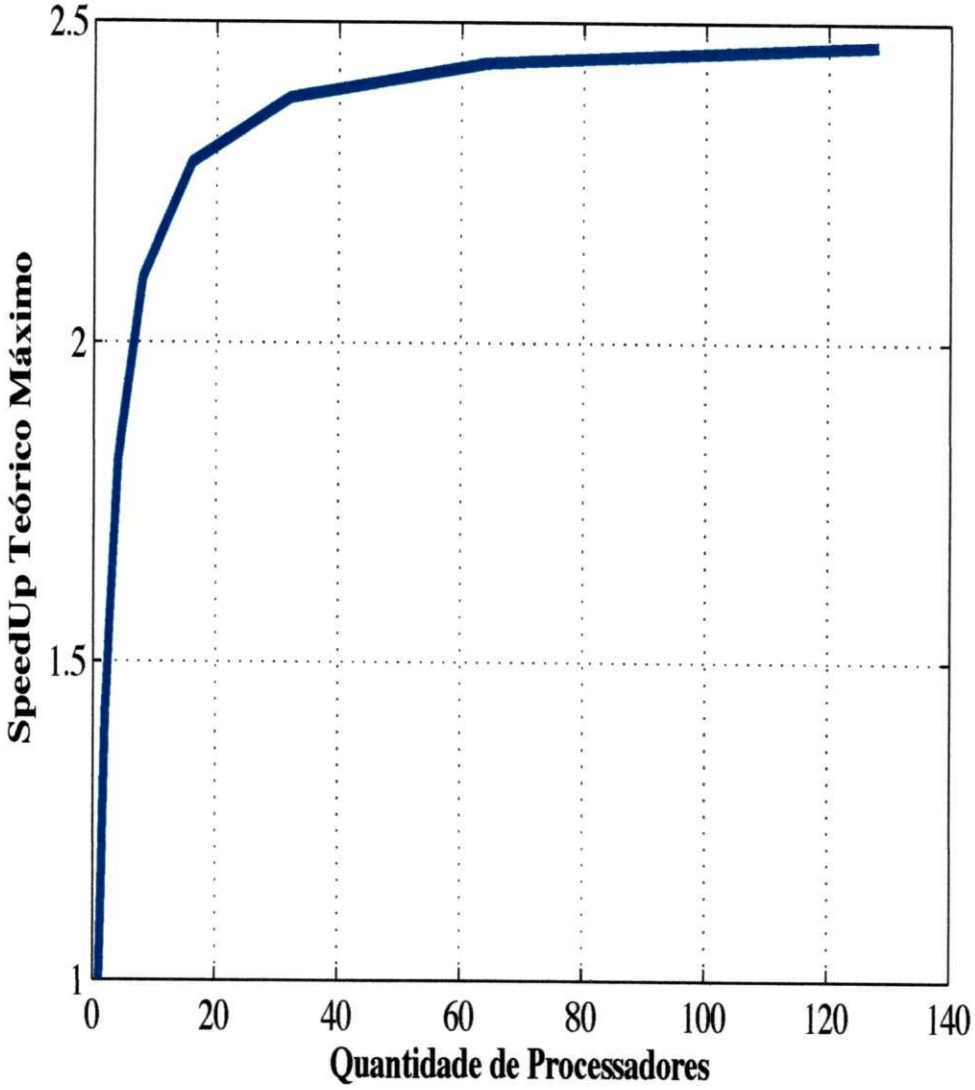


Figura 6.3: Estimativas de *speedup* máximo.

Segundo Grama, Gupta e Kumar (1993), o *speedup* não aumenta linearmente com o aumento do número de computadores, ao contrário, o *speedup* tende a saturar e a eficiência cai à medida que mais computadores são adicionados para processamento. A eficiência do processamento paralelo é dada pela razão entre o *speedup* e a quantidade de computadores utilizada para obter tal *speedup*, como é apresentado na Equação 6.4.

$$E = \frac{Speedup}{Computadores} \tag{6.4}$$

Na Tabela 6.3, são apresentados os valores dos tempos médios de processamento, desvios padrão dos tempos de processamento, quantidades de computadores e escalabilidade dos experimentos. Os valores de escalabilidade são calculados utilizando a mesma equação que é utilizada para calcular o *speedup*, o que difere para escalabilidade é que as quantidades de imagens são diferentes para cada quantidade de computador utilizada.

Tabela 6.3: Tempos médios, desvios padrão e valores de escalabilidade para as diferentes quantidades de computadores utilizadas.

Computadores	**Tempo Médio (Minutos)**	**Desvio Padrão**	**Escalabilidade**
1	128,31	18,63	1
2	237,35	37,32	0,54
3	402,12	49,07	0,32
4	388,82	107,30	0,33
5	398,21	87,80	0,32

A afirmação feita por Grama, Gupta e Kumar (1993) de que a eficiência decai à medida que mais computadores são utilizados para processamento pode ser comprovada a partir dos resultados de eficiência apresentados na Tabela 6.4. Um das implicações mais importantes apresentadas por Grama, Gupta e Kumar (1993) é que para alguns algoritmos paralelos é possível aumentar a carga de trabalho em uma proporção diferente do aumento da quantidade de processadores de modo que a eficiência permaneça em um determinado patamar. Portanto, para que os valores de eficiência apresentados na Tabela 6.4 para as quantidades de computadores 3, 4 e 5 não decaissem, seria necessário aumentar ainda mais a quantidade de imagens processadas.

Tabela 6.4: Valores da eficiência calculada para os experimentos de escalabilidade.

Computadores	**Escalabilidade**	***Eficiência***
1	1	1
2	0,54	0,27
3	0,32	0,11
4	0,33	0,08
5	0,32	0,06

6.3 Considerações Finais

Neste capítulo, foram apresentados os resultados experimentais da avaliação da abordagem de paralelização proposta. Esses resultados foram avaliados por meio das métricas conhecidas como *speedup*, escalabilidade e eficiência.A possibilidade de obter valores mais altos de eficiência deverá ser estudado em trabalhos futuros de forma a modelar o sistema para verificar com mais exatidão em quais situações é possível obter *speedup* superlinear utilizando a abordagem proposta.

Capítulo 7

Conclusão e Trabalhos Futuros

Neste capítulo, será apresentado um breve sumário do que foi exposto nos demais capítulos deste documento, assim como serão formuladas conclusões sobre a abordagem proposta e sugeridas proposições para trabalhos futuros.

7.1 Sumário da Pesquisa Realizada

Nesta tese, é apresentada uma nova abordagem para a detecção invariante à rotação de faces humanas em imagens digitais e um método para a paralelização de treinamentos de cascatas de classificadores combinados por meio de *boosting*. A abordagem de detecção de faces possui como principais características a exploração da invariância por treinamento nos nós da árvore de classificadores e a exploração do compartilhamento de características entre imagens de faces em diferentes poses.

Para verificar a qualidade dos resultados obtidos pelo detector de faces, ele foi testado com duas bases de imagens, a saber: CMU Test Sets (ROWLEY; BALUJA; KANADE, 1998b) e FDDB (JAIN; LEARNED-MILLER, 2010). Os resultados de detecção foram comparados com os resultados de outras abordagens, tais como: Rowley, Baluja e Kanade (1998b), Jones e Viola (2003), Viola e Jones (2004), Huang et al. (2007) e *faceDotCom*[1]. Além disso, foi realizada uma extensiva revisão bibliográfica na área de detecção de faces.

[1] http://www.face.com

Para tornar factível a construção de uma árvore de classificadores treinados com milhões de imagens de não faces, foi proposta uma abordagem de paralelização do método de treinamento de cascatas de classificadores proposto por Viola e Jones (2004). A abordagem paralelizada foi avaliada por meio do cálculo do *speedup* e da escalabilidade.

7.2 Principais Contribuições

Os resultados de detecção de faces foram compatíveis com o estado da arte, em alguns casos superiores. Por exemplo, pode ser visto na Figura 5.11 que o detector proposto obteve taxas de verdadeiros positivos superiores às taxas obtidas pelos detectores de Viola e Jones (2004), de Rowley, Baluja e Kanade (1998b) e *faceDotCom*, quando a quantidade de falsos positivos é menor do que 30. Apenas uma das abordagens concorrentes obteve resultados superiores, a de Huang et al. (2007), por uma diferença em torno de 8%. Porém, o gráfico dos resultados de Huang et al. (2007) apresenta problemas quanto à cobertura, aparecendo truncado até uma pequena quantidade de falsos positivos e os autores não mencionam as métricas usadas para avaliar seu detector. Adicionalmente, a abordagem proposta possui menor complexidade computacional em termos de quantidade de níveis da árvore de classificadores e quantidade de nós de processamento.

Em relação aos resultados da abordagem mais semelhante à proposta, que é a de Jones e Viola (2003), os resultados foram excelentes; em alguns pontos da curva ROC, foram maiores que 5%, especificamente, na faixa de 125 a 300 falsos positivos (ver Figura 5.22). Em determinado ponto da curva ROC, os resultados de Rowley, Baluja e Kanade (1998b) foram 7% inferiores, na faixa próxima a 120 falsos positivos. Esses resultados podem ser visualizados no gráfico da Figura 5.22.

A paralelização do código de treinamento viabilizou todos os resultados obtidos para invariância à rotação. Os treinamentos utilizando a implementação original do OpenCV levavam no mínimo 15 dias para serem concluídos, sendo executada em um único computador. Com a abordagem paralelizada, um treinamento completo de 20 estágios, que processa bilhões de imagens no último estágio, leva menos de 24 horas para ser concluído em um *cluster* de 5 computadores.

Uma diretriz utilizada no aperfeiçoamento de classificadores é que a análise criteriosa das imagens de faces que não são detectadas ajuda a identificar os padrões de erros e permite que imagens que possuem tal padrão sejam incorporadas ao treinamento aumentando, consequentemente, as taxas de verdadeiros positivos e reduzindo as de falsos positivos (ver Seção 5.1). Essa diretriz foi aplicada nos treinamentos realizados nesta tese e mostrou-se bastante eficaz. Vale salientar que o método conhecido como *bootstrapping*, que é realizado automaticamente na abordagem de Viola e Jones (2001), de certa forma, realiza o aperfeiçoamento de classificadores. Porém, a análise que foi realizada nesta tese foi um exame visual das imagens de teste. Essa análise permite que padrões com maior abstração, tais como uso de óculos de sol, possam ser identificados por humanos e imagens desse tipo sejam adicionadas ao conjunto de treinamento.

Outro fator que foi comprovado experimentalmente nesta tese é que a divisão da imagem de face em regiões e a extração de características de cada região dificulta ou impossibilita a obtenção de invariância à rotação (como pode ser visto nos resultados experimentais apresentados no Apêndice A). Uma exceção seria o uso de um classificador capaz de lidar com esse tipo de divisão da imagem, por exemplo alguns classificadores que utilizam kernel RBF (*Radial Basis Function* - Função de Base Radial).

Uma das contribuições desta tese é a investigação da influência sobre os resultados de detecção do modo como as imagens de treinamento são recortadas e como as regiões detectadas são marcadas na imagem. Os resultados experimentais comprovaram que treinamentos realizados com imagens de faces recortadas de modo mais restrito, que deixam de fora partes da face como o queixo e linha do cabelo, obtêm resultados inferiores aos treinamentos realizados com recortes mais amplos. Uma comprovação ainda mais importante obtida por esta tese é o fato de que algumas métricas de avaliação de detectores de faces são sensíveis ao modo como as imagens são marcadas após a detecção. Na Seção 5.2, são apresentados resultados experimentais que comprovam o fato de que as métricas utilizadas pelo protocolo da FDDB são sensíveis a área de abrangência da região marcada. Isso implica no fato de que é possível obter resultados na FDDB cerca de 30% superiores sem realizar novos treinamentos, apenas aumentando a área da região detectada.

Devido ao fato de que a classe negativa (não faces) possui tamanho praticamente infi-

nito, para treinar um classificador para a detecção de faces é necessário usar uma grande quantidade de imagens de não faces. No caso de uma árvore de classificadores como a que foi proposta nesta tese, foram empregadas bilhões de imagens de não faces. Um treinamento desse tipo só é factível utilizando computação de alto desempenho. Nesta tese, foi proposta uma abordagem de paralelização do método de treinamento de cascatas de classificadores proposto por Viola e Jones (2001) que possui os seguintes diferenciais em relação a outras propostas: é uma abordagem híbrida (utiliza passagem de mensagens entre computadores e *threads* para o processamento *multi-core*) e também paraleliza a etapa de recorte das imagens. As abordagens de paralelização, discutidas na revisão bibliográfica, propuseram abordagens de paralelização apenas para o método *AdaBoost*. A abordagem proposta nesta tese paraleliza toda a abordagem de treinamento de cascatas de classificadores de Viola e Jones (2001).

7.3 Trabalhos Futuros

As duas principais extensões desta pesquisa são: realizar experimentos com a árvore *multiview* completa com os ramos de perfil treinados e tornar possível a detecção de faces invariante à rotação em vídeo de tempo real. Para realizar a primeira tarefa, será necessário despender algum tempo com a realização do treinamento e a análise criteriosa dos resultados da detecção, a fim de investigar pontos a serem melhorados, como foi realizado para as cascatas de faces frontais (vide Seção 5.1).

Para a segunda proposição de trabalho futuro, será necessária uma pesquisa na área de rastreamento de objetos em vídeo. Além disso, será necessário um esforço de implementação para tornar o código de detecção *multithreaded*, habilitando-o a explorar adequadamente o potencial dos processadores modernos que possuem vários núcleos.

Além disso, alguns experimentos devem ser realizados para modelar a abordagem paralela de modo a verificar exatamente em quais situações é possível obter *speedup* superlinear e até que ponto essa característica do sistema pode ser explorada para aumentar a velocidade do processamento paralelo.

Por fim, outras características poderiam ser avaliadas na abordagem de árvore de classificadores, como, por exemplo, o método LBP. Já há uma versão de treinamento de cascata de LBP implementada no OpenCV, não seria muito complicado testar esse tipo de característica, o que mostra a flexibilidade da abordagem proposta, no tocante à possibilidade de treinamento usando vários tipos diferentes de características.

7.4 Trabalhos Publicados e em Fase de Redação

Nesta seção, são apresentados os artigos publicados contendo resultados da pesquisa desenvolvida nesta tese e os artigos que serão submetidos. Pretende-se publicar mais dois artigos. Um deles tratará da abordagem de paralelização proposta para o método de treinamento de cascatas de classificadores e o outro apresentará a abordagem de proposta para detecção de faces em imagens digitais com invariância à rotação no plano.

Os seguintes artigos foram publicados, contendo resultados da pesquisa desenvolvida nesta tese:

- PEREIRA, E. T.; GOMES, H. M.; MOURA, E. S.; CARVALHO, J. M.; ZHANG, T. Investigation of Local and Global Features for Face Detection. In: IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing, 2011, Paris. CIMSIVP 2011 Proceedings, 2011;
- PEREIRA, E. T.; GOMES, H. M.; CARVALHO, J. M. Integral Local Binary Patterns: a Novel Approach Suitable for Texture-Based Object Detection Tasks. In: 23rd SIBGRAPI - Conference on Graphics, Patterns and Imagens, 2010, Gramado, RS. Proceedings of SIBGRAPI 2010, 2010. v. 1. p. 201-208.

Os seguintes artigos estão sendo produzidos sobre as abordagens de paralelização e de detecção de faces invariante à rotação no plano, respectivamente:

- A Hybrid Parallel Approach of Cascade Classifier Training for Face Detection. PEREIRA, E. T.; GOMES, H. M.; CARVALHO, J. M. Periódicos alvo: Cluster Computing - The Journal of Networks, Software Tools and Applications;

- Invariance by Training of a Haar-like Boosted Cascade Detector: A Case Study on Face Detection. PEREIRA, E. T.; GOMES, H. M.; CARVALHO, J. M. Periódicos alvo: Pattern Recognition, Pattern Recognition Letters ou Journal of the Brazilian Computer Society.

Bibliografia

ALT, H.; BEHRENDS, B.; BLöMER, J. Approximate matching of polygonal shapes. In: *Proceeding of the seventh annual symposium on Computational Geometry*. [S.l.]: ACM, 1991. p. 186–193.

ANILA, S.; DEVARAJAN, N. Simple and fast face detection system based on edges. *International Journal of Universal Computer Science*, v. 1, p. 54–58, 2010.

BAILLY-BAILLIÉRE, E.; BENGIO, S.; BIMBOT, F.; HAMOUZ, M.; KITTLER, J.; MARIéTHOZ, J.; MATAS, J.; MESSER, K.; POPOVICI, V.; PORÉE, F.; RUIZ, B.; THIRAN, J.-P. The banca database and evaluation protocol. In: *Proceedings of the 4th International Conference on Audio and Video-Based Biometric Person Authentication*. [S.l.: s.n.], 2003. p. 625–638.

BALAS, B. J.; SINHA, P. *Dissociated Dipoles: Image Representation via Non-Local Comparisons*. [S.l.], 2003. Relatório Técnico AIM-2003-018.

BERG, T. L.; BERG, A. C.; EDWARDS, J.; FORSYTH, D. A. Who's in the picture. In: *Neural Information Processing Systems*. [S.l.: s.n.], 2004. p. 137–144.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006.

BLUMER, A.; EHRENFEUCHT, A.; HAUSSLER, D.; WARMUTH, M. Occam's razor. *Information Processing Letters*, v. 24, n. 6, p. 377–380, 1987.

BRADSKY, G.; KAEHLER, A. *Learning OpenCV: Computer Vision with the OpenCV Library*. [S.l.]: O'Really, 2008.

BRUBAKER, S. C.; WU, J.; SUN, J.; MULLIN, M. D.; REHG, J. M. *On the Design of Cascades of Boosted Ensembles for Face Detection.* [S.l.], 2005. Relatório Técnico GIT-GVU-05-28.

BUZBEE, B. L. The efficiency of parallel processing. *Frontiers of Supercomputing*, p. 71–75, 1983.

CHEN, H.-Y.; HUANG, C.-L.; FU, C.-M. Hybrid-boost learning for multi-pose face detection and facial expression recognition. In: *IEEE International Conference on Multimedia and Expo*. [S.l.: s.n.], 2007. p. 671–674.

CRISTIANINI, N.; SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines.* [S.l.]: Cambridge University Press, 2000.

CROW, F. Summed-area tables for texture mapping. In: *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. [S.l.]: 207-212, 1984.

DEB, K.; AGRAWAL, S. Understanding interactions among genetic algorithm parameters. In: *Foundations of Genetic Algorithms*. [S.l.]: Morgan Kaufmann, 1998. p. 265–286.

DEGTYAREV, N.; SEREDIN, O. Comparative testing of face detection algorithms. In: *Proceedings of the 4th international conference on Image and signal processing*. [S.l.: s.n.], 2010. p. 200–209.

DIETTERICH, T. G. Ensemble methods in machine learning. In: *Lecture Notes in Computer Science*. [S.l.]: Springer Verlag, 2000. p. 1–15.

DONG, J.-X.; KRZYZAK, A.; SUEN, C. Y. A fast svm training algorithm. In: *First International Workshop on Pattern Recognition with Support Vector Machines*. [S.l.: s.n.], 2002. p. 53–67.

DONG, J.-X.; KRZYZAK, A.; SUEN, C. Y. Fast svm training algorithm with decomposition on very large data sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 4, p. 603–618, 2005.

DUIN, R. P. W.; TAX, D. M. Experiments with classifier combining rules. In: *First International Workshop on Multiple Classifier Systems*. [S.l.: s.n.], 2000. p. 16–29.

FAN, R.-E.; CHEN, P.-H.; LIN, C.-J. Working set selection using the second order information for training svm. *Journal of Machine Learning Research*, v. 6, p. 1889–1918, 2005.

FARKAS, L. G. *Anthropometry of the Head and Face*. [S.l.]: Raven Press, 1994.

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. Unpublised manuscript available electronically (on author web pages, or by email request). An extended abstract appeared in Computational Learning Theory, Springer-Verlag, 1995. 1995.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: *Proceedings of the Thirteenth International Conference on Machine Learning*. [S.l.: s.n.], 1996. p. 325–332.

FREUND, Y.; SCHAPIRE, R. E. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, v. 5, n. 14, p. 771–780, 1999.

FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: A statistical view of boosting. *The Annals of Statistics*, v. 28, n. 2, p. 337–407, 2000.

GALTIER, V.; PIETQUIN, O.; VIALLE, S. Adaboost parallelization on pc clusters with virtual shared memory for fast feature selection. In: *Signal Processing and Communications*. [S.l.: s.n.], 2007. p. 165–168.

GEORGHIADES, A. S.; BELHUMEUR, P. N.; KRIEGMAN, D. J. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 6, p. 643–660, 2001.

GONZALEZ, R. C.; WOODS, R. E. *Processamento digital de imagens*. [S.l.]: Addison Wesley, 2010.

GRAMA, A. Y.; GUPTA, A.; KUMAR, V. Isoefficiency: Measuring the scalability of parallel algorithms and architectures. *IEEE Parallel and Distributed Technology*, v. 1, n. 3, p. 12–21, 1993.

HADID, A. The local binary pattern approach and its applications to face analysis. In: *First Workshop on Image Processing Theory, Tools and Applications*. [S.l.: s.n.], 2008. p. 1–9.

HADID, A.; PIETIKÄINEN, M.; AHONEN, T. A discriminative feature space for detecting and recognizing faces. In: *IEEE Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2004. p. 797–804.

HAN, C. H.; SIM, K.-B. Real-time face detection using adaboost algorithm. In: *International Conference on Control, Automation and Systems.* [S.l.: s.n.], 2008. p. 1892–1895.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation.* [S.l.]: Prentice Hall, 1999.

HILL, M. D.; MARTY, M. R. Amdahls law in the multicore era. *Computer*, v. 41, n. 7, p. 33–38, 2008.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. *A Pratical Guide to Support Vector Classification.* 2009.

HUANG, C.; AI, H.; LI, Y.; LAO, S. Vector boosting for rotation invariant multi-view face detection. In: *10th IEEE International Conference on Computer Vision.* [S.l.: s.n.], 2005. p. 446–453.

HUANG, C.; AI, H.; LI, Y.; LAO, S. High-performance rotation invariant multiview face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 4, p. 671–686, 2007.

HUANG, Z.; SHI, X. A distributed parallel adaboost algorithm for face detection. In: *Intelligent Computing and Intelligent Systems (ICIS).* [S.l.: s.n.], 2010. p. 147–150.

ISARD, M.; BLAKE, A. Condensation - conditional density propagation for visual tracking. *International Journal of Computer Vision*, v. 29, n. 1, p. 5–28, 1998.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 1, p. 4–37, 2000.

JAIN, V.; LEARNED-MILLER, E. *FDDB: A Benchmark for Face Detection in Unconstrained Settings.* [S.l.], 2010. Relatório Técnico UM-CS-2010-009.

JAIN, V.; LEARNED-MILLER, E. Online domain adaption of a pre-trained cascade of classifiers. In: *IEEE Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2011. p. 577–584.

JAIN, V.; MUKHERJEE, A. *The Indian Face Database*. 2002.

JESORSKY, O.; KIRCHBERG, K. J.; FRISHHOLZ, R. W. Robust face detection using the hausdorff distance. In: *Third International Conference on Audio and Video-based Biometric Person Authentication*. [S.l.: s.n.], 2001. p. 90–95.

JIN, H.; LIU, Q.; LU, H.; TONG, X. Face detection using improved lbp under bayesian framework. In: *Third International Conference on Image and Graphics (ICIG'04)*. [S.l.: s.n.], 2004. p. 306–309.

JONES, M.; VIOLA, P. *Fast Multi-view Face Detection*. [S.l.], 2003. Relatório Técnico TR2003-96.

KIENZLE, W.; BAKIR, G.; FRANZ, M.; SCHöLKOPF, B. Face detection - efficient and rank deficient. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2005. p. 673–680.

KITTLER, J.; HATEF, M.; DUIN, R. P.; MATAS, J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 3, p. 226–239, 1998.

KRESININ, I. A.; SEREDIN, O. S. Excluding cascading classifier for face detection. In: *International Conference on Computer Graphics and Vision*. [S.l.: s.n.], 2009. p. 380–381.

KUNCHEVA, L. I. *Combining Pattern Classifiers: Methods and Algorithms*. [S.l.]: Wiley Interscience, 2004.

LAZAREVIC, A.; OBRADOVIC, Z. Boosting algorithms for parallel and distributed learning. *Distributed and Parallel Databases*, n. 11, p. 203–229, 2002.

LEE, K.; HO, J.; KRIEGMAN, D. Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 27, n. 5, p. 684–698, 2005.

LI, S. Z.; ZHANG, Z. Floatboost learning and statistical face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 9, p. 1112–1123, 2004.

LIENHART, R.; KURANOV, A.; PISAREVSKY, V. *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection.* [S.l.], 2002. Relatório Técnico Microprocessor Research Lab and Intel Labs.

LIN, C.; FAN, K.-C. Human face detection using geometric triangle relationship. In: *15th International Conference on Pattern Recognition (ICPR'00).* [S.l.: s.n.], 2000. v. 2, p. 941–944.

MARCEL, S.; RODRIGUEZ, Y. *Torch3Vision Machine Vision Library.* 2010. Http://torch3vision.idiap.ch.

MASIP, D.; BRESSAN, M.; VITRIÀ, J. Feature extraction methods for real-time face detection and classification. *EURASIP Journal on Applied Signal Processing*, v. 2005, n. 13, p. 2061–2071, 2005.

MATAS, J.; HAMOUZ, M.; JONSSON, K.; KITTLER, J.; LI, Y.; KOTROPOULOS, C.; TEFAS, A.; PITTAS, I.; TAN, T.; YAN, H.; SMERALDI, F.; CAPDEVIELLI, N.; GERSTNER, W.; ABDELJAOUED, Y.; BIGUN, J.; BEN-YACOUB, S.; MAYORAZ, E. Comparison of face verification results on the xm2vts database. In: *International Conference on Pattern Recognition.* [S.l.: s.n.], 2000. p. 858–863.

MCCANE, B.; NOVINS, K. On training cascade face detectors. In: *Image and Vision Computing.* [S.l.: s.n.], 2003. p. 239–244.

MERLER, S.; CAPRILE, B.; FURLANELLO, C. Parallelizing adaboost by weights dynamics. *Computational Statistics & Data Analysis*, v. 51, p. 2487–2498, 2007.

MEYNET, J.; ARSAN, T.; MOTA, J. C.; THIRAN, J. P. *Fast Multiview Face Tracking with Pose Estimation.* [S.l.], 2007. Relatório Técnico TR-ITS.2007.01.

MEYNET, J.; POPOVICI, V.; SORCI, M.; THIRAN, J.-P. Combining svms for face class modeling. In: *European Signal Processing Conference - EUSIPCO.* [S.l.: s.n.], 2005. p. 1–16.

MEYNET, J.; POPOVICI, V.; THIRAN, J.-P. Mixture of svms for face class modeling. In: *Workshop on Machine Learning for Multimodal Information Management.* [S.l.: s.n.], 2005. p. 173–181.

ODETAYO, M. O. Empirical study of the interdependencies of genetic algorithm parameters. In: *23rd EUROMICRO Conference on New Frontiers of Information Technology*. [S.l.: s.n.], 1997. p. 639–643.

OJALA, T.; PIETIKÄINEN, M.; HARWOOD, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, v. 29, n. 1, p. 51–59, 1996.

OJALA, T.; PIETIKÄINEN, M.; MÄENPÄÄ, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, p. 971–987, 2002.

OPEN Source Computer Vision Library - OpenCV. 2010. Http://opencv.willowgarage.com/wiki/.

OSUNA, E.; FREUND, R.; GIROSI, F. Training support vector machines: an application to face detection. In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. [S.l.]: IEEE Computer Society, 1997. p. 130–136.

PAPAGEORGIOU, C. P.; OREN, M.; POGGIO, T. A general framework for object detection. In: *Sixth International Conference on Computer Vision*. [S.l.: s.n.], 1998. p. 555–562.

PEREIRA, E. T.; GOMES, H. M.; CARVALHO, J. M. de. Integral local binary patterns: a novel approach suitable for texture-based object detection tasks. In: *Conference on Graphics, Patterns and Images, 23rd (SIBGRAPI)*. [S.l.: s.n.], 2010. p. 201–208.

PEREIRA, E. T.; GOMES, H. M.; MOURA, E. S.; CARVALHO, J. M. de; ZHANG, T. Investigation of local and global features for face detection. In: *IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP)*. [S.l.: s.n.], 2011. p. 114–121.

PHILIPS, P. J.; MOON, H. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 10, p. 1090–1104, 2000.

PORIKLI, F. Integral histogram: A fast way to extract histograms in cartesian spaces. In: *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).* [S.l.: s.n.], 2005. p. 829–836.

RABENSEIFNER, R.; HAGER, G.; JOST, G. Hybrid mpi/openmp parallel programming on clusters of multi-core smp nodes. In: *17th International Eumicro Conference on Parallel, Distributed and Network-Based Processing.* [S.l.: s.n.], 2009. p. 427–436.

RAMIREZ, G. A.; FUENTES, O. Face detection using combinations of classifiers. In: *Proceedings of The 2nd Canadian Conference on Computer and Robot Vision.* [S.l.: s.n.], 2005. p. 610–615.

RODRIGUEZ, Y. *Face Detection and Verification using Local Binary Patterns.* Tese (Doutorado) — Ecole Polytechnique Federale de Lausanne, 2006.

RODRIGUEZ, Y.; CARDINAUX, F.; BENGIO, S.; MARIÉTHOZ, J. Measuring the performance of face localization systems. *Image and Vision Computing Journal*, v. 24, n. 8, p. 882–893, 2006.

ROWLEY, H.; BALUJA, S.; KANADE, T. *Rotation Invariant Neural Network-Based Face Detection.* [S.l.], 1997. Relatório Técnico CMU-CS-97-201.

ROWLEY, H.; BALUJA, S.; KANADE, T. Neural network-basead face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, p. 23–38, 1998a.

ROWLEY, H.; BALUJA, S.; KANADE, T. Rotation invariant neural network-based face detection. In: *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 1998b. p. 38–44.

SAUQUET, T.; MARCEL, S.; RODRIGUEZ, Y. *Multiview Face Detection.* [S.l.], 2005. Relatório Técnico IDIAP-RR 05-49.

SCHAPIRE, R. E.; SINGER, Y. Improved boosting algorithms using confidence-rated predictions. In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory.* [S.l.: s.n.], 1998. p. 80–91.

SCHNEIDERMAN, H.; KANADE, T. A statistical method for 3d object detection applied to faces and cars. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2000a. p. 746–751.

SCHNEIDERMAN, H.; KANADE, T. A statistical model for 3d object detection applied to faces and cars. In: *IEEE Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2000b. p. 1–6.

SHI, Y. *Reevaluating Amdahls Law and Gustfsons Law.* [S.l.], 1996. Relatório Técnico Computer and Information Sciences Department, Temple University.

SIM, T.; BAKER, S.; BSAT, M. The cmu pose, illumination, and expression database. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 25, n. 12, p. 1615–1618, 2003.

SINHA, P. Qualitative representations for recognition. In: *Proceedings of the Second International Workshop on Biologically Motivated Computer Vision.* [S.l.]: Springer-Verlag, 2002. p. 249–262.

SINHA, P.; BALAS, B.; OSTROVSKY, Y.; RUSSEL, R. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Invited Paper, Proceedings of the IEEE*, v. 94, n. 11, p. 1948–1962, 2006.

SMITH, L. I. *A tutorial on Principal Component Analysis.* 2002.

SUNG, K.-K.; POGGIO, T. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 1, p. 39–51, 1998.

TAIGMAN, Y.; WOLF, L. *Leveraging Billions of Faces to Overcome Performance Barriers in Unconstrained Face Recognition.* [S.l.], 2011. Relatório Técnico arXiv:1108.1122v1-4-Aug-2011.

TIAN, Y. li; KANADE, T.; COHN, J. F. Recognizing action units for facial expression analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 23, n. 2, p. 1–19, 2001.

TOEWS, M.; ARBEL, T. Detection over viewpoint via the object class invariant. In: *International Conference on Pattern Recognition.* [S.l.: s.n.], 2006. p. 765–768.

TOEWS, M.; ARBEL, T. Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 9, p. 1567–1581, 2009.

TORRALBA, A.; MURPHY, K. P.; FREEMAN, W. T. Sharing features: efficient boosting procedures for multiclass object detection. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition - CVPR*. [S.l.: s.n.], 2004. p. 762–769.

TORRALBA, A.; MURPHY, K. P.; FREEMAN, W. T. Shared features for multiclass object detection. *Lecture Notes in Computer Science, Toward Category-Level Object Recognition*, v. 4170, p. 345–361, 2006.

TORRALBA, A.; MURPHY, K. P.; FREEMAN, W. T. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 5, p. 854–869, 2007.

VIOLA, P.; JONES, M. Robust real-time object detection. In: *Second International Workshop on Statistical and Computational Theories of Vision – Modeling, Learning, Computing, and Sampling*. [S.l.: s.n.], 2001. p. 1–25.

VIOLA, P.; JONES, M. J. Robust real-time face detection. *International Journal of Computer Vision*, v. 57, n. 2, p. 137–154, 2004.

WANG, H.; LI, P.; ZHANG, T. Proposal of novel histogram features for face detection. In: *ICAPR, Lecture Notes in Computer Science*. [S.l.: s.n.], 2005. p. 334–343.

WEBER, M. *Frontal face dataset*. California Institute of Technology: [s.n.], 2010. Http://www.vision.caltech.edu/html-files/archive.html.

WEICKER, K.; WEICKER, N. Basic principles for understanding evolutionary algorithms. *Fundamenta Informaticae*, v. 55, n. 3–4, p. 387–403, 2002.

WU, B.; AI, H.; HUANG, C.; LAO, S. Fast rotation invariant multi-view face detection based on real adaboost. In: *Sixth International Conference on Automatic Face and Gesture Recognition*. [S.l.: s.n.], 2004. p. 79–84.

XIAOHUA, L.; LAM, K.-M.; LANSUN, S.; JILIU, Z. Face detection using simplified gabor featues and hierarchical regions in a cascade of classifiers. *Pattern Recognition Letters*, n. 30, p. 717–728, 2009.

YAN, S.; SHAN, S.; CHEN, X.; GAO, W. Locally assembled binary (lab) feature with feature-centric cascade for fast and accurate face detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2008. p. 1–7.

YANG, M.-H.; KRIEGMAN, D. J.; AHUJA, N. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 1, p. 34–58, 2002.

ZENG, K.; TANG, Y.; LIU, F. Parallization of adaboost algorithm through hybrid mpi/openmp and transactional memory. In: *19th International Eumicro Conference on Parallel, Distributed and Network-Based Processing*. [S.l.: s.n.], 2011. p. 94–100.

ZHANG, C.; ZHANG, Z. *A Survey of Recent Advances in Face Detection*. [S.l.], 2010. Technical Report MSR-TR-2010-66.

ZHANG, J.; ZHANG, X.-D.; HA, S.-W. A novel approach using pca and svm for face detection. In: *Fourth International Conference on Natural Computation*. [S.l.]: IEEE Computer Society, 2008. p. 29–33.

ZHANG, L.; CHU, R.; XIANG, S.; LIAO, S.; LI, S. Z. Face detection based on multi-block lbp representation. In: *Lecture Notes in Computer Science*. [S.l.]: Springer-Verlag, 2007. p. 11–18.

Apêndice A

Resultados Experimentais Preliminares

Neste capítulo, é apresentada a abordagem proposta e os experimentos realizados com o propósito de validar os extratores de características e o detector de faces implementados para o exame de qualificação de tese de doutorado. Também é feita uma comparação entre os resultados obtidos pelo detector proposto com os resultados de outros detectores, a saber: *Torch3Vision Machine Learning Library Face Detector* (MARCEL; RODRIGUEZ, 2010), *Rotation Invariant Neural Network-Based Face Detector* (ROWLEY; BALUJA; KANADE, 1997) e OpenCV *Face Detector* (BRADSKY; KAEHLER, 2008).

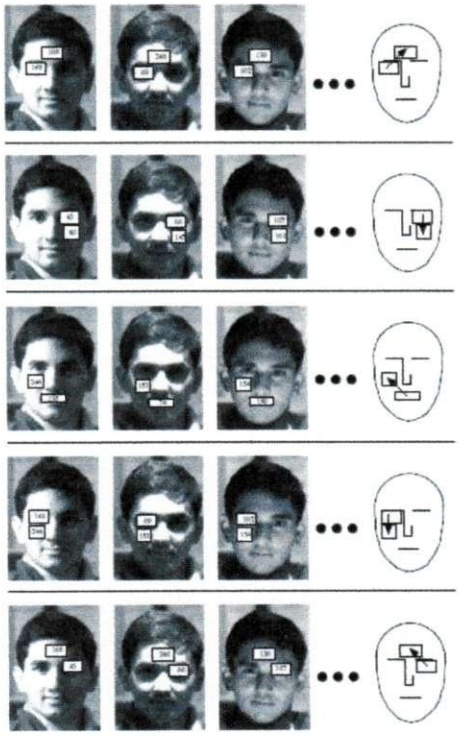
A.1 Extratores de Características

Nesta seção, são apresentadas as abordagens propostas para extração de características de imagens. As abordagens propostas são as Razões Otimizadas de Faces (ROF) e os Padrões Binários Locais Integrais (INTLBP - Integral Local Binary Patterns). Além disso, são descritos os Modelos de Razões de Faces, os Histogramas Integrais e os Padrões Binários Locais, que serviram de inspiração para a criação das ROF e dos INTLBP.

A.1.1 Modelos de Razões de Faces

Sinha (2002) propôs um modelo qualitativo para a representação de imagens humanas, o Modelo de Razões de Faces (*Face Ratios Template* - FRT). O FRT foi desenvolvido a partir de experimentos que comprovaram, dentre outros aspectos, que a razão (quociente) entre as intensidades médias de algumas regiões de imagens de faces possuem comportamento padronizado. Isso quer dizer que, para alguns pares de regiões de imagens de faces, a razão entre suas intensidades médias permanece constante para uma extensa faixa de variação do foco de iluminação, conforme pode ser observado na Figura A.1, na qual são destacadas regiões nas quais, para diferentes fontes de iluminação, a relação entre as regiões, expressa pela razão calculada, permanece a mesma.

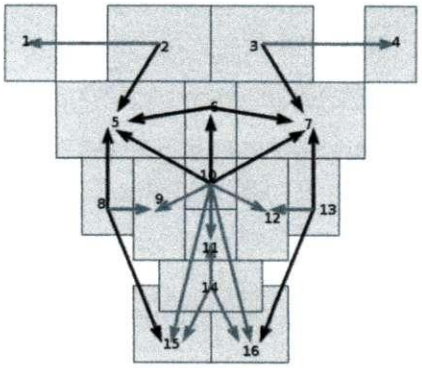
Figura A.1: Exemplos de regiões cujas razões obedecem a um mesmo padrão, independentemente da direção da iluminação.



A partir da observação supramencionada, Sinha (2002) propôs um modelo que indica quais são as regiões da face a partir das quais pode-se extrair razões que obedeçam a esse comportamento padrão. O modelo em questão também indica qual é a direção da variação das intensidades médias de cada região (ver Figura A.2). Por meio de vários experimentos,

Sinha (2002) demonstrou que esse modelo apresenta robustez a variações de iluminação.

Figura A.2: Modelo proposto por Sinha (2002). As setas representam as direções de crescimento das intensidades médias entre pares de regiões da face. Fonte: Sinha (2002).



Apesar de prover um meio rápido para extração de características e de ser robusto a variações de iluminação, o modelo FRT apresenta uma dificuldade intrínseca quando se deseja detectar faces em imagens: trata-se de um modelo *rígido*, que funciona apenas para imagens de faces frontais verticais (sem variações de orientação). Nenhum trabalho foi encontrado na literatura demonstrando o contrário. Com o objetivo de tratar esse problema, propõe-se, a seguir, uma abordagem inspirada em FRT e *Dissociated Dipoles*.

A.1.1.1 Razões Otimizadas de Faces - ROF

Outra abordagem para a extração de razões de faces foi proposta por Pereira et al. (2011). A nova abordagem, denominada Razões Otimizadas de Faces (ROF), constitui uma das contribuições originais da proposta de tese. Ao contrário do modelo FRT, proposto por Sinha (2002), o modelo ROF pode ser otimizado para imagens de faces obtidas de vários pontos de vista (por exemplo, faces frontais e em perfil). O modelo ROF é inspirado no modelo FRT e na técnica *Dissociated Dipoles* (BALAS; SINHA, 2003) e utiliza uma representação integral de imagens (CROW, 1984) para aumentar a velocidade de processamento.

Um dos fatores que traz versatilidade ao modelo ROF é o fato de que as regiões a partir das quais são extraídas as razões são obtidas por meio da otimização de algoritmos genéticos. Portanto, o modelo ROF também pode, em tese, funcionar como método geral para a extração de características de imagens de diferentes objetos (muito embora, até a presente

data, ainda não tenham sido realizados testes experimentais conclusivos para dar suporte a essa afirmação). Para tal propósito, seriam necessários conjuntos adequados de imagens representativas de tais objetos, de modo que esses objetos possuíssem alguma regularidade em sua estrutura.

Pode-se dizer que algoritmos genéticos são técnicas universais de otimização (WEICKER; WEICKER, 2002). Contudo, o melhor conjunto de parâmetros considerados em um dado problema pode não ser o melhor conjunto para outro problema. A escolha do melhor conjunto de parâmetros para a otimização via algoritmos genéticos é uma tarefa árdua e essa escolha tem sido, em geral, baseada na experiência dos usuários ou, nos piores casos, por tentativa e erro. Alguns estudos têm sido conduzidos com o intuito de sistematizar a tarefa de escolha dos parâmetros para a otimização de algoritmos genéticos (DEB; AGRAWAL, 1998; ODETAYO, 1997). Alguns resultados serão comentados neste capítulo para justificar os parâmetros empregados nas otimizações apresentadas.

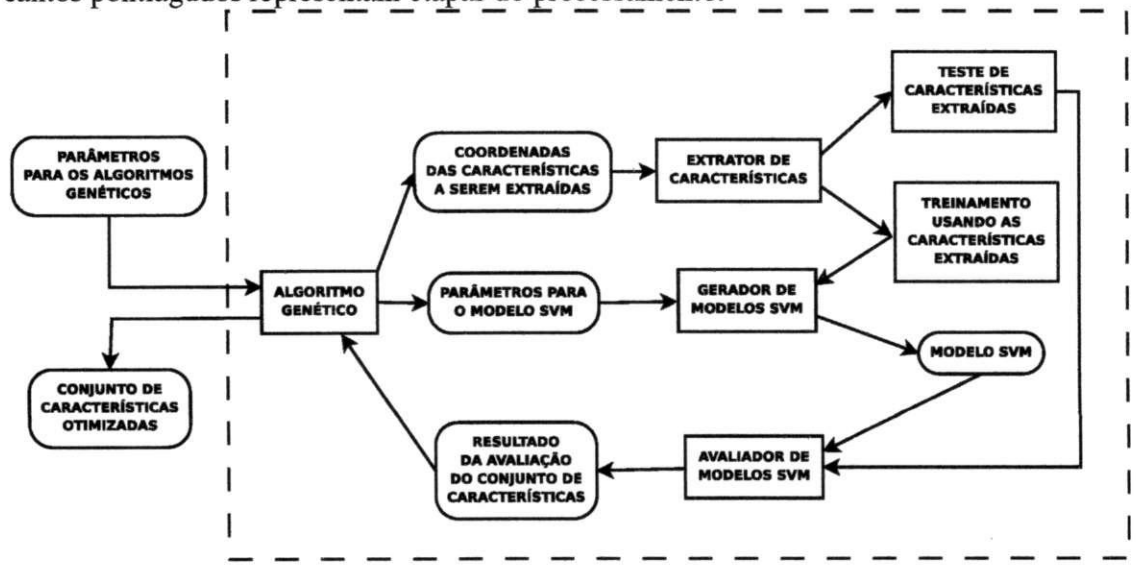
A partir de um método genérico que pode ser aplicado a outras categorias de problemas (detecção de faces em perfil, detecção de outros objetos, e.g., carros), nesta seção propõe-se uma estratégia de otimização para razões de faces. O processo de otimização foi realizado como segue. Em primeiro lugar, os parâmetros do algoritmo genético (quantidade de indivíduos por geração, quantidade de gerações, probabilidade de recombinação e probabilidade de mutação) são passados para o sistema. Com esses parâmetros, o AG cria as coordenadas para regiões candidatas e as repassa ao extrator de características para a geração do modelo SVM.

O AG também gera os parâmetros para o treinamento do modelo SVM (os valores dos parâmetros custo e gama da função de *kernel*). Com as características extraídas e os parâmetros obtidos do AG, um modelo SVM é gerado, e é usado para classificar outro conjunto de imagens (diferente daquele que foi usado para gerar o modelo SVM). As estatísticas obtidas pela avaliação são passadas para a função objetivo do AG. A partir desse ponto, o AG evolui até atingir o número máximo de gerações determinadas pelo usuário ou até o classificador SVM atingir a precisão máxima.

É importante enfatizar que as regiões obtidas são inicialmente extraídas de um conjunto de imagens, as quais são usadas, posteriormente, para a criação de um modelo SVM. Após a

criação do modelo, as mesmas regiões são extraídas de um conjunto de imagens totalmente diferente (conjunto de teste) e são usadas para avaliar a precisão da classificação usando o modelo SVM previamente criado. O processo de otimização de características está representado na Figura A.3.

Figura A.3: Abordagem para a extração de características. As caixas com cantos arredondados representam dados de entrada ou resultados de processamento, enquanto aquelas com cantos pontiagudos representam etapas do processamento.



Cada indivíduo na população do AG representa um conjunto de coordenadas de regiões candidatas. Portanto, o número de fenótipos em cada cromossomo é igual ao número de regiões vezes 8. O número 8 advém do fato de que para cada característica de razão é necessário um par de regiões e cada região é representada por 4 números (x, y, $largura$, $altura$).

Um fator importante do extrator de características na abordagem ROF é seu conjunto de regras para punir más soluções candidatas. Como as características que se deseja extrair são razões entre médias de intensidades de pixels em determinadas regiões das imagens, pode ocorrer que a intensidade média de uma região seja 0; então, se essa média for usada como denominador, ocorrerá um erro de divisão por zero. Portanto, o extrator de características deve verificar a ocorrência de valores 0 em denominadores e prover um meio de punição para os candidatos que apresentarem este evento. O meio adotado para punir tais candidatos é simplesmente não passar as razões que teriam valor 0 no denominador para o vetor de

características. Isso é possível porque a biblioteca utilizada para implementação do método SVM, a LibSVM (FAN; CHEN; LIN, 2005), trabalha com representações esparsas e computa vetores de características com ausência de elementos. A habilidade de a LibSVM lidar com representações esparsas permite que o AG não descarte soluções quando fenótipos impróprios ocorrerem. Portanto, o número máximo de *bons* fenótipos pode ser determinado e o AG buscará boas soluções que contenham de um até o número máximo de bons fenótipos.

Como exemplo, nos experimentos realizados foi estabelecido como número máximo de fenótipos o valor 800. O valor 800 significa que haverá um número máximo de boas regiões igual a 200 (cada uma usando os fenótipos correspondentes às coordenadas x, y e as medidas de $largura$ e $altura$) e 100 razões de faces (correspondendo à divisão de pares de regiões vizinhas na sequência cromossômica). Entretanto, algumas vezes as coordenadas ou os valores de $largura$ e $altura$ são impróprios de modo que o extrator de características usa apenas os valores apropriados para gerar características, o que permite que o AG evolua, mesmo quando houver somente um pequeno número de regiões a serem extraídas. Além disso, conforme ficará claro quando da descrição dos experimentos, tal fato permite que pequenos conjuntos de características que apresentam bons resultados sejam usados.

A.1.2 Histogramas Integrais

Após a aplicação bem sucedida de representações integrais de imagens na extração de características e classificação de padrões (VIOLA; JONES, 2004; LIENHART; KURANOV; PISAREVSKY, 2002; LI; ZHANG, 2004), a atenção da comunidade de análise de padrões tem se voltado para a pesquisa de meios para a detecção de faces em tempo real. Devido a esse objetivo, algumas novas representações de características de imagens foram propostas e uma delas é a representação por meio de Histogramas Integrais (PORIKLI, 2005). Os histogramas são características importantes que podem ser extraídas de imagens e podem ser usados em combinação com outras características para detectar e reconhecer faces.

Os problemas descritos acima levam à busca de novos meios de representação de imagens. Wang, Li e Zhang (2005) argumentam que o melhor compromisso entre a distribuição

estrutural e a retenção de boas propriedades da imagem para a estimação de classe são os histogramas. Logo, é evidente a necessidade de se representar os histogramas de modo que não seja necessário calculá-los para cada nova janela deslizante. Visando atingir esse objetivo, os histogramas integrais foram desenvolvidos, os quais são descritos a seguir.

Para uma imagem com C colunas e L linhas, um histograma integral (PORIKLI, 2005; WANG; LI; ZHANG, 2005) é representado por uma matriz com $(L+1)\times(C+1)$ linhas e N colunas (N corresponde à quantidade de *bins* de cada histograma pré-computado). O valor de cada elemento da matriz pode ser calculado utilizando a Equação A.1 (WANG; LI; ZHANG, 2005).

$$H_{x,y}[N] = \sum_{x' \leq x, y' \leq y} \delta(x', y') \tag{A.1}$$

em que $\delta(x', y') = 1$, ou $\delta(x', y') = 0$, se o valor do pixel (x, y) pertence ou não ao n-ésimo *bin* do histograma. Como em uma representação integral de imagem, o histograma de qualquer localização da imagem pode ser extraído usando poucas recorrências pela Equação A.2 (WANG; LI; ZHANG, 2005).

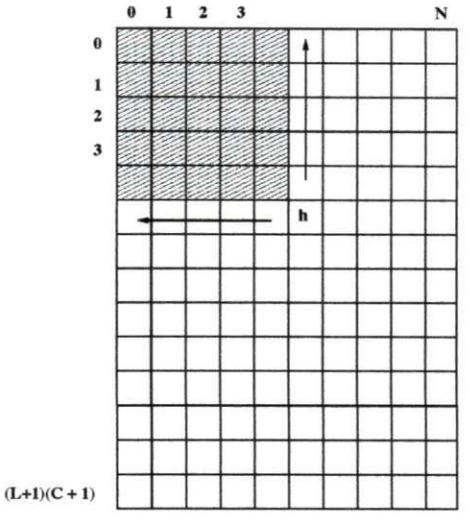
$$\begin{aligned} &H_{x,y}[n] = H_{x-1,y}[n] + h_{x,y}[n], \\ &h_{x,y}[n] = h_{x,y-1}[n] + \delta(x, y), \\ &n = 1, ..., N \end{aligned} \tag{A.2}$$

Conforme pode ser visto na Figura A.4, para todas as colunas em qualquer coordenada (x, y) da matriz de histogramas, pode-se obter o histograma da região de (x,y) para cima e para a esquerda desse ponto. Se o histograma integral for pré-computado em um sistema de detecção de faces, os histogramas de qualquer janela deslizante poderão ser calculados rapidamente em tempo constante para qualquer localização.

A.1.3 Padrões Binários Locais

O operador de Padrões Binários Locais (*Local Binary Patterns* - LBP) tem sido amplamente utilizado para representar, detectar e reconhecer faces (HADID, 2008). Algumas razões para a popularidade do operador LBP são sua simplicidade e robustez a variações de iluminação

Figura A.4: Ilustração de Histograma Integral. Nesta figura, as variáveis L, C e N correspondem às quantidades de linhas e colunas da imagem e a quantidade de bins de um histograma pré-computado, respectivamente.



e orientação. O LBP foi originalmente proposto por Ojala, Pietikäinen e Harwood (1996) como um histograma com N níveis ($N = 2^P$, P é a quantidade de vizinhos em uma máscara pré-determinada) e tem se estendido de vários modos. Uma dessas extensões é o método LBP invariante à rotação (LBP^{ri}) (OJALA; PIETIKÄINEN; MÄENPÄÄ, 2002), no qual há somente $P + 2$ níveis no histograma.

Para representar faces usando o operador LBP, é necessário que a disposição espacial dos elementos faciais seja codificada nos vetores de características extraídos. Para que seja possível obter a codificação de elementos faciais, uma abordagem é dividir a imagem de face em regiões e extrair os histogramas LBP, independentemente, para cada uma dessas regiões, após o que todos os histogramas de cada imagem são concatenados em apenas um vetor de características para representar faces.

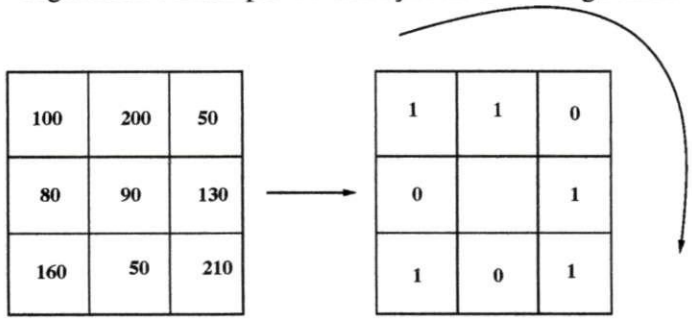
Embora Ojala, Pietikäinen e Mäenpää (2002) afirmem que seu método seja invariante a transformações monotônicas de tons de cinza, outra melhoria foi realizada sobre o método LBP de modo a dar-lhe invariância à iluminação. Jin et al. (2004) propuseram uma variação do método LBP, os Padrões Binários Locais Melhorados (*Improved Local Binary Patterns* - ILBP). Os autores afirmam, demonstrando por experimentação, que o método ILBP apresenta invariância à iluminação. A principal diferença em relação ao método LBP proposto

por Ojala, Pietikäinen e Harwood (1996) é que, ao invés de utilizar o valor de cinza do pixel central como valor para limiarização, o método usa a média dos valores de cinza dos pixels contidos na máscara.

Para testar esse método, Jin et al. (2004) realizaram alguns experimentos com detecção de faces. As classes de faces e não-faces foram modeladas usando gaussianas multivariadas sobre imagens obtidas da base FERET (PHILIPS; MOON, 2000). Os testes foram realizados sobre imagens obtidas das bases YaleB *face dataset* (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001) e MIT-CMU *test set* (ROWLEY; BALUJA; KANADE, 1998a), obtendo taxas de falsos positivos de $2{,}99 \times 10^{-7}$ e verdadeiros positivos acima 90%. Esses resultados indicam que empregar ILBP para extrair características torna o detector de faces robusto a variações de iluminação.

Um código LBP é obtido a partir de uma limiarização de níveis de cinza de pixels. Em tal limiarização, é realizado o deslizamento de uma janela sobre a imagem, sendo o pixel central comparado a seus vizinhos. Se o valor de cinza do vizinho é maior que o valor de cinza do pixel central, o valor 1 é atribuído ao código, caso contrário, o valor 0 é atribuído. O processo de limiarização é exemplificado na Figura A.5, na qual o pixel para o qual está sendo calculado o código LBP possui valor de intensidade 90 e é comparado com os valores de intensidade de seus vizinhos em sentido horário a partir do pixel superior esquerdo. O valor resultante, em binário, para o código do exemplo apresentado na Figura A.5 é 11011010.

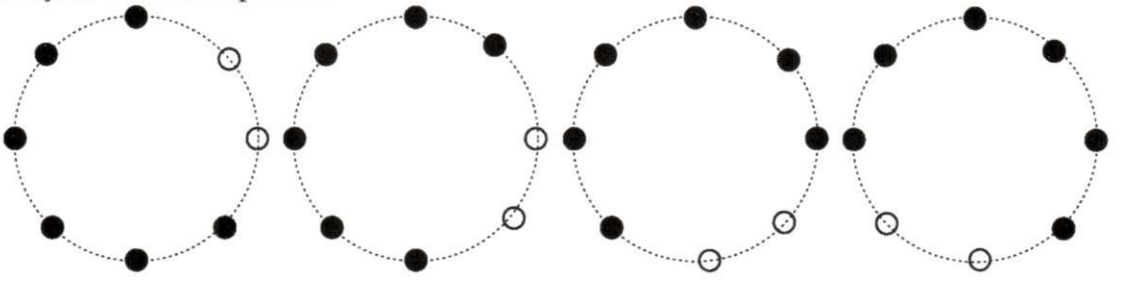
Figura A.5: Exemplo de extração de um código LBP.



A extensão do método LBP proposta por Ojala, Pietikäinen e Mäenpää (2002) é chamada de Padrões Uniformes Invariantes à Escala ($LBP^{ri}_{P,R}$, em que P é a quantidade de vizinhos e R é o raio da vizinhança). O código $LBP^{ri}_{P,R}$ é invariante à rotação de faces no plano. Esta propriedade decorre da observação de que há alguns códigos, chamados de códigos uni-

formes, que podem ser agrupados pelos mesmos rótulos devido a possuírem características semelhantes. Por exemplo, na Figura A.6 há quatro padrões similares. Todos possuem uniformidade igual a 2, sendo seus códigos binários 10011111, 11001111, 11100111, 11110011, respectivamente. A uniformidade é a contagem de transições de 1 para 0, ou de 0 para 1.

Figura A.6: Ilustração de padrões com uniformidade similar e número de vizinhos igual a 8. O mesmo rótulo é atribuído aos padrões apresentados, visto que eles correspondem a rotações do mesmo padrão.



Ojala, Pietikäinen e Mäenpää (2002) propuseram uma medida de similaridade entre códigos LBP baseada na quantidade de transições de 1 para 0 ou vice-versa. Essa medida estabelece que um código é uniforme se ele possui o número de transições menor ou igual a 2. Todos os códigos uniformes possíveis para uma quantidade de vizinhos $P = 8$ são mostrados na Tabela A.1. Como os códigos são circulares, também são apresentadas algumas variações de códigos aos quais são atribuídos o mesmo rótulo. Por exemplo, o código 00000001 possui oito variações que correspondem ao deslocamento do número 1 para a esquerda.

Para extrair os códigos LBP e realizar a medida de uniformidade, são utilizadas as Equações A.3 e A.4, que foram propostas por Ojala, Pietikäinen e Mäenpää (2002). Na Equação A.3, P, R, g_p e g_c indicam a quantidade de pixels vizinhos, o raio de vizinhança, o valor de cinza do pixel vizinho e o valor de cinza do pixel central, respectivamente. A função $s(.)$ indica o sinal de seu argumento. Os códigos com medida de uniformidade menor ou igual a 2 são agrupados sob o mesmo rótulo, e os códigos não uniformes são agrupados sob o rótulo $P + 1$. Na Equação de uniformidade A.4, as diferenças entre os valores de cinza do pixel central são comparadas com os valores de seus vizinhos (a função s verifica o sinal da

Tabela A.1: Os nove diferentes valores de códigos LBP uniformes para $P = 8$.

Código	Exemplos de Variações	Rótulo
00000000	-	0
00000001	00000010, 00000100	1
00000011	00000110, 00001100	2
00000111	00001110, 00011100	3
00001111	00011110, 00111100	4
00011111	00111110, 01111100	5
00111111	01111110, 11111100	6
01111111	11111110, 11111101	7
11111111	-	8

subtração entre seus valores).

$$LBP^{ri2}_{P,R} = \begin{cases} \sum_{p=0}^{P-1} s(g_p - g_c) & \text{se } U(LBP_{P,R}) \leq 2 \\ P+1 & \text{caso contrário} \end{cases} \tag{A.3}$$

em que:

$$\begin{aligned} U(LBP_{P,R}) = &|s(g_{P-1} - g_c) - s(g_0 - g_c)| \\ &+ \sum_{p=1}^{P-1} |s(g_p - g_c) - s(g_{p-1} - g_c)| \end{aligned} \tag{A.4}$$

O número de códigos diferentes deveria ser 2^P, mas devido ao agrupamento dos códigos de uniformidade similares, o número de códigos é reduzido para $P + 2$. Os códigos não-uniformes são agrupados em um único rótulo, chamado de miscelânea (*miscellaneous*). Após a codificação da imagem em LBP uniforme, um histograma com $P + 2$ bins é construído usando os códigos uniformes extraídos. O histograma gerado é usado para representar a imagem e pode ser dado como entrada para classificadores em tarefas de aprendizagem. Para a tarefa particular de aprendizagem de imagens de faces, a disposição dos componentes da face é descrita pela subdivisão da imagem em regiões.

Histogramas diferentes são extraídos para cada região e os histogramas são agrupados em um único vetor. Por exemplo, se a imagem for dividida em 9 regiões e $P = 4$, o vetor de características final terá 54 elementos ($9 \times (P + 2)$). Portanto, além de obter invariância à rotação, o operador LBP também produz representações compactas de texturas de imagens.

Com o intuito de realizar quantizações em qualquer resolução e obter um operador multi-resolução, Ojala, Pietikäinen e Mäenpää (2002) propuseram a criação de vetores de características compostos por características extraídas de variações de P e R (quantidade de vizinhos e raio do centro até a vizinhança, respectivamente). Contudo, há algumas críticas a essa abordagem. Em primeiro lugar, os histogramas extraídos utilizando diferentes valores de P e de R devem ser estatisticamente independentes. Tal independência, em geral, não é garantida. Além disso, a quantidade de elementos no histograma final pode vir a ser bastante elevada. Por exemplo, se a imagem for dividida em 9 regiões e os valores de P forem iguais a 4, 8, 16 e 24, haverá uma quantidade total de elementos igual a $(4+2)+(8+2)+(16+2)+(24+2) \times 9 = 540$. Portanto, a quantidade de elementos no vetor final é dependente da quantidade de resoluções a partir das quais os códigos LBP são extraídos. Essa quantidade pode crescer, reduzindo desta forma, a velocidade de processamento. Finalmente, o agrupamento de códigos uniformes similares em rótulos únicos pode induzir à perda de precisão nos resultados, conforme é demonstrado experimentalmente no Apêndice A.

A.1.4 Padrões Binários Locais Integrais

Embora a extração de códigos LBP seja razoavelmente rápida, se ela for executada milhares de vezes, o tempo de processamento será elevado. A maioria dos algoritmos de detecção de faces emprega algum método de deslizamento de janela para buscar faces em imagens. Para entender o problema do tempo de processamento, considere-se uma imagem de resolução 320×240 pixels. Se o tamanho da janela for 20×20 pixels e o passo de deslocamento for 5, as características serão extraídas mais de 2600 vezes para somente uma escala. Se cada extração durar $1/2600$ segundos, o deslizamento da janela por toda a imagem levará mais de 1 segundo apenas para a primeira escala, o que tornará o método impraticável para processamento em tempo real.

Pereira, Gomes e Carvalho (2010) propuseram uma abordagem que consiste em pré-processar os histogramas LBP acessando, a cada deslizamento da janela, a matriz pré-computada de histogramas LBP. Portanto, essa abordagem possui dois estágios principais. Primeiro, uma imagem LBP é criada, composta pela substituição dos valores dos pixels por seus códigos LBP correspondentes. Segundo, um histograma integral é extraído da imagem LBP. A execução desses estágios de pré-processamento antes do deslizamento da janela aumenta a velocidade de extração de características e, ao mesmo tempo, melhora o processo de detecção de faces em dois aspectos: invariância à escala e rotação. A invariância a rotações no plano advém da abordagem LBP^{ri}, enquanto a invariância à escala, dos histogramas integrais.

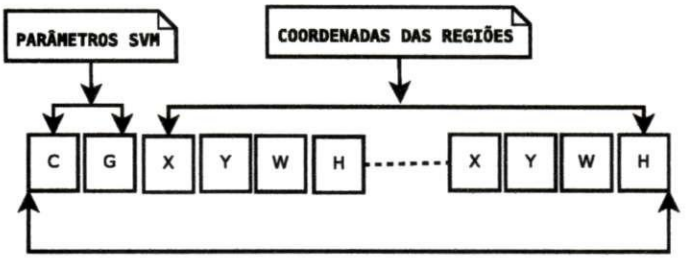
Há uma diferença importante entre o método LBP invariante à rotação (LBP^{ri}) proposto por Ojala, Pietikäinen e Mäenpää (2002) e o método LBP invariante à rotação proposto por Pereira, Gomes e Carvalho (2010). O LBP^{ri} agrupa padrões uniformes similares sob o mesmo rótulo. Na abordagem $INTLBP$, todos os padrões uniformes são usados separadamente para formar o histograma final. Por exemplo, para $P = 8$, o histograma LBP^{ri} teria 10 *bins* diferentes e o $INTLBP$ teria 128 *bins*. Embora a quantidade de bins seja muito mais alta, o tempo final necessário para processar uma imagem é menor do que o tempo necessário usando LBP^{ri}. Conforme será explicado no Apêndice A, a utilização de todos os valores uniformes como rótulos diferentes possibilita melhor precisão nos resultados.

A.2 Experimentos e Resultados Para Razões Otimizadas de Faces

Dois tipos de experimentos foram realizados: (a) extração de características e (b) classificação de imagens de faces frontais e em perfil. O primeiro tipo de experimento teve como objetivo extrair o melhor conjunto de razões, de modo a maximizar a precisão da classificação entre as classes faces e não-faces. O segundo objetivou validar os resultados os obtidos pela otimização de algoritmos genéticos. Tal validação foi realizada mediante a geração de modelos SVM, a partir das características extraídas. Uma série de experimentos foi executada para escolher o melhor conjunto de parâmetros a ser usado como entrada para o AG.

Para entender melhor o processo de escolha dos parâmetros, uma visão esquemática de um cromossomo típico é representada na Figura A.7.

Figura A.7: Ilustração de um cromossomo típico usado nas otimizações de algoritmos genéticos.



A otimização de algoritmos genéticos (AG) usou 2400 imagens para treinamento e 2400 imagens para testes (1200 imagens de faces e 1200 imagens de não faces em cada conjunto). A justificativa para se utilizar apenas 2400 imagens decorre do que foi exposto por Jain, Duin e Mao (2000), que afirmaram que o desempenho de um classificador depende da relação entre os tamanhos das amostras, das quantidades de características e da complexidade do classificador. Dessa afirmação, é extraída uma regra para determinação da quantidade mínima de amostras necessárias para se treinar um classificador. Segundo a regra, a quantidade mínima de amostras deve ser igual a 10 vezes a quantidade de características de cada amostra. Como a quantidade máxima de razões que será extraída na otimização de algoritmos genéticos é igual a 100, foi estabelecido que seriam usadas 1200 imagens de cada classe.

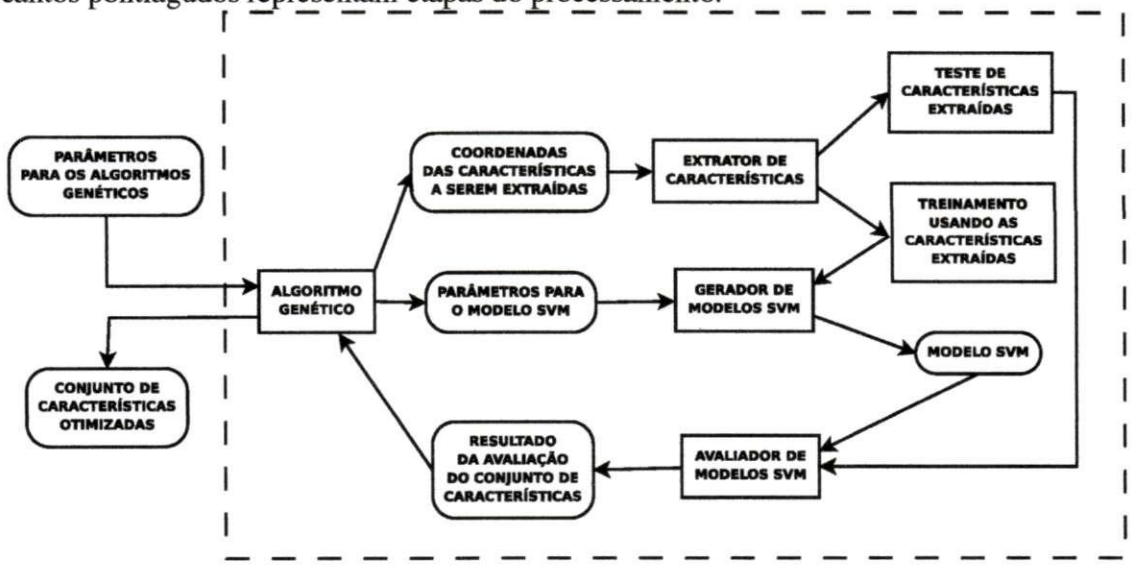
Conforme pode ser visto na Figura A.7, o AG otimiza as coordenadas das regiões e dois parâmetros que serão usados pelo classificador SVM. O AG foi otimizado com uma população de estado estável (*steady state population*). Nessa população, uma quantidade previamente determinada de indivíduos é criada e adicionada à população atual. Então, os escores de todos os indivíduos são medidos e somente parte deles *sobrevive* na próxima geração. Para escolher os melhores parâmetros para a otimização final, algumas otimizações intermediárias foram executadas. Essas otimizações foram baseadas em princípios propostos por Weicker e Weicker (2002) e por Deb e Agrawal (1998).

Segundo o primeiro princípio, um valor de probabilidade de mutação alto não é bom para o processo de otimização. Embora a mutação ajude ao algoritmo genético a sair de platôs, se essa probabilidade for alta, o AG gerará soluções inapropriadas e a evolução não será suave.

Outro princípio usado é que a probabilidade de recombinação não pode ser nem muito alta nem muito baixa. Se for muito alta, a probabilidade de recombinação agirá demasiadamente na busca de boas soluções; se for muito baixa, não ajudará na exploração de boas soluções.

As imagens de faces foram extraídas dos conjuntos: *YaleB Image Database* (LEE; HO; KRIEGMAN, 2005), *Color Feret Image Database* (PHILIPS; MOON, 2000) e *BioID Image Database* (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001). As imagens de não-faces foram extraídas de imagens obtidas da *World Wide Web*. Cada imagem possui resolução 29×35 pixels e, para as imagens de faces, a área de face está contida na imagem (incluindo todo o contorno da cabeça). A justificativa para o uso da imagem da cabeça inteira provém de duas observações. O processo de otimização é ilustrado na Figura A.8.

Figura A.8: Abordagem para a extração de características. Os blocos com cantos arredondados representam dados de entrada ou resultados de processamento, enquanto aqueles com cantos pontiagudos representam etapas do processamento.



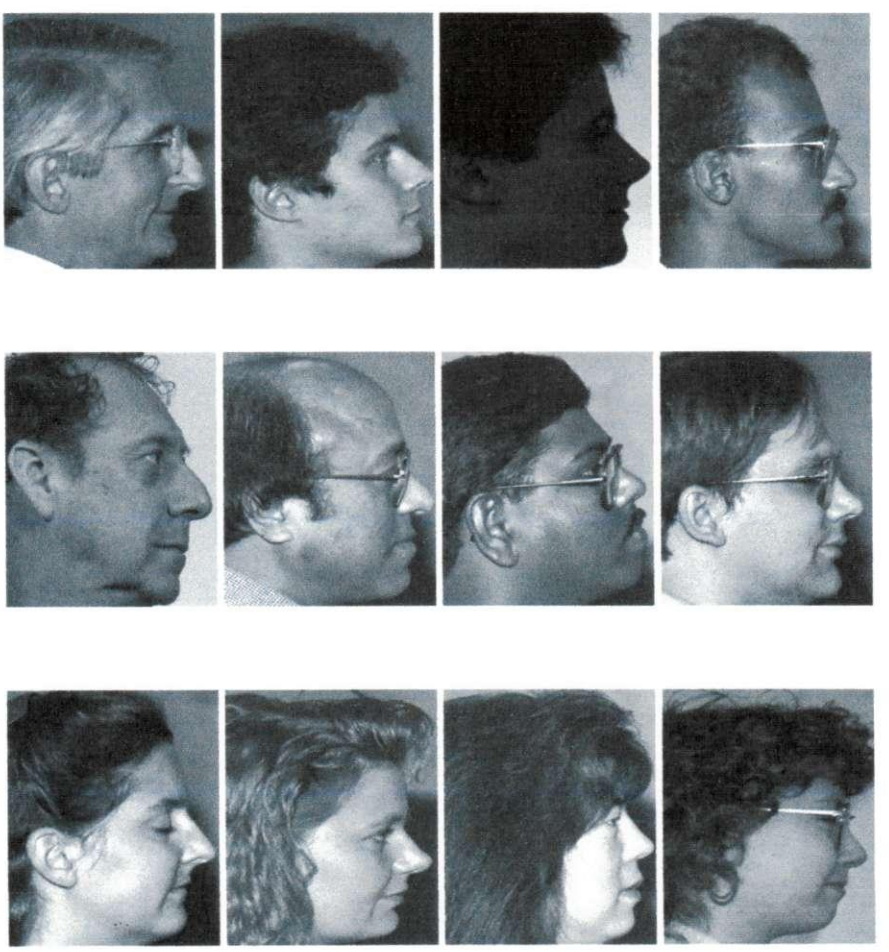
A primeira observação foi apresentada por Rodriguez et al. (2006). Os autores argumentam que não há um método universal de avaliação da precisão de detectores de faces e a medida da precisão dos sistemas de localização deve estar relacionada ao objetivo de aplicação dos resultados. Portanto, como a maioria dos detectores de faces estão relacionados a sistemas de reconhecimento ou identificação de faces. Na abordagem proposta neste apêndice toda a face será usada.

A segunda observação foi proposta por Sinha et al. (2006). Os autores demonstram que o sistema visual humano usa a informação da face holisticamente para realizar detecção e reconhecimento. Esse fato implica que as características externas da face (linha do queixo, orelhas e linha do cabelo, por exemplo) são importantes para detecção e reconhecimento. Portanto, se é desejado que um sistema de detecção de faces atue de modo inspirado biologicamente, tal sistema deve levar em consideração as características externas da face e não apenas as características internas (e.g., olhos, nariz e boca). Na Figura A.9, são mostrados exemplos de imagens frontais de faces e na Figura A.10, são mostradas imagens de faces vistas de perfil usadas nos experimentos para avaliação da abordagem proposta.

Figura A.9: Exemplos de imagens de faces frontais usadas na otimização de algoritmos genéticos com modelos SVM.

Portanto, os parâmetros do AG foram escolhidos por experimentação. Os experimentos foram divididos em dois estágios. Primeiro, a quantidade de gerações, o tamanho da população e a probabilidade de reposição foram fixados em 1500, 25 e 0,6, respectivamente. Os valores para probabilidades de mutação e recombinação foram variados. Para prover melhor entendimento das razões pelas quais alguns parâmetros foram escolhidos para o próximo estágio de experimentos, nas Tabelas A.2 e A.3 listam-se os melhores valores obtidos pelas combinações de probabilidades de cruzamento e mutação para faces frontais e em perfil, respectivamente.

Figura A.10: Exemplos de imagens de faces em perfil usadas na otimização de algoritmos genéticos com modelos SVM.



As baixas probabilidades de mutação usadas nos experimentos são justificadas pelo fato

Tabela A.2: Melhores resultados para cada um dos seis experimentos realizados para auxiliar a escolha do melhor par de valores de probabilidades de mutação e cruzamento para imagens de faces frontais.

Experimento	Mutação	Recombinação	Taxa de Acerto (%)
1	0,01	0,8	96,9295
2	0,05	0,8	95,8921
3	0,1	0,8	92,8631
4	0,01	0,9	95,8506
5	0,05	0,9	95,8506
6	0,1	0,9	92,8216

Tabela A.3: Melhores resultados para cada um dos seis experimentos realizados para auxiliar a escolha do melhor par de valores de probabilidades de mutação e cruzamento para imagens de faces de perfil.

Experimento	Mutação	Recombinação	Taxa de Acerto (%)
1	0,01	0,8	96,2827
2	0,05	0,8	95,9615
3	0,1	0,8	95,3648
4	0,01	0,9	96,4204
5	0,05	0,9	96,2827
6	0,1	0,9	95,6402

de que altos valores poderiam introduzir instabilidade à evolução e *comprometer* boas soluções. O objetivo de se ter um pequeno número de indivíduos na população é aumentar a velocidade da evolução pois, dependendo dos valores dos parâmetros do SVM, a criação do modelo e seu teste pode ser lenta. Um ponto importante que deve ser notado é que o número máximo de genes usados é 802, mas somente as coordenadas que obedecerem a algumas regras de integridade serão usadas. Essas regras de integridade permitem que apenas algumas poucas quantidades de características sejam avaliadas pelo SVM aumentando a velocidade de avaliação dos indivíduos. O processo de otimização do AG levou um tempo médio de 36 horas, sendo executado em um processador Intel Centrino Core2 Duo com 1 GB de RAM. Os

resultados obtidos pelos melhores indivíduos de cada geração dos processos de otimização são mostrados nas Figuras A.11, A.12, A.13 e A.14.

Com o intuito de melhorar os resultados para a classificação de faces frontais, buscaram-se melhores parâmetros para a geração de modelos SVM usando um novo conjunto de imagens contendo 2326 imagens de faces e 95990 imagens de não-faces. Essa busca foi realizada a partir da variação dos valores custo e gama da função de *kernel* SVM entre -10 e 10 com passo 2. A busca foi realizada utilizando um teste de validação cruzada. O melhor resultado obtido pelo teste de validação cruzada foi 99,94%. Após a obtenção dos melhores valores para os parâmetros de custo e gama, um teste foi realizado usando um conjunto de imagens diferente para validar os resultados. O conjunto de validação foi composto por 5383 imagens de faces e 102695 imagens de não-faces, tendo sido obtida uma taxa de acerto de 98,8666%.

Uma característica importante do método proposto é que, mesmo o número máximo de características sendo igual a 100, o AG atingiu um resultado ótimo utilizando apenas 21 características. Apesar da taxa de 98,8666% não ser a máxima, o pequeno número de características necessárias pode ser calculado rapidamente, usando a imagem integral, e pode ser combinado com outros métodos para atingir altas taxas de acerto. Nas Tabelas A.4 e A.5, são mostradas todas as regiões obtidas de faces em perfil e frontais para gerar as 21 características otimizadas pelo AG.

Tabela A.4: Regiões otimizadas pelo algoritmo genético para imagens de faces de perfil. Em cada linha, a coluna numerador (N) indica a região cujo valor médio de intensidade será dividido pela região denominador (D) correspondente.

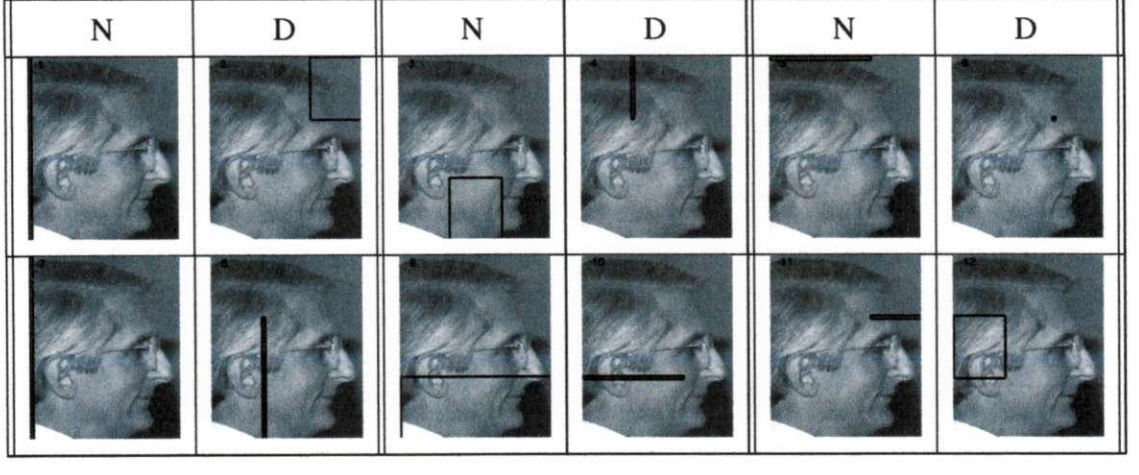
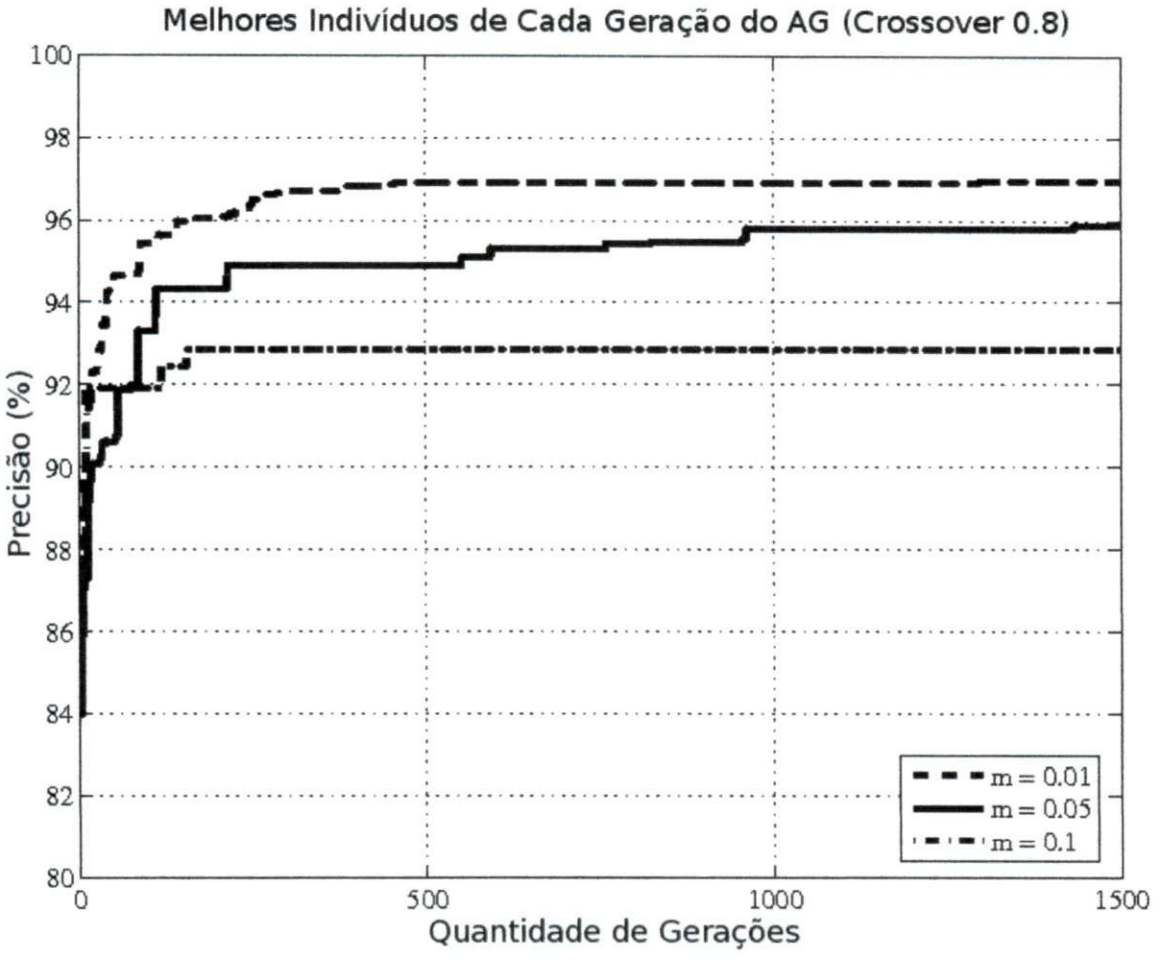


Figura A.11: Melhores indivíduos de cada geração nos processos de otimização utilizando probabilidade de cruzamento igual 0,8



Conforme pode ser visto na Tabela A.5, há algumas similaridades entre as regiões propostas por Sinha (2002) e as regiões obtidas pelo método ora proposto. Além disso, algumas regiões do modelo otimizado pelo AG são idênticas, implicando que o conjunto de regiões pode ser reduzido para 32. Por exemplo, as regiões do modelo de números 11, 32, 24, 34 e 36 são idênticas. Algumas regiões do modelo AG que são similares àquelas do modelo de Sinha (2002) são: GA-14/Sinha-14, GA-7/Sinha-5-6-7 e GA-26/Sinha-1.

Figura A.12: Melhores indivíduos de cada geração nos processos de otimização utilizando probabilidade de cruzamento igual 0,9

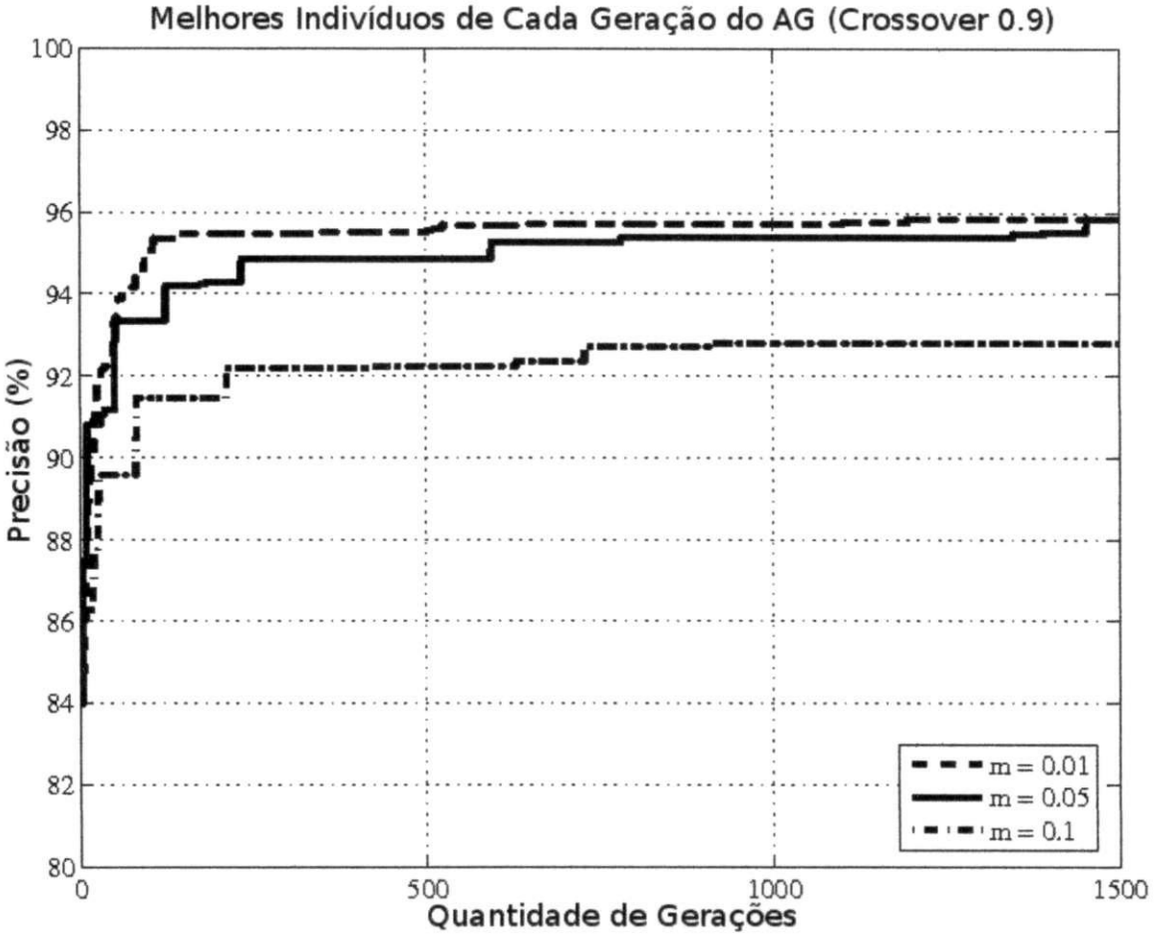


Figura A.13: Melhores indivíduos de cada geração na otimização de algoritmos genéticos para imagens de faces em perfil usando probabilidade de cruzamento 0,8

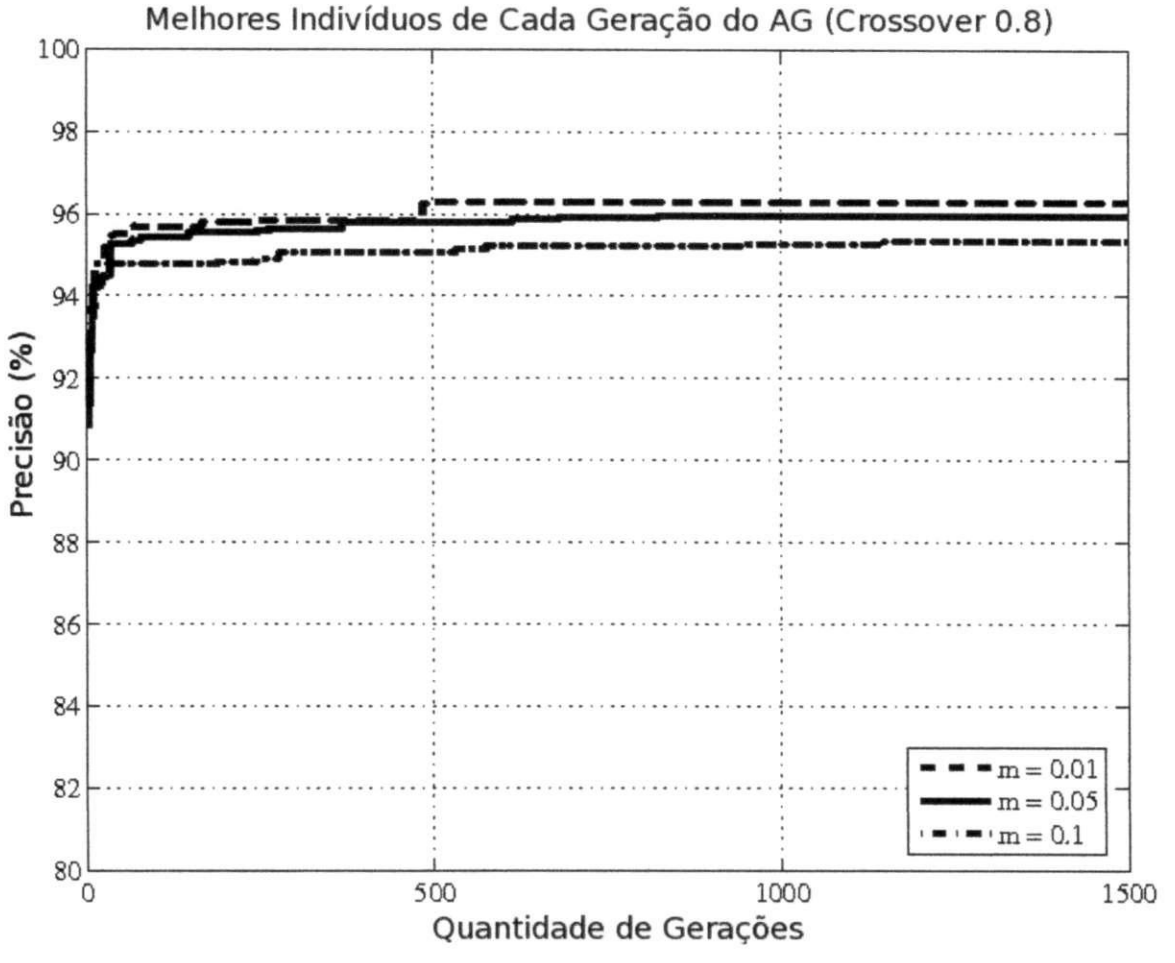


Figura A.14: Melhores indivíduos de cada geração na otimização de algoritmos genéticos para imagens de faces em perfil usando probabilidade de cruzamento 0,9

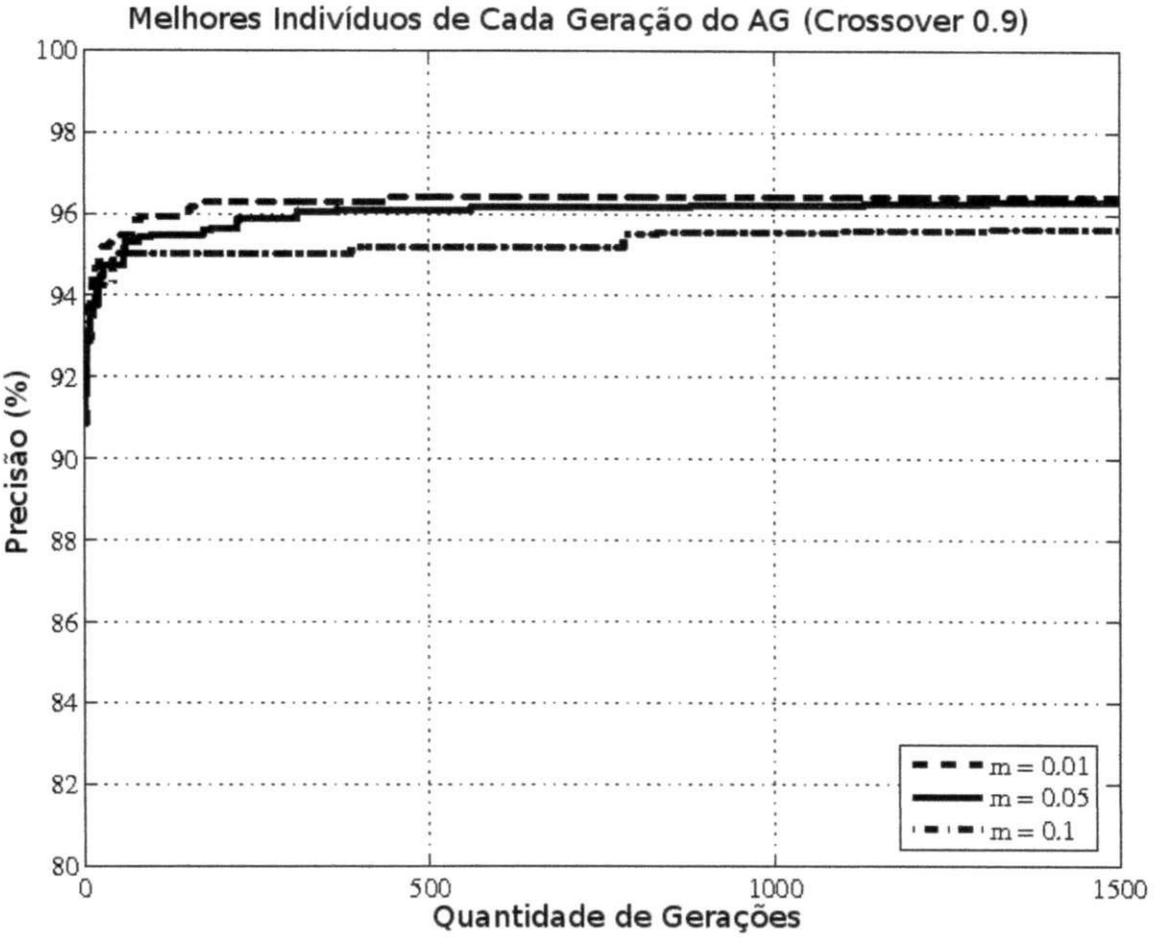


Tabela A.5: Regiões otimizadas pelo algoritmo genético para imagens de faces frontais. Em cada linha, a coluna numerador (N) indica a região cujo valor médio de intensidade será dividido pela região denominador (D) correspondente.

A.3 Experimentos com Histogramas Integrais

Como a intenção da abordagem proposta para a detecção de faces é realizar uma combinação de classificadores, de modo a obter o melhor resultado possível, com alta velocidade de processamento e, tendo em vista que os histogramas integrais são uma ferramenta valiosa para extração de características em tempo real, foram realizados experimentos para verificar a viabilidade da utilização de histogramas de valores de cinza dos pixels das imagens para representá-las. Além disso, já havia trabalhos na literatura especializada sobre o uso de histogramas em detecção de faces. Zhang, Zhang e Ha (2008) afirmam que os histogramas das intensidades dos pixels de imagens de faces possuem comportamento gaussiano, enquanto que os histogramas de imagens que não contém faces não possuem comportamento gaussiano.

Para que seja possível extrair características de imagens de faces de modo a obter boas representações, a disposição espacial dos elementos que compõem a face deve ser representada nas características extraídas. Portanto, a simples extração de um histograma de cinza de uma imagem de face não traria informação relevante sobre a disposição de elementos estruturais tais como: olhos, nariz e boca. Para tratar tal problema, optou-se por dividir a imagem de face em sub-regiões e extrair um histograma para cada região. Em seguida, os histogramas locais são concatenados em um único vetor de características e esse vetor é utilizado para treinamento e testes de classificação de imagens.

Após a decisão sobre a divisão da imagem de face ser tomada, outros problemas surgiram. Em quantas regiões uma imagem de face deveria ser dividida? Quantos *bins* cada histograma componente deveria possuir? O fator que guiou a escolha desses parâmetros foi o tempo de processamento. Como seriam utilizadas SVM para gerar modelos de faces e sabe-se que o tempo de otimização de SVM é sensível à quantidade de elementos do vetor de características, optou-se pela melhor combinação de parâmetros que não degradasse a velocidade de processamento, mas que obtivesse boas taxas de acerto.

O processo de geração de modelos SVM foi praticamente o mesmo utilizado na geração final de modelos SVM para ROF (Razões Otimizadas de Faces), discutidas na subseção anterior. Inicialmente, foram extraídos histogramas de imagens de 29×35 pixels, após a di-

visão das imagens em 9 regiões de tamanhos iguais. Cada histograma continha 15 *bins*. Em seguida, os histogramas foram concatenados gerando um vetor de características contendo 135 elementos. Buscaram-se melhores parâmetros para a SVM, usando validação cruzada. Criou-se o modelo SVM, o qual foi testado sobre um conjunto de características de histogramas extraídos de um conjunto de imagens diferente do que foi usado para gerar o modelo. As características de teste foram extraídas de 26392 imagens de faces e 102695 imagens de não faces, tendo como resultado 98,6982% de acerto.

A.4 Experimentos com Integral LBP

Nesta seção, apresentam-se todos os experimentos realizados para avaliar a extração de características usando Padrões Binários Locais Integrais (INTLBP). Dois tipos de experimentos foram realizados: no primeiro, apresentado na Subseção A.4.1, compara-se o INTLBP com o LBP^{ri}, em relação a variações de escala; e no segundo, apresentado na Subseção A.4.2, comparam-se os requisitos totais de tempo de processamento para os dois métodos. Todos os experimentos foram realizados com um processador *Core 2 Duo Intel Centrino* com $2GB$ de RAM. O *software* foi produzido utilizando GNU $g++$ com o sistema operacional Linux Ubuntu, nenhum processo de usuário estava sendo executado durante os experimentos.

A.4.1 Experimentos de Avaliação de Desempenho

As imagens utilizadas nos experimentos descritos nesta seção foram obtidas das seguintes bases de imagens: FERET database (2505 imagens frontais) (PHILIPS; MOON, 2000), Cohn-Kanade AU-Coded Facial Expression Database (8785 imagens frontais) (TIAN; KANADE; COHN, 2001) e The Extended Yale Face Database B (704 imagens frontais) (LEE; HO; KRIEGMAN, 2005). As imagens de não-faces foram extraídas de imagens obtidas da *World Wide Web*. As imagens de não-faces foram recortadas, redimensionadas e verificadas visualmente. Todas as imagens foram divididas em dois conjuntos: treinamento e teste. O conjunto de treinamento foi composto de 6602 imagens de faces e

24636 imagens de não-faces, enquanto o conjunto de teste foi composto de 6605 imagens de faces e 36187 imagens de não-faces.

A quantidade de imagens de não-faces é maior do que a quantidade de imagens de faces com o intuito de representar a disparidade da quantidade de padrões existentes em imagens reais: em uma imagem, geralmente, há muito mais padrões de não-faces do que padrões de faces. Quatro resoluções de imagens foram testadas: 29×35, 58×70, 116×140 e 232×280 pixels. Um recorte retangular foi usado, ao invés de um recorte quadrado, pois o objetivo desse tipo de recorte era que a imagem contivesse toda a cabeça, incluindo a testa e as linhas do queixo.

O primeiro passo na experimentação foi extrair características LBP das imagens. Duas implementações diferentes de $LBP^{r,i2}_{P,R}$ foram usadas, uma similar à proposta por Ojala, Pietikäinen e Mäenpää (2002) e outra com as modificações aqui propostas. As imagens foram divididas em 25, 16, 9 e 4 regiões com dimensões iguais. Para todos os experimentos, os valores de P e R foram 4 e 2, respectivamente. Outros valores de P maiores do que 4 não foram testados porque o número de elementos no vetor final seria muito grande, comparado à quantidade de pixels nas imagens de menor resolução (29×35 pixels, correspondente a 1015 pixels).

Por exemplo, se a combinação de parâmetros $P = 8$ e a quantidade regiões igual a 25 tivesse sido usada, a quantidade de elementos no vetor final seria igual a 6400 (este número corresponde a $(2^8 \times 25)$. Todos os histogramas foram normalizados, uma vez que a normalização provê robustez a variações de escala, conforme será evidenciado nos experimentos. A quantidade de elementos no vetor de características final depende da quantidade de *bins* em cada histograma e da quantidade de vizinhos do pixel central da máscara LBP. Portanto, para o algoritmo LBP original, usando 25 regiões e $P = 8$, a quantidade total de elementos seria 250, correspondendo a $(8 + 2) \times 25$.

Para os experimentos de classificação e aprendizagem, a biblioteca LIBSVM (FAN; CHEN; LIN, 2005) foi usada para gerar modelos SVM. Um fator relevante em relação à geração de modelos SVM é a escolha de parâmetros. Nos experimentos apresentados nesta subseção, o procedimento proposto por Hsu, Chang e Lin (2009) foi adotado. Esse procedimento é realizado pelo treinamento e teste de diferentes pares de

parâmetros *custo* e *gama* sobre um *kernel* RBF usando validação cruzada. O melhor par de parâmetros é usado para treinar o modelo SVM final.

O primeiro conjunto de resultados é mostrado na Tabela A.6, na qual são condensados os resultados para os experimentos usando imagens que foram divididas em 25 sub-regiões. Pode-se observar que, para todas as resoluções de imagens de faces, os resultados para classificações de não-faces foram praticamente os mesmos (diferindo em menos de 1%). Em todas as tabelas de resultados apresentadas nesta seção, as colunas **VP** e **FP** contêm as taxas de verdadeiro positivo e falso positivo, respectivamente. Associada às taxas de VP e FP, está a medida **FM**, abreviação para o termo *F-measure*. A medida FM corresponde à média harmônica para precisão e *revocação* e é obtida segundo a Equação A.5. As linhas **MÉDIA** e **DP** contém a média e o desvio padrão calculados coluna a coluna das estatísticas acima dessas linhas.

Em relação aos experimentos com imagens divididas em 25 sub-regiões, há somente dois casos nos quais a classificação de não-faces usando 16 *bins* superou a classificação usando 6 *bins* por um valor superior a 1%. Isso ocorreu para resolução de treinamento 232×280 pixels e resoluções de teste 29×35 pixels e 58×70 pixels. Dentre todos os 16 testes, somente em 3 casos as taxas de classificação de faces usando 6 *bins* foram superiores às taxas de faces usando 16 *bins*. Contudo, as diferenças entre os valores maiores e seus correspondentes usando 16 *bins* foram bastante baixas, próximas a 1%.

Houve alguns resultados usando 16 *bins* muito mais altos do que seus correspondentes de 6 *bins*. Por exemplo, quando a resolução de treinamento utilizada foi de 58×70 pixels e a resolução de teste foi de 29×35 pixels, neste caso a diferença foi maior que 30%. Em alguns testes, taxas de classificação nulas ocorreram por causa de grandes disparidades entre as escalas das imagens de teste e de treinamento. Contudo, a maioria dos testes apresentou robustez a variações de escala. Para todas as resoluções de treinamento, os experimentos realizados dividindo as imagens em 25 sub-regiões obtiveram valores médios de *F-measure* maiores para $INTLBP$ do que para LBP^{ri}. Na resolução de treinamento 58×70 pixels, a

Tabela A.6: Comparação de resultados para 16 *bins* (INTLBP) e 6 *bins* (LBP^{ri}) quando as imagens foram divididas em 25 regiões.

Escala		INTLBP			LBP^{ri}		
TREINAMENTO	TESTE	VP	FP	FM	VP	FP	FM
232 x 280	29 × 35	0,00	99,95	0,00	0,05	98,38	0,0009
	58 × 70	25,51	99,97	0,4064	26,77	98,64	0,4179
	116 × 140	97,03	99,99	0,9849	88,84	99,26	0,9372
	232 × 280	99,79	99,99	0,9989	99,44	99,81	0,9962
MÉDIA		55,58	99,98	0,5976	53,78	99,02	0,5880
DP		50,55	0,02	0,4847	48.06	0,64	0,4698
116 × 140	29 × 35	0,00	99,99	0,00	0,08	99,80	0,0016
	58 × 70	90,96	99,99	0,9526	87,77	99,50	0,9324
	116 × 140	99,33	99,99	0,9966	97,53	99,59	0,9855
	232 × 280	93,93	99,98	0,9686	78,38	99,83	0,8780
MÉDIA		71,06	99,99	0,7295	65,94	99,68	0,6994
DP		47,50	0,01	0,4866	44,60	0,16	0,4672
58 × 70	29 × 35	32,58	100	0,4915	0,68	99,89	0,0135
	58 × 70	99,86	99,99	0,9992	97,94	99,78	0,9885
	116 × 140	92,63	99,98	0,9616	31,17	99,84	0,4747
	232 × 280	68,99	99,98	0,8164	2,89	99,97	0,0561
MÉDIA		73,52	99,99	0,8172	33,17	99,87	0,3832
DP		30,31	0,01	0,2310	45,36	0,08	0,4540
29 × 35	29 × 35	99,15	99,97	0,9956	97,94	99,57	0,9874
	58 × 70	32,19	99,99	0,4870	0,03	99,88	0,0006
	116 × 140	0,03	99,99	0,0006	0,00	99,99	0,00
	232 × 280	0,00	100,00	0,00	0,00	99,99	0,00
MÉDIA		32,84	99,99	0,3708	24,49	99,86	0,2470
DP		46,73	0,01	0,4755	48,96	0,20	0,4936

média dos *F-measures* foi 2,13 vezes maior para $INTLBP$ do que para LBP^{ri}.

$$FM = \frac{2 \times VP}{2 \times VP + FN + FP} \tag{A.5}$$

Na Tabela A.7, os resultados dos testes usando imagens subdivididas em 16 regiões são mostrados. Similarmente aos resultados da Tabela A.6, os resultados de classificação de não-faces são muito próximos tanto para os testes de 16 *bins* quanto para os testes de 6 *bins*. Os resultados de classificação de faces usando 6 *bins* foram maiores do que seus correspondentes usando 16 *bins* somente em 2 casos. Ambos ocorreram quando a resolução de treinamento foi 116 × 140 pixels e as resoluções de teste foram 29 × 35 pixels e 58 × 70 pixels, respectivamente.

Usando esse número de 16 regiões, alguns resultados extremamente altos ocorreram quando foram utilizados 16 *bins*, quando comparados com as escalas correspondentes usando 6 *bins*. Por exemplo, para a resolução de treinamento de 29 × 35 pixels e resolução de teste de 58 × 70 pixels, os resultados com 16 bins foram maiores que 92%. Para esse conjunto de resultados, cada resolução foi testada com imagens de resolução duas vezes maior do que a resolução de treinamento, os resultados de classificação de faces foram 90% maiores quando foram usados 16 *bins*. A *adaptabilidade em multi-escala* não ocorreu em nenhum caso usando 6 *bins*. Ao contrário, há alguns casos nos quais os resultados de classificação foram bastante inferiores para esse tipo de testes. Por exemplo, os resultados usando 6 *bins*, com resolução de treinamento de 29 × 35 pixels e resolução de teste de 58 × 70 pixels, alcançaram taxas de acerto ou precisão 90% menores quando comparados com seus correspondentes usando 16 *bins*. Para essa quantidade de regiões, as médias de *F-measure* foi aproximadamente duas vezes maior para $INTLBP$ do que para LBP^{ri} em dois casos, resoluções de treinamento 58 × 70 e 29 × 35 pixels.

Somente quatro testes com 6 *bins* atingiram taxas de acerto maiores que os correspondentes usando 16 *bins* para classificação de faces e dividindo-se a imagem em 9 regiões, conforme pode ser visto na Tabela A.8. Os resultados médios para classificação de faces foram muito similares quando a resolução de treinamento foi 232 × 280 pixels: 48,38% para 16 *bins* e 48,61% para 6 *bins*. Em três testes, os resultados usando 16 *bins* foram maiores

que os testes correspondentes usando 6 bins por um fator maior que 40%.

Na Tabela A.9 são apresentados os resultados para as classificações usando imagens divididas em 4 regiões. Pode ser inferido dos resultados das Tabelas A.8 e A.9 que o uso de menos regiões reduz a precisão da classificação. Por exemplo, em todos os resultados com 16 *bins*, usando 25 e 16 regiões, a precisão dos testes de classificação de faces, quando a resolução de treinamento era a mesma que a resolução de teste, atingiu resultados maiores que 99% de precisão. Contudo, para os mesmos casos usando as quantidades menores de regiões nenhum dos testes obteve 99% de precisão. Outra observação importante é que as diferenças de precisão entre os testes com 16 *bins* e 6 *bins* tornaram-se ainda mais evidentes quando uma pequena quantidade de regiões foi usada.

O único caso em que a média dos *F-measures* foi maior para LBP^{ri} do que $INTLBP$ foi quando a imagem era dividida em 9 regiões e a resolução de treinamento foi de 232×280 pixels, em todos os outros experimentos o $INTLBP$ obteve médias de *F-measures* mais altas que o LBP^{ri}. Além disto, a maior robustez do método $INTLBP$ a variações de escala pode ser melhor percebida quando se utiliza uma quantidade menor de regiões, para o caso de 4 regiões as médias de *F-measures* para $INTLBP$ são muito mais altas que as médias correspondentes para LBP^{ri}.

Os resultados apresentados nas Tabelas A.6, A.7, A.8 e A.9 confirmam uma das hipóteses deste trabalho: o uso de todos os códigos LBP, ao invés de uma quantidade reduzida de códigos rotulados por uniformidades similares, melhora extremamente os resultados de classificação e dá a abordagem LBP uma considerável robustez a variações de escala.

A.4.2 Mensuração de Tempos de Processamento

Existem duas abordagens principais para deslizamento de janelas em sistemas de detecção de objetos. A primeira abordagem é fixar o tamanho inicial da janela e, a cada ciclo de varredura, aumentar (ou diminuir) as dimensões da janela. A segunda abordagem cria uma pirâmide de imagens, cada uma correspondendo a uma diminuição da resolução da imagem original e o deslizamento da janela é realizado sobre cada nível da pirâmide usando uma janela de tamanho fixo para todos os níveis. De modo a avaliar o desempenho de tempo da

extração de características LBP usando o LBP^{ri} e o $INTLBP$, dois tipos de experimentos foram realizados. O primeiro tipo usou um deslizamento de janela com variação de escala, o segundo tipo usou uma pirâmide de imagens em diferentes resoluções. Ambos foram realizados sobre 100 imagens com resolução de 320 × 240 pixels.

A quantidade de janelas analisada sobre uma imagem é uma função de dois parâmetros principais: o fator de escala e o tamanho do passo. O fator de escala determina o grau de aumento no tamanho da janela. Por exemplo, um fator de escala comumente utilizado é 1,1. Portanto, se a janela inicial possui resolução 29 × 35 pixels, o próximo tamanho será 31,9 × 38,5 pixels e a resolução será aumentada até atingir o menor valor das dimensões da imagem. A janela deve ser deslocada considerando um passo pré-definido. O tamanho do passo influenciará a precisão dos resultados e a velocidade de processamento. A mensuração do tempo de processamento pode ser influenciada por vários fatores. Por isso, os experimentos descritos nesta subseção foram realizados utilizando 100 imagens para estimar o tempo médio de processamento. Para os dois tipos de processamento, os fatores de escala foram 1,1, 1,5 e 1,9, os tamanhos de passos foram de 5 a 25 com incrementos de 5 a cada iteração.

Os resultados para a variação de escala dos experimentos de janela deslizante são mostrados na Tabela A.10, que apresenta três colunas: quantidade de janelas, tempo de processamento para a abordagem integral e tempo de processamento para a abordagem tradicional. A partir da análise dos dados, pode ser observado que o $INTLBP$ é menos sensível a variações de quantidade de janelas do que o LBP^{ri}. Por exemplo, quando a quantidade de janelas foi aumentada por um fator de 125,56 (5776200/46000) os tempos de processamento do $INTLBP$ e do LBP^{ri} foram aumentados por um fator de 11,23 (180,75/16,1) e 124,06 (1901,83/15,33), respectivamente.

Além disso, os melhores resultados de tempo registrados para o $INTLBP$ ocorreram quando uma grande quantidade de janelas foi usada (a partir de 69400 janelas). Portanto, considerando que a precisão de classificação aumenta quando mais janelas são usadas na abordagem de janela deslizante, o $INTLBP$ pode ser considerado a melhor escolha para extração de características em termos de precisão de classificação e tempo de processamento.

Os experimentos usando pirâmides de imagens obtiveram, para algumas combinações de parâmetros, a mesma quantidade de janelas. Cinco grupos de três experimentos tiveram

a mesma quantidade de janelas, com diferentes conjuntos de parâmetros. Os tempos de processamento de cada quantidade de janelas são apresentados na Tabela A.11 correspondem à média de tempos obtidos nos três experimentos de mesma quantidade de janelas. As pirâmides de imagens foram formadas por quatro níveis, cada nível correspondendo a resoluções de 100%, 75%, 50% e 25% da imagem original.

No caso das pirâmides de imagens, o tempo de processamento do $INTLBP$ foi melhor do que o tempo de processamento do LBP^{ri} somente nos experimentos com maiores quantidades de imagens. A explicação para tal comportamento advém do fato de que, a cada iteração, a imagem LBP deve ser criada quatro vezes e a criação da imagem LBP é a parte mais onerosa do algoritmo $INTLBP$. Contudo, conforme foi observado nos resultados dos experimentos sintetizados na Tabela A.10, o crescimento do tempo de processamento do $INTLBP$ é menos acentuado do que o tempo correspondente usando LBP^{ri}. Pode ser observado na Tabela A.11 que quando a quantidade de janelas passa de 220800 para 836000, os aumentos nos tempos de processamentos são de 1,43 e 3,80, para $INTLBP$ e LBP, respectivamente. Os gráficos das Figuras A.15 e A.16 enfatizam os melhores resultados do $INTLBP$.

Figura A.15: Representação gráfica dos dados da Tabela A.10. Dados obtidos dos experimentos com redimensionamento da janela deslizante.

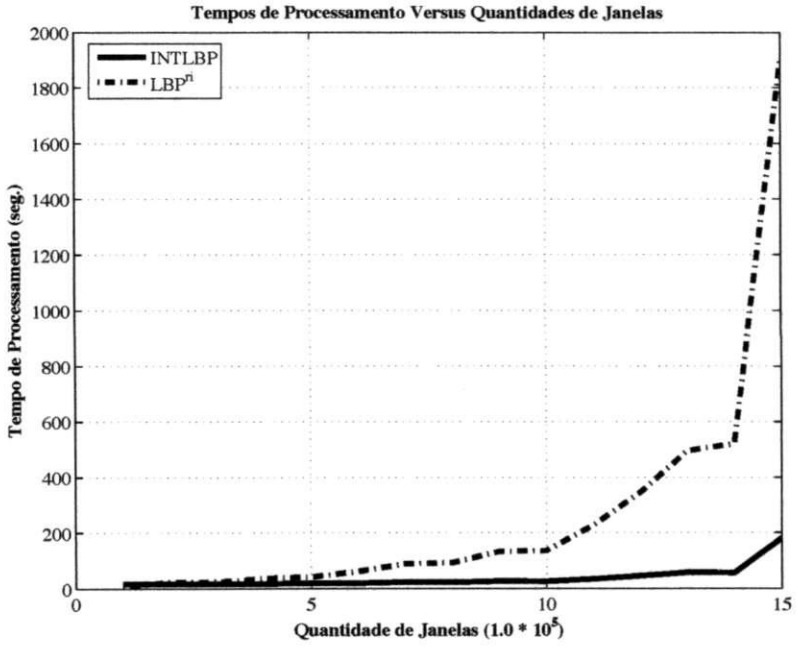
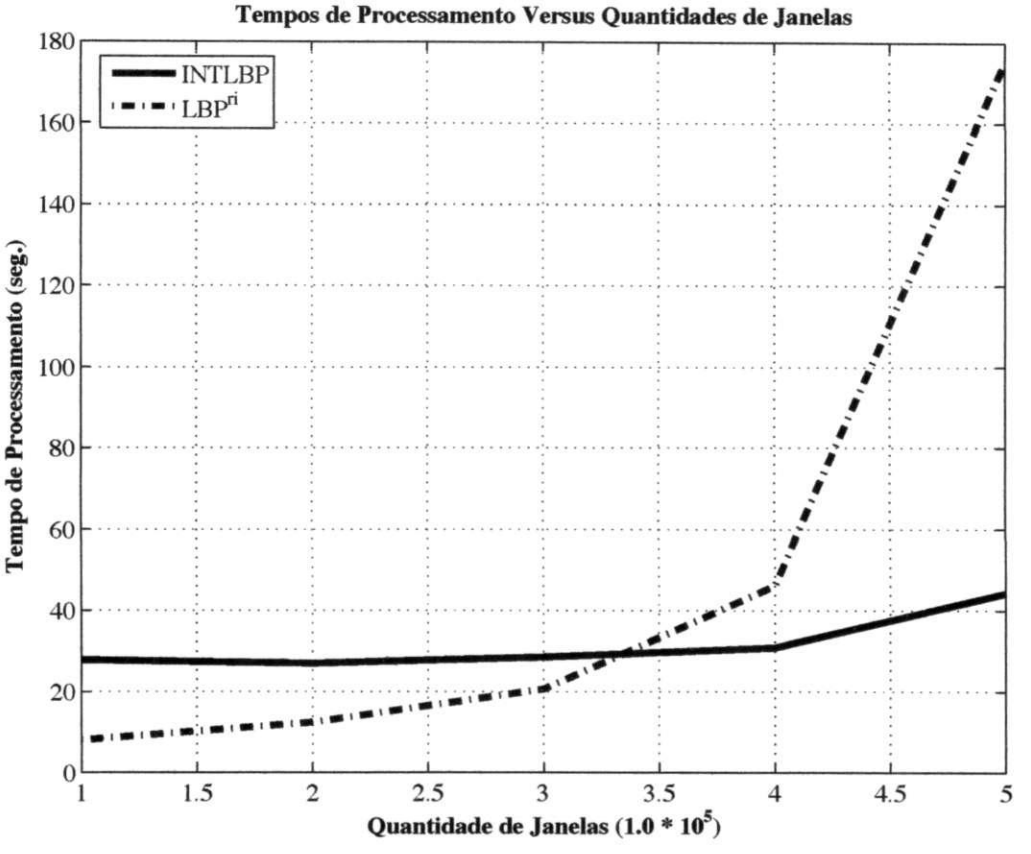


Figura A.16: Representação Gráfica dos dados da Tabela A.11. Dados obtidos dos experimentos utilizando pirâmide de imagem e tamanho de janela deslizante fixo.



Os resultados experimentais apresentados nesta subseção evidenciaram que a representação LBP Integral para extração de características de textura de imagens melhora a qualidade dos resultados em dois aspectos principais. Primeiro, a representação $INTLBP$ adiciona robustez a variações de escala nas imagens de teste. Além disto, a representação $INTLBP$ melhora a velocidade de processamento durante a busca de padrões em tarefas genéricas de detecção de objetos, tais como detecção de faces.

Tabela A.7: Comparação de resultados para 16 bins (INTLBP) e 6 bins LBP^{ri} quando as imagens foram divididas em 16 regiões.

Escala		INTLBP			LBP^{ri}		
TREINAMENTO	TESTE	VP	FP	FM	VP	FP	FM
232 × 280	29 × 35	0,00	99,69	0,0000	0,00	99,15	0,0000
	58 × 70	14,70	99,87	0,2560	11,36	98,71	0,2017
	116 × 140	94,29	99,98	0,9705	83,06	98,91	0,9021
	232 × 280	99,74	99,98	0,9986	97,32	99,56	0,9842
MÉDIA		52,18	99,88	0,5563	47,94	99,08	0,5220
DP		52,16	0,14	0,5056	49,36	0,36	0,4944
116 × 140	29 × 35	0,00	99,70	0,0000	0,18	99,79	0,0036
	58 × 70	81,38	99,90	0,8968	83,39	99,35	0,9062
	116 × 140	99,18	99,98	0,9958	96,02	99,18	0,9756
	232 × 280	92,93	99,98	0,9633	71,19	99,54	0,8295
MÉDIA		68,37	99,89	0,7140	62,69	99,47	0,6787
DP		46,17	0,13	0,4778	42,89	0,2619	0,4540
58 × 70	29 × 35	50,33	99,87	0,6690	27,89	99,81	0,4355
	58 × 70	99,41	99,98	0,9969	96,99	99,56	0,9825
	116 × 140	98,05	99,96	0,9900	26,45	99,42	0,4164
	232 × 280	83,85	99,94	0,9119	5,97	99,75	0,1124
MÉDIA		82,91	99,94	0,8920	39,33	99,64	0,4867
DP		22,83	0,05	0,1536	39,73	0,18	0,3622
29 × 35	29 × 35	97,96	99,96	0,9895	97,71	99,39	0,9854
	58 × 70	93,54	99,93	0,9663	0,97	99,78	0,0192
	116 × 140	29,10	99,95	0,4506	0,00	99,95	0,0000
	232 × 280	8,55	99,99	0,1575	0,00	99,99	0,0000
MÉDIA		57,29	99,96	0,6410	24,67	99,78	0,2512
DP		45,23	0,03	0,4071	48,69	0,27	0,4896

Tabela A.8: Comparação de resultados para 16 *bins* (INTLBP) e 6 *bins* (LBP^{ri}), quando a imagem foi dividida em 9 regiões.

Escala		$INTLBP$			LBP^{ri}		
TREINAMENTO	TESTE	VP	FP	FM	VP	FP	FM
232 × 280	29 × 35	0	99,68	0,0000	0,82	98,59	0,0160
	58 × 70	8,99	99,82	0,1647	21,68	97,98	0,3505
	116 × 140	85,74	99,94	0,9229	75,35	98,38	0,8516
	232 × 280	98,80	99,93	0,9936	96,58	99,27	0,9790
MÉDIA		48,38	99,84	0,5203	48,61	98,56	0,5493
DP		51,09	0,12	0,5110	44,82	0,54	0,4472
116 × 140	29 × 35	0,00	99,65	0,0000	1,18	99,52	0,0232
	58 × 70	79,38	99,87	0,8844	81,35	98,93	0,8919
	116 × 140	96,33	99,93	0,9810	96,02	98,69	0,9732
	232 × 280	85,97	99,89	0,9240	66,83	99,31	0,7979
MÉDIA		65,42	99,84	0,6974	61,35	99,11	0,6716
DP		44,17	0,13	0,4666	41,84	0,37	0,4381
58 × 70	29 × 35	50,16	99,83	0,6673	10,93	99,53	0,1962
	58 × 70	96,96	99,89	0,9840	96,43	99,17	0,9777
	116 × 140	95,08	99,68	0,9732	58,38	99,22	0,7336
	232 × 280	73,37	99,61	0,8445	24,97	99,61	0,3984
MÉDIA		78,89	99,75	0,8672	47,68	99,38	0,5765
DP		21,94	0,13	0,1476	38,11	0,22	0,3474
29 × 35	29 × 35	96,41	99,92	0,9813	92,25	98,42	0,9519
	58 × 70	80,39	99,85	0,8906	2,63	98,86	0,0507
	116 × 140	19,77	99,88	0,3298	0,00	99,61	0,0000
	232 × 280	0,59	99,96	0,0117	0,00	99,90	0,0000
MÉDIA		49,29	99,90	0,5533	23,72	99,20	0,2506
DP		46,30	0,05	0,4620	45,70	67,87	0,4681

Tabela A.9: Comparação de resultados para 16 *bins* (INTLBP) e 6 *bins* (LBP^{ri}), quando a imagem foi dividida em 4 regiões.

Escala		*INTLBP*			LBP^{ri}		
TREINAMENTO	TESTE	VP	FP	FM	VP	FP	FM
232 × 280	29 × 35	0,00	99,93	0,0000	4,29	97,20	0,0801
	58 × 70	59,64	99,82	0,7463	63,60	95,71	0,7576
	116 × 140	86,37	99,66	0,9252	74,46	96,06	0,8348
	232 × 280	95,53	99,71	0,9757	91,20	97,67	0,9425
MÉDIA		60,39	99,78	0,6618	58,39	96,66	0,6540
DP		43,04	0,12	0,4520	37,81	0,93	0,3894
116 × 140	29 × 35	16,65	99,73	0,2848	4,04	98,89	0,0768
	58 × 70	87,86	99,73	0,9341	80,61	97,46	0,8803
	116 × 140	92,57	99,71	0,9600	89,58	96,97	0,9302
	232 × 280	76,88	99,83	0,8685	62,53	98,29	0,7614
MÉDIA		68,49	99,75	0,7619	59,19	97,90	0,6622
DP		35,18	0,05	0,3204	38,45	0,85	0,3966
58 × 70	29 × 35	73,41	99,90	0,8462	17,18	99,01	0,2908
	58 × 70	92,43	99,72	0,9593	91,23	97,67	0,9427
	116 × 140	90,55	99,56	0,9482	56,72	97,36	0,7118
	232 × 280	69,39	99,65	0,8176	40,53	98,78	0,5719
MÉDIA		81,45	99,71	0,8928	51,42	98,21	0,6293
DP		11,73	0,14	0,0715	31,11	0,81	0,2726
29 × 35	29 × 35	93,85	99,55	0,9660	87,54	95,90	0,9136
	58 × 70	33,32	99,90	0,4995	1,48	97,83	0,0286
	116 × 140	1,61	99,96	0,0317	0,00	98,90	0,0000
	232 × 280	0,03	99,98	0,0006	0,00	99,61	0,0000
MÉDIA		32,20	99,85	0,3745	22,26	98,06	0,2355
DP		43,87	0,20	0,4556	37,70	1,61	0,4522

Tabela A.10: Tempos de processamento para 100 imagens de 320 × 240 pixels usando o método de janela deslizante com variações de escala.

JANELAS	$INTLBP$ (seg.)	LBP^{ri} (seg.)
46000	16,1	15,33
69400	16,51	23,36
71400	16,61	23,81
108400	17,39	36,28
123200	19,4	40,96
181800	19,29	60,82
258200	23,07	87,75
276000	22,18	91,46
392000	26,08	132,38
405800	25,54	135,08
677600	33,9	226,75
1052600	45,19	346,64
1482800	56,87	494,75
1562000	56,13	520,94
5776200	180,75	1901,83

Tabela A.11: Tempos de processamento para 100 imagens de resolução 320 × 240 pixels usando o método de deslocamento de janelas em pirâmides de imagens.

JANELAS	$INTLBP$ (seg.)	LBP^{ri} (seg.)
37800	27,88	8,00
58800	27,06	12,4
98400	28,62	20,63
220800	30,95	46,15
836000	44,28	175,30

A.5 Experimentos de Detecção de Faces

Nesta seção, são discutidos os experimentos realizados para verificar a precisão do detector de faces frontais aqui proposto. Os experimentos foram realizados utilizando imagens das bases BioID Face Database (JESORSKY; KIRCHBERG; FRISHHOLZ, 2001), Caltech Frontal Face Dataset (WEBER, 2010), YaleB (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001) e CMU (ROWLEY; BALUJA; KANADE, 1998b). As imagens frontais contidas na base YaleB são acompanhadas de arquivos contendo as coordenadas correspondentes às posições dos olhos e do centro da face.

A Base BioID e a base CMU são acompanhadas de arquivos contendo as coordenadas de pontos fiduciais da face, dentre eles os centros dos olhos. Por meio dessas marcações, foi possível realizar experimentos para testar, também, a robustez do classificador a oclusões dos olhos e da boca. Algumas imagens da base Caltech foram selecionadas para criar uma nova base em que as imagens tinham rotação em plano de 45 a 360 graus, variando a cada 45 graus. Além disso, foram executados sobre as mesmas bases de imagens os detectores de faces OpenCV (OPEN..., 2010) que é uma implementação do método de Viola e Jones (2004), de Rowley, Baluja e Kanade (1998a), IDIAP-HarrScann e IDIAP-MLP.

A abordagem utilizada para verificar as detecções de faces é baseada em duas outras abordagens propostas por Rodriguez et al. (2006) e Lienhart, Kuranov e Pisarevsky (2002). Os primeiros propõem uma abordagem para extração das coordenadas da face tendo como base o centro dos olhos. Os segundos consideram uma detecção verdadeira se o centro da face detectada possui no máximo uma distância do centro da face de *ground truth* correspondendo a 30% da largura da face de *ground truth* e a largura da face detectada deve estar entre 50% e 150% da largura da face de *ground truth*. A abordagem para a determinação da largura da face utiliza conhecimentos de antropometria descritos no livro de Farkas (1994). Os resultados apresentados por Farkas (1994) permitem gerar a Equação A.6.

$$LARGURA\_FACE = \frac{K‘}{2 \times K“} \times DIST\_OLHOS \tag{A.6}$$

Nesta equação, $K‘$ corresponde a 139,1 (que é a medida média de uma face humana em

milímetros), K" corresponde a 33,4 (que é a metade da distância média em milímetros entre as pupilas de pessoas) e $DIST\_OLHOS$ é a medida em pixels da distância entre os olhos da imagem de face. A partir desta equação e das observações de Lienhart, Kuranov e Pisarevsky (2002) foi criado um programa para verificar automaticamente as detecções de faces das bases mencionadas anteriormente.

A.5.1 Detecção de Faces em Imagens com Oclusão

Utilizando as coordenadas dos olhos e do centro das faces da base YaleB (GEORGHIADES; BELHUMEUR; KRIEGMAN, 2001), um subconjunto da base foi selecionado para gerar variações contendo oclusões no olho direito, olho esquerdo e boca, respectivamente. As oclusões foram geradas por meio da inserção na imagem de elipses preenchidas com preto. Para a oclusão dos olhos, os centros das elipses correspondem aos centros dos olhos. Para oclusão da boca, o centro da elipse corresponde à coordenada-x do centro da face e à coordenada-y do centro da face mais a metade da distância entre os olhos.

A elipse sobre o olho direito (utilizando como referencial o observador) possui uma rotação de 45° no sentido horário, a elipse sobre o olho esquerdo (também utilizando como referencial o observador) possui uma rotação de 45° no sentido anti-horário e a elipse sobre a boca é posicionada na horizontal correspondente à imagem. As figuras contidas na Tabela A.12 apresentam amostras de imagens da base YaleB com oclusão. Os experimentos de detecção de faces sobre imagens com oclusão foram realizados em 230 imagens.

As imagens utilizadas foram as imagens frontais dos 10 indivíduos da base YaleB com a fonte de iluminação variando de $-35°$ a $+35°$ em elevação e $-20°$ a $+20°$ em azimute, incluindo-se as imagens com iluminação ambiente. Os resultados das detecções de faces usando os classificadores OpenCV, Rowley-Et-Al, IDIAP-HaarScan, IDIAP-MLPScan e Proposto, podem ser vistos na Tabela A.13.

Pelos dados apresentados na Tabela A.13, observa-se que apenas o classificador proposto atingiu mais de 50% de acerto para todos os tipos de oclusão testados. Tanto para a oclusão no olho direito, quanto para a oclusão no olho esquerdo, todos os detectores obtiveram

Tabela A.12: Amostras de imagens com oclusão, nas quais foram detectadas faces pelo detector proposto.

Olho Direito	Olho Esquerdo	Boca
		

altas taxas de falsos positivos. No entanto, para as imagens com oclusão da boca, todos os detectores obtiveram resultados de precisão diferentes de zero. Apesar do detector de faces OpenCV ter obtido um resultado de 52,61% de acerto para imagens com oclusão da boca, os resultados obtidos pelo detector proposto ainda foram mais altos, 73,913%.

No geral, os piores detectores foram os IDIAP, pois não detectaram nenhuma face quando havia oclusão dos olhos e suas taxas de falsos positivos foram as mais elevadas dentre todos os detectores. Uma justificativa para as altas taxas de falsos positivos está relacionada à grande resolução das imagens testadas (640×480 pixels), essa resolução acarreta na necessidade de testar milhares de janelas durante a varredura. Por exemplo, para o classificador proposto, foram processadas 28910 janelas para cada imagem.

Tabela A.13: Tabela para comparação dos resultados de detecção de faces para imagens da base YaleB com oclusões. Os valores sob as colunas VP correspondem às taxas de verdadeiros positivos e os valores sob as colunas FP correspondem às quantidades de falsos positivos.

BASE	Olho Direito		Olho Esquerdo		Boca	
	VP (%)	FP	VP (%)	FP	VP (%)	FP
OPENCV	0,869	82	0,434	44	52,61	16
ROWLEY	0,435	2	0,000	1	3,478	0
IDIAP-HS	0,000	82	0,000	72	1,739	64
IDIAP-MLP	0,000	350	0,000	214	28,261	263
PROPOSTO	55,652	115	74,348	86	73,913	87

Então, para se obter a pior taxa real de falsos positivos no processo de classificação de imagens como faces ou não-faces, basta dividir o pior valor de falsos positivos da Tabela A.13 por 28910, o que resulta em $0,5/28910 = 0,0000173$; ou seja, para cada um milhão de janelas que não contém faces o classificador proposto classifica incorretamente 17,3 delas.

Logo, se esta quantidade de janelas for multiplicada por 4 e o pior resultado do detector de Rowley, Baluja e Kanade (1998a) presente na Tabela A.13 for dividido pelo valor resultante, a quantidade real de falsos positivos para o pior resultado de Rowley, Baluja e Kanade (1998a) na Tabela A.13 será $86,9/(4 \times 246766) = 0,00008804$. Ou seja, para cada um

milhão de janelas que não contém faces, o detector de Rowley, Baluja e Kanade (1998a) classifica incorretamente 88,04 janelas.

Os valores reportados por Viola e Jones (2004) para experimentos realizados sobre a base CMU Test Set com melhores resultados de verdadeiros positivos são 94,1% e 0,00005621 de falsos positivos, ou seja 56,21 falsas detecções para cada um milhão de testes. Portanto, para as imagens testadas neste experimento, o detector proposto obteve resultados compatíveis com os existentes na literatura da área. Na Tabela A.13, as três colunas principais Olho Dir, Olho Esq. e Boca, indicam os resultados para oclusão no olho direito, olho esquerdo e boca, respectivamente, enquanto as subcolunas VP e FP, indicam os resultados de verdadeiro positivos e falsos positivos. Os valores de verdadeiros positivos correspondem a quantidade de faces detectadas dividida pela quantidade de faces presentes. Os valores de falsos positivos correspondem a quantidade de detecções que não são faces dividida pela quantidade de imagens total.

Os detectores de faces analisados também foram executados sobre as bases de imagens originais. Pode-se observar na Tabela A.14 que os resultados obtidos são diferentes dos resultados apresentados no artigo de Rowley, Baluja e Kanade (1997). Essa discrepância de resultados deve-se ao que já foi observado anteriormente em relação a região de face delimitada pelos detectores.

Em algumas detecções, o detector de Rowley, Baluja e Kanade (1997) obteve como regiões de faces áreas que não continham todos os elementos principais das faces. Em alguns casos, a boca ficava de fora da marcação e, em outros, os olhos ou partes dos olhos ficavam de fora. Portanto, o detector de faces de Rowley, Baluja e Kanade (1997) não poderia ser aplicado satisfatoriamente como extrator de faces para um sistema de reconhecimento de faces. Na Figura A.17, apresentam-se excertos de imagens que possuem marcações com o problema mencionado.

Figura A.17: Exemplos de imagens de faces da base CMU, nas quais alguns dos resultados do detector de Rowley são insatisfatórios para o reconhecimento de faces.

O detector de faces IDIAP-MLP apresentou bons resultados para as bases BioID e YaleB, porém seus resultados para a base CMU não foram satisfatórios. Esses maus resultados devem estar relacionados à grande variabilidade de poses das faces nesta base e a documentação deste detector não menciona se o mesmo foi treinado para grandes variações de pose. A versão IDIAP-HS obteve péssimos resultados, o que se repetiu para todos os experimentos. O detector do OpenCV obteve ótimos resultados, com exceção para a base CMU. A mesma observação feita quanto ao treinamento dos detectores IDIAP serve para o detector OpenCV, o mesmo não foi treinado para grandes variações de pose.

Quanto aos resultados obtidos para o detector proposto, as taxas de verdadeiros positivos foram baixas, em relação àqueles obtidos pelos demais detectores para as bases BioID e CMU. No caso da base BioID, uma explicação para o resultado de 79,118% é a presença de faces com partes cortadas, faltando partes da cabeça. Como o detector proposto foi treinado com imagens de faces contendo toda a cabeça, este apresentou dificuldades em classificar imagens nas quais toda a cabeça não estivesse presente. Na Figura A.18 apresentam-se algumas imagens usadas para teste da base BioID em que toda a cabeça não está contida na imagem.

Os resultados de verdadeiros positivos do detector proposto, quando executado sobre a base CMU, também foram baixos. Neste caso, valem as observações relativas à grande variabilidade de posições da face e ao fato de o detector ter sido treinado apenas com imagens de faces frontais. Outro fator importante que dificultou a detecção de faces sobre a base CMU, foi a péssima qualidade de algumas imagens contidas nesta base. Para algumas imagens, é difícil até para um usuário humano identificar os elementos constituintes de algumas faces e.g., olhos, nariz e boca. Além disso, a menor resolução de face para a qual o detector proposto foi treinado é de 29×35 e a base CMU contém uma grande quantidade de imagens de resolução inferior a 29×35.

A.5.2 Detecção de Faces em Imagens Rotacionadas

Foram selecionadas da base Caltech 221 imagens que não fizeram parte do treinamento para serem rotacionadas no plano da imagem de 45 a 360 graus, variando a cada 45 graus. As

imagens selecionadas foram as de número 230 a 450. A quantidade total de imagens rotacionadas foi 1768, correspondendo a 221×8 (8 é a quantidade de variações de rotação para cada imagem). Os resultados da detecção de faces pelos detectores proposto e de Rowley, Baluja e Kanade (1997) sobre essas imagens rotacionadas são mostrados na Tabela A.15.

Embora Rowley, Baluja e Kanade (1997) apresentem vários experimentos demonstrando a robustez de seu detector a variações de rotação das faces, o detector de faces disponibilizado na *World Wide Web* pelos autores em *http://vasc.ri.cmu.edu/NNFaceDetector* não obteve resultados satisfatórios para todos os ângulos testados nos experimentos descritos nesta subseção. Conforme pode ser observado na Tabela A.15, o detector de Rowley, Baluja e Kanade (1997) não obteve resultados acima de 50% de acerto para as orientações de 45°, 135° e 270°. Tais resultados com taxas abaixo do esperado devem-se a alguns fatores que serão explicados a seguir.

Em primeiro lugar, muitos dos resultados apresentavam marcação na face que não continha todos os componentes importantes, tais como olhos e boca. Alguns resultados apresentam o lado superior do quadrado que demarca a face posicionado sobre a pupila do indivíduo. Outros resultados apresentam a boca cortada pelo lado inferior do quadrado e, muitas vezes, o quadrado contém apenas o nariz do alvo. Se os fatores evidenciados por Rodriguez et al. (2006) e por Sinha et al. (2006) forem levados em consideração, somente podem ser consideradas como faces as detecções que contiverem tanto as características internas (sobrancelhas, olhos, nariz e boca) quanto as características externas da face (cabeça, testa, orelhas e queixo). Portanto, muitas das detecções obtidas pelo detector de faces de Rowley, Baluja e Kanade (1997) foram consideradas incorretas por que não seriam adequadas para efetuar o reconhecimento das faces detectadas.

Outro fator que contribui para aumentar a complexidade do problema está relacionado ao posicionamento das marcações das faces. Como a figura geométrica escolhida para demarcar a face na imagem é retangular, quando a orientação é um múltiplo de 45° os olhos podem ficar de fora, por exemplo. Esse problema ocorre tanto para o detector de Rowley, Baluja e Kanade (1997) quanto para o detector aqui proposto e pode ser melhor entendido a partir da observação das imagens das figuras da Tabela A.16 e da Tabela A.17 as

quais contém as mesmas imagens com as marcações de obtidas pelo método proposto e pelo método de Rowley, Baluja e Kanade (1997), respectivamente.

Ao observar as imagens das Tabelas A.16 e A.17 na linha correspondente à rotação de 45° percebe-se que os resultados tanto para o detector proposto quanto para o detector comparado não são muito precisos na demarcação da face. Enquanto o detector proposto deixa praticamente toda a boca dos indivíduos de fora, o detector comparado (ROWLEY; BALUJA; KANADE, 1997) deixa parte dos olhos de fora. Além disso, observa-se que o detector proposto demarcou, para todas as imagens apresentadas na Tabela A.16, parte da testa e as sobrancelhas corretamente. Ao contrário, o detector comparado apresenta problemas quanto a deixar de fora as sobrancelhas dos indivíduos, o que está em desacordo com o que foi proposto por Sinha et al. (2006): as sobrancelhas são um fator importantíssimo para a detecção e o reconhecimento de faces humanas.

Figura A.18: Exemplos de imagens de faces da base BioID que não apresentam toda a cabeça.

Tabela A.14: Tabela para comparação dos resultados de detecção de faces para as bases de imagens sem alteração. Os valores sob as colunas VP correspondem às taxas de verdadeiros positivos e os valores sob as colunas FP correspondem às taxas de falsos positivos.

BASE	BioID		CMU		YaleB	
	VP (%)	FP	VP (%)	FP	VP (%)	FP
OPENCV	96,368	33	52,837	152	95,217	24
ROWLEY	83,268	15	72,603	36	91,304	0
IDIAP-HS	68,482	15	34,051	25	51,739	34
IDIAP-MLP	86,252	254	43,639	126	88,261	84
PROPOSTO	79,118	195	47,554	298	95,217	51

Tabela A.15: Tabela para comparação dos resultados de detecção de faces para imagens da base Caltech com rotações. A primeira linha apresenta as taxas de verdadeiros positivos obtidas pelo detector proposto por Rowley et al [RBK98b]. A segunda linha apresenta as taxas de verdadeiros positivos pelo detector proposto.

ANGULO / *DETECTOR*	45°	90°	135°	180°	225°	270°	315°	360°
ROWLEY	48,416	82,353	42,534	66,063	73,303	46,154	50,679	77,376
PROPOSTO	18,099	52,489	63,80	52,036	57,013	31,674	11,312	75,113

Tabela A.16: Amostras de imagens com rotação, nas quais foram detectadas faces pelo detector proposto.

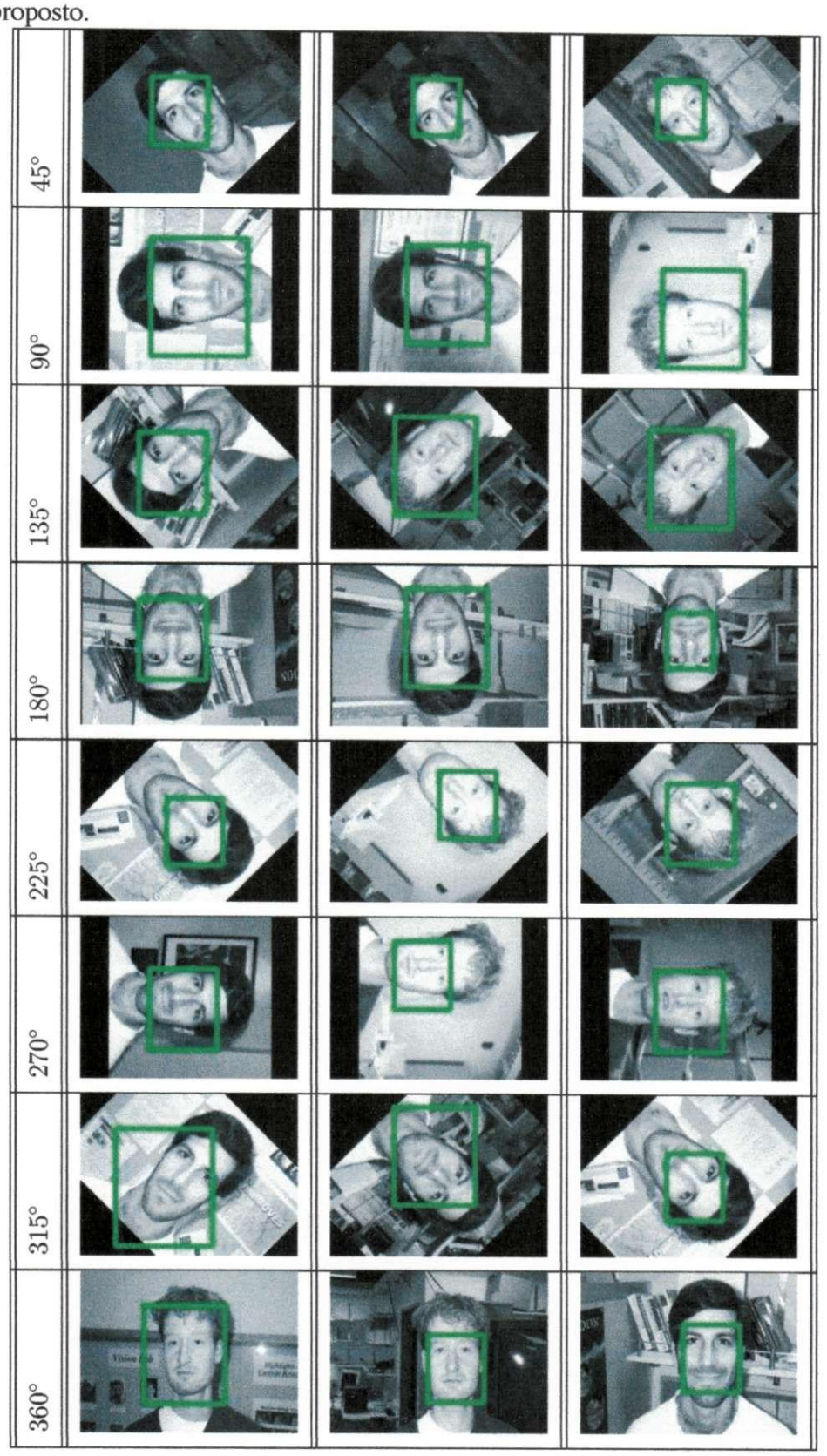
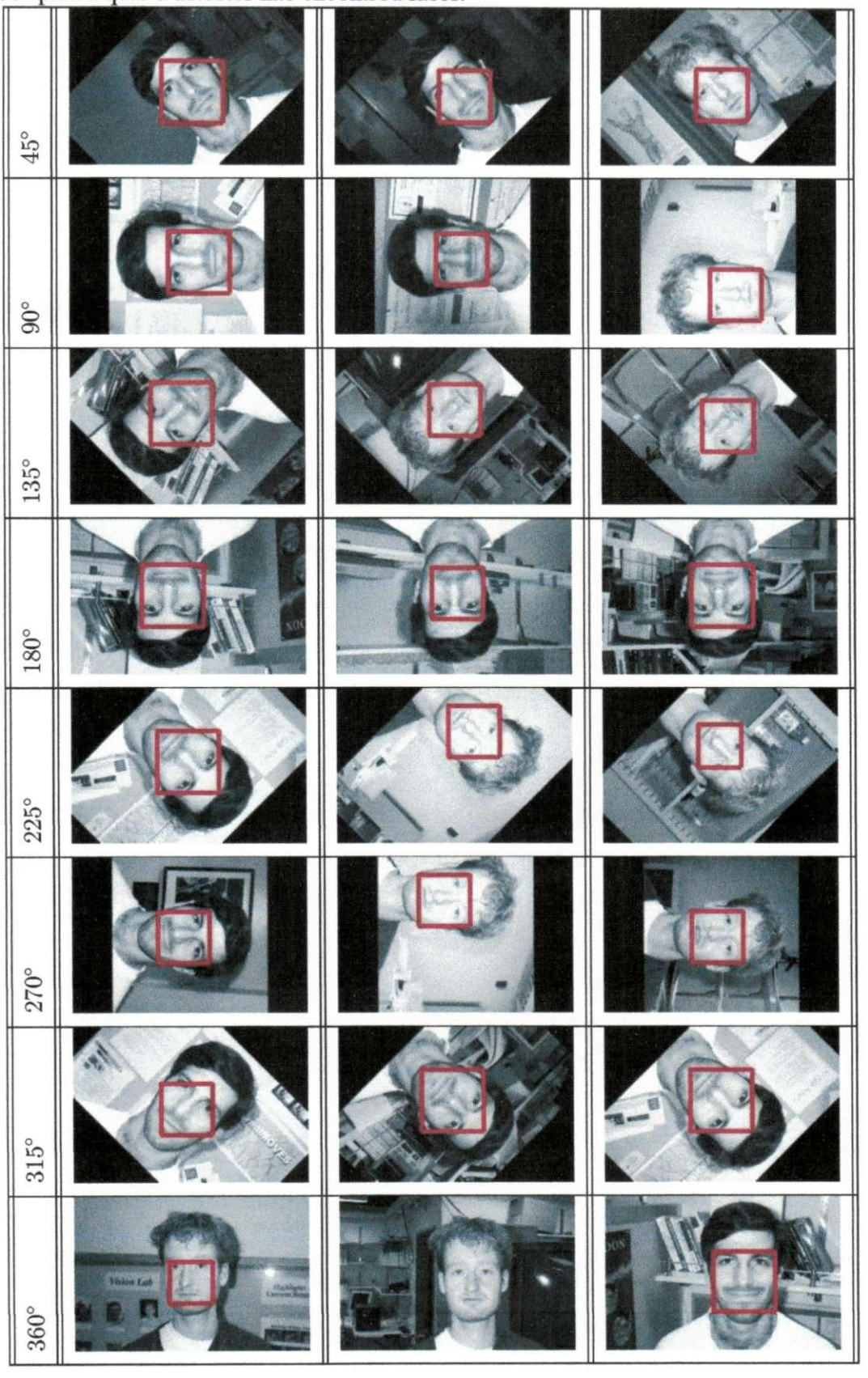


Tabela A.17: Amostras de imagens com rotação, nas quais foram detectadas faces pelo detector de Rowley et al.[RBK98b], com exceção da segunda imagem da linha correspondente a 360° para a qual o detector não encontrou faces.



A.6 Considerações Finais

Neste apêndice, a abordagem proposta, no documento de Proposta de Tese de Doutorado, para detecção de faces humanas em imagens digitais foi apresentada, a qual é composta por três classificadores obtidos pela combinação de modelos SVM. Cada modelo SVM foi gerado utilizando características diferentes dos outros, a saber: Razões Otimizadas de Faces, LBP Integral e Histogramas Integrais. Dentre os métodos de extração de características utilizados, os dois primeiros citados anteriormente constituem contribuições da abordagem proposta. Então, além de fornecer dois novos métodos para extração de características, essa proposta também apresenta um método para combiná-los, de modo a obter um detector de faces robusto à oclusão e rotação.

Foram apresentados também os experimentos realizados para verificar a precisão dos resultados obtidos pelo detector de faces proposto, bem como validá-lo. Os experimentos realizados tiveram como objetivo avaliar a robustez do detector de faces à oclusão e rotação, bem como comparar os resultados obtidos com resultados de detecção realizada por outros detectores disponíveis na *World Wide Web*.

Na primeira parte desta pesquisa de tese de doutorado, foram selecionados quatro tipos de características para treinamento de classificadores: razões otimizadas de faces (ROF), análise de componentes principais (PCA), padrões binários locais invariantes à rotação (LBPRI) e histogramas de níveis de cinza (HT). Os padrões LBPRI e HT foram implementados utilizando a abordagem integral. Uma descrição detalhada dessas características pode ser encontrada em seções anteriores deste capítulo.

Observando os resultados experimentais contidos neste apêndice, os quais foram obtidos por meio da utilização das características mencionadas e do treinamento de máquinas de vetores de suporte (SVM), é possível verificar que a abordagem inicialmente proposta nesta tese possui invariância à iluminação e oclusão, porém apresenta alguns problemas relacionados à invariância à rotação e à velocidade de processamento.

As características LBPRI e HT podem ser consideradas invariantes à rotação. Porém, na abordagem proposta inicialmente, essas características foram extraídas de modo local e não global. A extração local refere-se à divisão da imagem em regiões e à extração

das características de cada região individualmente, a fim de formar um vetor híbrido que reúne as características extraídas de todas as regiões. Essa extração local de características comprometeu a obtenção de invariância à rotação que a abordagem almeja alcançar.

Apesar de as principais características utilizadas (LBPRI e HT) terem sido implementadas por meio de uma abordagem integral semelhante à *integral image*, a velocidade de processamento de detecção de faces resultante ainda é baixa, em relação a outras abordagens existentes na literatura. Por exemplo, a abordagem de Huang et al. (2007) é capaz de processar 4 imagens de resolução 320×240 pixels por segundo. A abordagem inicialmente proposta nesta tese é capaz de processar uma imagem com tal resolução em 15 segundos. O principal motivo para a baixa velocidade de processamento é a utilização de SVM. Os modelos SVM gerados contêm milhares de vetores de suporte e, durante o deslizamento de janela sobre a imagem, cada vetor de características é multiplicado pelos vetores de suporte, acarretando baixa velocidade de processamento. Além disso, essa abordagem não utiliza o método de cascata de classificadores para descartar uma grande quantidade de janelas candidatas em estágios iniciais de classificação. Diante do exposto, surgiu a necessidade de implementar modificações na abordagem inicialmente proposta o que justifica a mudança de abordagem ocorrida nesta tese.