UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LOCALIZAÇÃO E EXTRAÇÃO AUTOMÁTICA DE TEXTOS EM IMAGENS COMPLEXAS

Dissertação submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.

Candidato: André Pires Nóbrega Tahim

Orientador: Prof. Rui Seara, Dr.

Florianópolis, 17 de março de 2010.

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LOCALIZAÇÃO E EXTRAÇÃO AUTOMÁTICA DE TEXTOS EM IMAGENS COMPLEXAS

Dissertação submetida
à Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Mestre em Engenharia Elétrica.

André Pires Nóbrega Tahim

Florianópolis, 17 de março de 2010.

LOCALIZAÇÃO E EXTRAÇÃO AUTOMÁTICA DE TEXTOS EM IMAGENS COMPLEXAS

André Pires Nóbrega Tahim

'Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em *Comunicações e Processamento de Sinais*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

Prof. Rui Seara, Dr.
Orientador

Prof. Roberto de Souza Salgado, Ph.D.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Rui Seara, Dr.
Presidente

Prof. Sidnei Noceti Filho, D.Sc.

Prof. Fernando Santana Pacheco, Dr.

Dedico esta dissertação aos meus pais, Marilda e Demétrio.

AGRADECIMENTOS

Aos meus pais, que não mediram esforços para me apoiar em todas as minhas escolhas. Realmente não sei expressar a minha gratidão diante de tanta dedicação, senão dizendo amo vocês.

Aos meus irmãos, que também fizeram papel de pai, pelo companheirismo e generosidade.

Aos amigos Tito, Cultura e Cabeça que me acompanharam na alegria e na tristeza, na saúde e na doença.

Ao amigo Ênio, pessoa espetacular que muito me ajudou durante toda essa caminhada.

Aos professores Rui Seara e Sidnei Noceti pela orientação, compreensão e apoio.

A Luciane Camilotti por todo suporte, carinho e companhia que tem me dedicado.

A galera do sindicato do LCMI, turma realmente especial que eu agradeço por fazer parte.

Ao prof. Fernando pelas valiosas críticas e orientações.

A Marcos Matsuo, visto que muito deste trabalho é resultado do seu empenho diário.

Aos amigos Walter e Elton, pela hospitalidade, favores, conselhos e agradáveis conversas.

A todos os amigos do Linse pela prazerosa convivência diária. Para registrar, um muito obrigado a Juan, Francisco, Calisto, Douglas, Rodrigo, Javier, Marcos Odebrecht, Simone, Pierre, Sandra, Vanessa, Ana, Letícia, Mariane, Pedro, Neco, Dudu, Daiana, Monique, Augusto, Feijão, Rafael, Jonathan, Bernardo, Aurêncio, Felipe, Orlando e Fernando.

Resumo da Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Engenharia Elétrica.

LOCALIZAÇÃO E EXTRAÇÃO AUTOMÁTICA DE TEXTOS EM IMAGENS COMPLEXAS

André Pires Nóbrega Tahim

Março/2010

Orientador: Prof. Rui Seara, Dr.
Área de concentração: Comunicações e Processamento de Sinais.
Palavras-chave: Extração da informação textual (TIE), extração de texto, localização de texto, OCR, SVM.

RESUMO: Existe uma busca crescente por técnicas de extração de informação textual (TIE) em imagens devido ao grande número de aplicações que elas possibilitam. Dentre as aplicações mais relevantes estão os sistemas de busca por imagens na Web, reconhecimento de placas veiculares e gerenciamento de base de dados de imagens. Entretanto, a maioria dos sistemas TIE é dedicada a aplicações em que se conhece o plano de fundo e/ou fontes, dimensões e orientação dos caracteres. Esta dissertação considera a localização e a extração de texto em imagens coloridas sem restrições. Em tal situação, desconhece-se o plano de fundo e o tipo de caractere presente na imagem (i.e., imagens complexas). A técnica de identificação textual proposta neste trabalho utiliza, seqüencialmente, as abordagens baseadas em região e textura. A primeira visa localizar as regiões da imagem candidatas a texto por meio da seleção de contornos de maior magnitude do gradiente de intensidade. A última verifica, dentre as regiões candidatas, aquelas que possuem texto embutido. Tal verificação é realizada por meio de 16 atributos (11 estruturais e 5 texturais) extraídos das regiões localizadas. Esses atributos alimentam um classificador *support vector machine* (SVM) que rotula as regiões localizadas como texto ou não-texto. As regiões classificadas como textuais são então submetidas à técnica proposta de extração de texto, a qual evita segmentações incorretas nas bordas dos caracteres devido aos artefatos incluídos durante o processo de compressão da imagem. A abordagem proposta é robusta na detecção de

textos oriundos de imagens complexas com vistas a diferentes orientações, dimensões e cores do texto, além de prover uma confiável binarização das regiões localizadas. O sistema TIE proposto apresenta resultados competitivos, tanto em precisão quanto em taxa de reconhecimento, quando comparados com outros sistemas da literatura técnica corrente.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

AUTOMATIC TEXT LOCATION AND EXTRACTION IN COMPLEX IMAGES

André Pires Nóbrega Tahim

March/2010

Advisor: Prof. Rui Seara, Dr.
Area of Concentrations: Communications and Signal Processing.
Keywords: Text information extraction (TIE), text extraction, text location, OCR, SVM.

ABSTRACT: There is an increasing search for techniques of text information extraction (TIE) from images due to the large number of applications that they make possible. Among the most relevant applications, search machine systems for Web images can be highlighted, as well as license plate recognition and management of image database. However, the majority of TIE systems are proposed for applications in which the background and/or font, dimension and orientation of characters are known. This dissertation is focused on text location and extraction from general purpose color images. In this situation, there is no information about the background and text present in the images (i.e. complex images). The proposed identification technique sequentially uses an approach based on region and texture. The former aims to locate the text candidate regions by selecting the image contours of higher gradient magnitude. The latter verifies, among the text candidate regions, those that have embedded text. Such verification is performed by using 16 attributes (11 structural and 5 textural) extracted from the text candidate regions. These attributes feed a support vector machine (SVM) classifier that labels the text candidate regions as text or non-text. The regions classified as text are then submitted to the text extraction algorithm, which prevents incorrect segmentations in the character boundaries caused by artifacts included during the image compression process. The proposed approach is robust in text detection from complex images with respect to different size, orientation and text color; moreover, it provides a reliable text binarization. The proposed TIE system exhibits competitive

results for both precision and recall rate, as compared with other approaches from the current technical literature.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

CBIR *content based image retrieval*

VIRS *visual information retrieval systems*

HTML *hypertext markup language*

TIE *text information extraction*

BB *bounding box*

OCR *optical character recognition*

ICDAR *International Conference on Document Analysis and Recognition*

SVM *support vector machine*

CC componentes conectados

FFT *fast Fourier transform*

PCA *principal component analysis*

KLT *Karhunen-Loève transform*

MSE *mean square error*

PSF *point spread function*

FS *feature selection*

RBF *radial basis function*

ANN *artificial neural network*

SRM *structural risk minimization*

C classe caractere

C+ classe mais de um caractere

NC classe não-caractere

GT *ground-truth*

CVPR *computer vision and pattern recognition*

CD *compact disc*

Lista de Símbolos

$\mathcal{T}$	Limiar que determina a escolha da cor semente no método K-means recorrente
$\mathrm{BB}_{\mathrm{altura}}$	Altura (eixo vertical) de uma *Bounding Box*
$\mathrm{BB}_{\mathrm{area}}$	Área de um *Bounding Box*
$\mathrm{BB}_{\mathrm{largura}}$	Largura (eixo horizontal) de uma *Bounding Box*
$\mathrm{I}_{\mathrm{altura}}$	Altura (eixo vertical) de uma imagem
$\mathrm{I}_{\mathrm{area}}$	Área de uma imagem
$\mathrm{I}_{\mathrm{largura}}$	Largura (eixo horizontal) de uma imagem
$\boldsymbol{G}_{\mathbf{x}}$	Imagem referente a derivada parcial em relação ao eixo x da imagem em níveis de cinza ($\boldsymbol{P}_{\mathbf{1}}$)
$\boldsymbol{G}_{\mathbf{y}}$	Imagem referente a derivada parcial em relação ao eixo x da imagem em níveis de cinza ($\boldsymbol{P}_{\mathbf{1}}$)
$\boldsymbol{H}_{\mathbf{h}}$	Filtro derivativo de Prewitt na direção horizontal
$\boldsymbol{H}_{\mathbf{v}}$	Filtro derivativo de Prewitt na direção vertical
$\boldsymbol{Y}$	Imagem de luminância
d_{JY}	Critério de dissimilaridade entre cores proposta por Jain e Yu
$d_{\text{city-block}}(\boldsymbol{x},\boldsymbol{y})$	Distância *city-block* entre duas cores: $\boldsymbol{x}$ e $\boldsymbol{y}$
$G_{\theta}(i,j)$	Direção do gradiente do pixel na posição (i,j)

$G_{\mathrm{mag}}(i,j)$	Magnitude do gradiente do pixel na posição (i,j)
K	Constante fixa que indica o peso relacionado a variância dos pixels sob a máscara na determinação do limiar local.
L	Limiar proposto por Bernsen que determina se o pixel deve receber um limiar local $T(x,y)$ ou se deve ser considerado plano de fundo
$m(x,y)$	Média de intensidade dos pixels sob a janela (ou máscara) W
M_{avg}	Razão entre a média da magnitude do gradiente dos pixels de contorno e o limiar L_{M}
P_{E}	Probabilidade de erro
$r(x,y)$	Dimensões laterais da janela adaptativa $W(x,y)$
$s(x,y)$	Desvio padrão da intensidade dos pixels sob a janela (ou máscara) W
$s_r(x,y)$	Desvio padrão da intensidade dos pixels sob a janela adaptativa $W(x,y) = r(x,y) \times r(x,y)$
T	Limiar global de binarização (Método de Otsu)
$T(x,y)$	Limiar local de binarização
T_σ	Limiar que determina quando a janela adaptativa $W(x,y)$ para a limiarização local possui o tamanho adequado
W	Janela (ou máscara) de tamanho fixo utilizada para a limiarização local.
$W(x,y)$	Janela adaptativa de limiarização local para cada pixel da imagem
Z_{max}	Pixel de maior valor de intensidade sob a janela W
Z_{min}	Pixel de menor valor de intensidade sob a janela W
$\boldsymbol{P_1}$	Imagem em níveis de cinza obtida da análise de componentes principais (primeiro componente principal) da imagem colorida
$\mathrm{A_{int}}$	Área de intersecção

A_{loc}	Área localizada
BB_{gt}	*Bounding Box ground-truth*
BB_{loc}	*Bounding Box* localizado
BB_{min}	Mínimo retângulo que contém os BB_{gt} e o BB_{loc}

Capítulo 1

Introdução

Atualmente, as informações veiculadas nos diferentes meios de comunicação se apresentam em variados formatos. Até recentemente, a difusão da maior parte das informações era puramente textual; porém, com o advento da Internet associada à crescente capacidade de compressão e transmissão de dados, disseminou-se uma grande quantidade de informações em forma de imagem e vídeo digital. Além disso, diversas organizações mantém enormes bases de dados na forma de imagens e vídeos com interesse em pesquisas médicas, entretenimento, comércio, segurança, entre outros. Contudo, os sistemas de busca de tais arquivos ainda são baseados em texto, conseqüentemente, para que a pesquisa e recuperação de um determinado arquivo seja eficiente, cada arquivo deve ser textualmente descrito por intervenção humana. Uma vez que a descrição manual de cada arquivo é morosa e subjetiva, torna-se necessário para gerenciamento, indexação e recuperação de tais arquivos, sistemas capazes de descrever o conteúdo de imagens automaticamente.

Como indexar automaticamente arquivos de imagens/vídeos baseado no seu conteúdo? Esse é um problema desafiador que vem ganhando interesse comercial e crescente atenção dos pesquisadores. Os sistemas que permitem a indexação automática baseado no conteúdo são chamados *content based image retrieval* (CBIR) ou *visual information retrieval systems* (VIRS) [21].

O conteúdo de uma imagem pode ser dividido em duas categorias principais: *conteúdo perceptual* e *conteúdo semântico*. O conteúdo perceptual compreende descritores de baixo nível, tais como cor, intensidade, forma, textura; enquanto o conteúdo semântico significa a identificação de objetos,

eventos e suas relações. O conteúdo perceptual de uma imagem é facilmente obtido; porém, diferentemente do conteúdo semântico, não revelam uma idéia precisa do significado da imagem.

Diversos estudos sobre a extração do conteúdo semântico em imagens utilizam características tais como face, veículos e ação humana. Um descritor de alto nível geralmente embutido em imagens é o *texto*. Técnicas que avaliam o conteúdo semântico por meio da extração e reconhecimento de textos superpostos em imagens ou que aparecem naturalmente na cena vêm suscitando o interesse de muitos pesquisadores pelos seguintes motivos:

- textos possuem informações semânticas úteis para descrever o conteúdo de uma imagem;
- a extração é relativamente mais simples do que outras características semânticas.

Técnicas de extração de textos em imagens continuam a ser amplamente pesquisadas, uma vez que textos embutidos em imagens possuem alguma relação com a ação ou o local que a imagem representa. Além disso, a localização, extração e reconhecimento automático de texto em imagens permite a automatização de diversas aplicações, tais como, o reconhecimento de endereços em envelopes [30], [68], a identificação de placas veiculares permitindo o monitoramento e fiscalização de possíveis infratores [2], [10], [17], busca de cenas específicas por meio das legendas e créditos em bancos de dados de vídeo [3], [11], [41], [56], pesquisas na Web [4], [60], dentre muitas outras.

Um exemplo bastante relevante é a extração textual visando a indexação de imagens presentes na Web. Os projetistas de sítios Web constantemente criam textos sob a forma de imagens, tais como botões e *banners*. Tal recurso é utilizado para suprir as limitações estilísticas da linguagem utilizada para a construção de páginas Web, denominada *hypertext markup language* (HTML). Um estudo realizado por Antonacopoulos et al. [4] mostra que cerca de 17% do número de palavras visíveis na Web está sob a forma de imagens, sendo que 76% dessas imagens não podem ser encontradas pelos sistemas de busca atuais [1], tornando todo esse conteúdo completamente inacessível aos usuários do sistema. Além disso, palavras embutidas em imagens impedem que sistemas de leitura para deficientes visuais consigam vocalizar essas informações.

Os sistemas capazes de extrair a informação textual de imagens são conhecidos como *text information extraction* (TIE). As abordagens para a localização e reconhecimento de texto utilizados pelos sistemas TIE estão intimamente relacionadas à maneira que o texto está inserido na imagem e ao plano de fundo ao qual o texto está sobreposto. A próxima seção classifica e

descreve os tipos de imagem de acordo com o plano de fundo e o tipo de texto que possuem.

1.1 Texto em Imagens

Os pesquisadores costumam definir três categorias de imagens quanto ao tipo de texto que elas possuem [32]: *imagem-documento*, *imagem-artificial (superposto)* e *imagem-cena*.

As *imagens-documento* são caracterizadas por possuírem texto sobre um plano de fundo homogêneo, contendo caracteres alinhados horizontalmente com poucas variações de fonte, cor[1] e possuindo um alto contraste com plano de fundo. As *imagens-documento* geralmente são digitalizadas por meio de *scanners*, em que se obtêm imagens de alta resolução sob condições de iluminação controlada. Tais características tornam a extração dos caracteres relativamente mais simples do que as outras duas categorias de imagens.

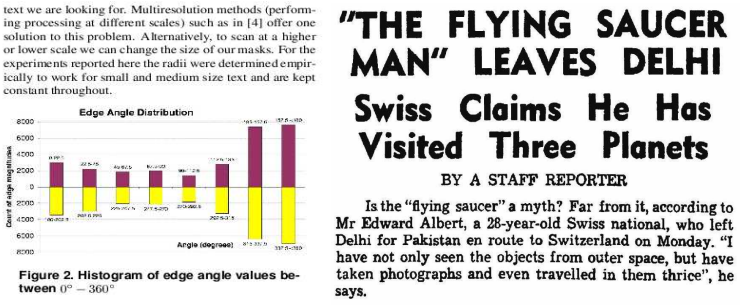
Uma vez que a maior parte da informação em uma *imagem-documento* é textual e apresenta-se sobre um plano de fundo homogêneo, realiza-se a identificação do *layout* da página (*page layout analysis*) [28] separando o texto dos gráficos e figuras. Após tal identificação, sistemas conhecidos como *optical character recognition* (OCR) convertem as regiões textuais da imagem em texto plano. Os sistemas de OCR foram inicialmente criados para a digitalização de documentos, visando o armazenamento, edição e busca automática. Atualmente, os sistemas de OCR possuem altas taxas de reconhecimento de caracteres (95% a 99%) para as imagens caracterizadas como documento. Exemplos de *imagens-documento* são ilustradas na Fig. 1.1.

As *imagens-artificiais* são caracterizadas por textos que são sobrepostos a uma determinada imagem, geralmente colorida, por meio de edição. Tais imagens, diferentemente das *imagens-documento*, podem apresentar caracteres sobrepostos a planos de fundo complexos, com uma grande diversidade de tamanhos, estilo, cor e orientação. Os *designers*, preocupados em chamar a atenção, freqüentemente buscam caracteres estilizados, apresentando uma grande diversidade de cores e orientações sobre um plano de fundo texturizado. Exemplos comuns de *imagens-artificiais* são *banners*, capas de *compact disc* (cd), livros e revistas, como ilustrado na Fig. 1.2.

É importante notar que os caracteres em imagens desta categoria geralmente possuem contraste com plano de fundo, uma vez que foram supostamente criados para serem lidos com facilidade.

As imagens em que os caracteres fazem naturalmente parte da cena são definidas como *imagens-cena*. A extração de texto de tais imagens apresenta

[1] As *imagens-documento* geralmente são monocromáticas ou em níveis de cinza, com caracteres em preto sobre um fundo branco.



(a) (b)

Fig. 1.1: Exemplos de *imagens-documento*: (a) Trecho de um artigo. (b) Notícia de jornal.

desafios ainda maiores do que as *imagens-documento* e *artificiais*. Uma vez que o texto é parte integrante da cena, este pode apresentar-se sob condições de iluminação não-uniforme, oclusão, possuir baixo contraste ao plano de fundo, diversas orientações e distorções de perspectiva. Além disso, as *imagens-cena* são afetadas por variações nos parâmetros das câmeras, tais como foco, iluminação, movimento, etc. Exemplos de *imagens-cena*[2] são apresentadas na Fig. 1.3.

Este trabalho visa a extração e reconhecimento de texto apenas em *imagens-artificiais* e *imagens-cena*. Esses dois conjuntos de imagem são englobados por um conjunto ainda mais amplo denominado *imagens complexas*, definido por Zhong et al. [70] como: *"Imagens em que os caracteres não podem ser segmentados do plano de fundo por simples limiarizações, e a cor, tamanho, fonte e orientação do texto são desconhecidos"*.

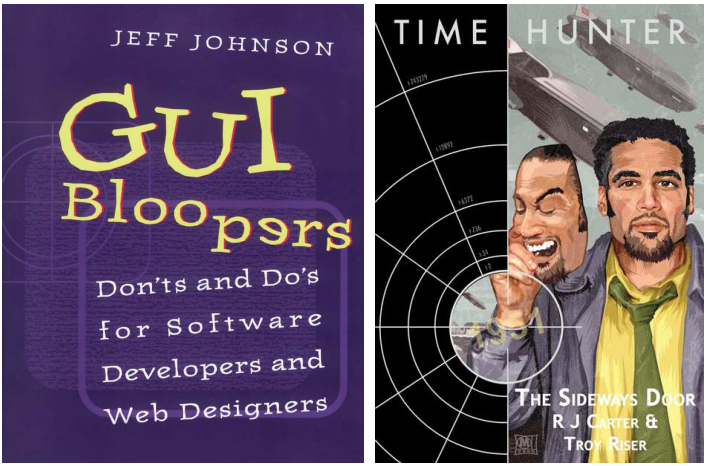
1.2 Características Textuais

Antes de iniciar a descrição das etapas realizadas pelos sistemas TIE, listam-se nesta seção algumas características textuais comumente utilizadas pelos algoritmos de TIE para identificar as regiões da imagem que possuem texto.

1. **Geometria**

   - *Dimensões*. Embora as dimensões dos caracteres possam variar bastante, algumas inferências podem ser feitas de acordo com a

[2]Imagens obtidas da *International Conference on Document Analysis and Recognition* (ICDAR) 2003.



(a) (b)

Fig. 1.2: Exemplos de *imagens-artificiais*: (a) imagem colorida com caracteres não-alinhados possuindo diferentes orientações e fontes estilizadas. (b) caracteres sobre um plano de fundo colorido (canto inferior direito).

aplicação. Um exemplo é a identificação das placas veiculares, as quais possuem uma certa razão de aspecto. Uma filtragem por meio de tal característica facilita a seleção das regiões candidatas a conter a placa.

- *Alinhamento*. Os caracteres geralmente aparecem agrupados e alinhados em uma determinada direção, geralmente horizontal. Tal característica não se aplica às *imagens-cena*, uma vez que essas podem apresentar textos alinhados em qualquer direção e/ou apresentar distorções de perspectiva.
- *Coesão espacial*. Texto é naturalmente um agregado de caracteres que apresentam-se em uma determinada orientação, exibindo alturas e espaçamento similares (veja Fig. 1.4). Tal característica é comumente denominada coesão espacial.

2. **Contraste (cor e intensidade)**

- A cor e a intensidade são características amplamente utilizadas na localização de texto em diversos sistemas TIE. Uma vez que os caracteres devem ser lidos, esses devem possuir um adequado

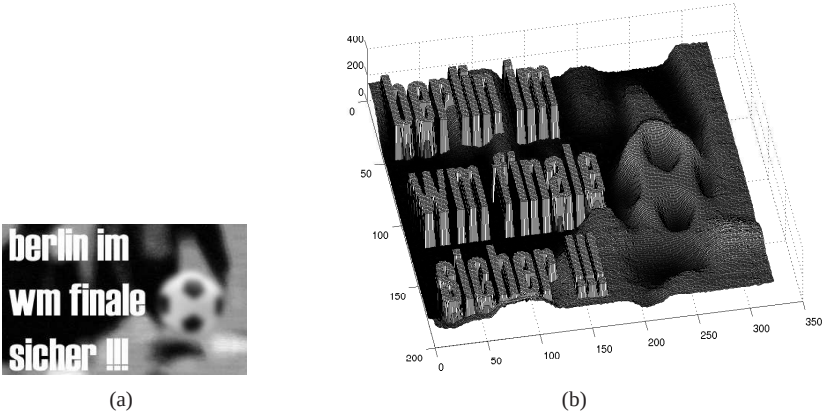
(a) (b)

Fig. 1.3: Exemplos de *imagens-cena*: (a) imagem contendo variações de iluminação sobre os caracteres devido às distorções geométricas em um plano de fundo texturizado. (b) imagem contendo caracteres com distorção de perspectiva e iluminação não-uniforme (placa inferior).

Fig. 1.4: Similaridades geométricas dos caracteres: alinhamento (marcador em azul), altura (marcadores em vermelho) e espaçamento (marcadores em verde).

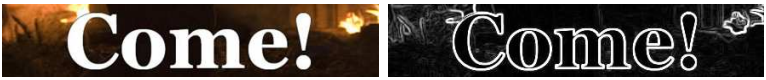
contraste de crominância (imagem colorida) ou intensidade (imagem em níveis de cinza) com o plano de fundo. O contraste dos caracteres trata-se de uma variação abrupta de intensidade na região que define os limites entre o plano de fundo e o corpo dos caracteres, como ilustrado na Fig. 1.5.

- O corpo (traço) dos caracteres geralmente possui uma cor (imagem colorida) *perceptualmente* uniforme em toda a sua extensão [veja os caracteres em branco na Fig. 1.5(a)]. No entanto, os caracteres geralmente contém de dezenas a milhares de cores na extensão do seu corpo, tornando necessário para os sistemas de reconhecimento textual utilizar espaços de cor em conformidade com o sistema visual humano, em que cores distantes em tal espaço sejam perceptualmente diferentes para um observador humano. Utilizando adequados espaço de cor e métrica de similaridade, é possível assumir um grau de uniformidade de cor no corpo dos caracteres, permitindo assim agregar pixels com cores similares em regiões e, posteriormente, identificar quais regiões representam os caracteres.



(a) (b)

Fig. 1.5: Contraste entre caracteres e o plano de fundo. (a) imagem original em níveis de cinza; (b) imagem em níveis de cinza apresentada em 3 dimensões.



(a) (b)

Fig. 1.6: Imagem original e sua correspondente imagem magnitude do gradiente. (a) Imagem original. (b) Imagem de magnitude do gradiente (as regiões em branco mais intenso possuem maior magnitude do gradiente).

3. **Bordas**

- A variação de cor (imagem colorida) ou intensidade (imagem em níveis de cinza) nas bordas dos símbolos textuais são geralmente mais evidentes do que em outros objetos da imagem [43]. Tal variação geralmente é quantificada por meio do operador gradiente, cujo valor da magnitude representa uma importante característica de seleção entre caracteres, objetos e plano de fundo [41]. Mesmo que o plano de fundo apresente regiões com variações de cor (imagem colorida) ou intensidade (imagem em níveis de cinza), geralmente o valor da magnitude do gradiente em tais regiões é inferior às regiões das bordas dos caracteres, como ilustrado na Fig. 1.6.
- Caracteres legíveis geralmente apresentam uma variação abrupta de cor (ou intensidade em imagens em níveis de cinza) no limite entre os pixels de contorno (bordas) dos caracteres e o plano de

(a) (b)

Fig. 1.7: Caracteres policromáticos e as bordas correspondentes. (a) Imagem com variação de cor no corpo do caractere (degradê entre o branco e o amarelo). (b) Bordas extraídas da imagem.

fundo. É importante notar que não importa o número de cores presente no corpo dos caracteres, a legibilidade de um caractere é dependente apenas do contraste do plano de fundo com os pixels de contorno (veja Fig. 1.7). Os contornos textuais podem caracterizar completamente o caractere. Além disso, tais contornos podem ser completamente extraídos caso possuam contraste com o plano de fundo, como mostrado na Fig. 1.7(b).

- Caracteres podem ser vistos como um aglomerado de bordas com coesão espacial. Sendo assim, diversos pesquisadores determinam a *densidade de bordas* em toda a imagem por ser improvável que regiões de baixa densidade possuam texto [8], [11].

4. **Textura**

   - Observando uma tira de jornal de humor a alguns metros de distância, podemos dizer rapidamente onde o texto está presente sem verdadeiramente identificar os caracteres individualmente. Isso indica que o texto possui uma *regularidade* capaz de diferenciá-lo de outros objetos e do plano de fundo (veja Fig. 1.8). A *regularidade* dos textos, dada pela similaridade da dimensão, espessura do traço, orientação e distância entre caracteres, faz com que o texto possua uma textura com componentes freqüenciais distintos dos outros objetos da imagem [66].

1.3 Descrição do Problema

Esta seção visa descrever o problema relacionado a extração de texto em imagens complexas, bem como a abordagem utilizada por grande parte dos sistemas TIE da literatura para a solução de tal problema.

(a) (b)

Fig. 1.8: Regularidade dos caracteres (textura).

Os sistemas de OCR atuais possuem uma alta taxa de reconhecimento para *imagens-documento*, porém tais sistemas são incapazes de reconhecer a informação textual em imagens complexas [70]. Um sistema de OCR convencional aplica binarizações locais ou globais à imagem em níveis de cinza de alta resolução (100-300 dpi) [20], visando separar os caracteres do plano de fundo. No entanto, imagens complexas em sua maioria são coloridas, de baixa resolução e apresentam artefatos incluídos durante o processo de compressão; tais características impossibilitam a separação dos caracteres do plano de fundo por simples binarizações (local ou global).

Chen [11] realizou um experimento visando estabelecer a taxa de reconhecimento de caracteres por meio da alimentação direta de imagens complexas em um sistema de OCR convencional. O resultado obtido é que nenhum dos caracteres foi reconhecido, demonstrando a incapacidade dos sistemas de OCR convencionais em localizar e reconhecer a informação textual presente em tais tipos de imagens.

O primeiro sistema de OCR criado data da década de 50. Desde então, diversas pesquisas vêm sendo realizadas tornando os sistemas de OCR uma das mais bem sucedidas tecnologias no campo de reconhecimento de padrões e inteligência artificial [11]. Existem diversos sistemas de OCR com taxas de reconhecimento que variam entre 95% e 99% para *imagens-documento* [61]. Em decorrência disso, a solução apresentada pela maioria dos sistemas TIE

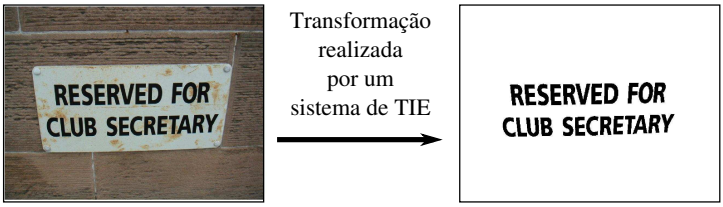


Fig. 1.9: Transformação realizada por um sistema TIE com o objetivo de separar os caracteres do plano de fundo complexo.

para *imagens complexas* é transformá-las em imagens com as características de *imagens-documento*, acoplando ao final do processo um sistema de OCR para o reconhecimento dos caracteres.

Desta forma, o objetivo dos sistemas TIE é preencher a lacuna existente entre a *imagem complexa* e a *imagem-documento*, visto que na última os caracteres podem ser reconhecidos por um sistema de OCR. A Fig. 1.9 apresenta a transformação de uma imagem complexa em uma imagem binária adequada ao reconhecimento dos caracteres por sistemas de OCR convencionais.

1.4 Extração da Informação Textual - TIE

Sistemas de extração da informação textual (TIE) em imagens complexas recebem como entrada uma imagem ou seqüência de imagens possuindo texto (em que tais imagens podem ser coloridas ou em níveis de cinza), retornando texto plano[3] como saída [32].

Um sistema TIE geralmente é dividido em quatro subsistemas: (i) localização; (ii) verificação; (iii) extração; (iv) OCR. Os três primeiros são responsáveis pela adaptação da imagem complexa a ser reconhecida pelo sistema de OCR, como ilustrado na Fig. 1.9. A arquitetura completa de um sistema TIE está ilustrada no diagrama de blocos da Fig. 1.10.

As subseções seguintes descrevem o propósito de cada etapa de um sistema TIE (veja Fig. 1.10).

1.4.1 Localização

A etapa de localização tem como propósito responder a pergunta: “onde está o texto na imagem?”. Segundo Zhong et al. [70], é impraticável reco-

[3]Informação descriptografada, facilmente manipulada computacionalmente. Um exemplo é texto no formato ASCII.

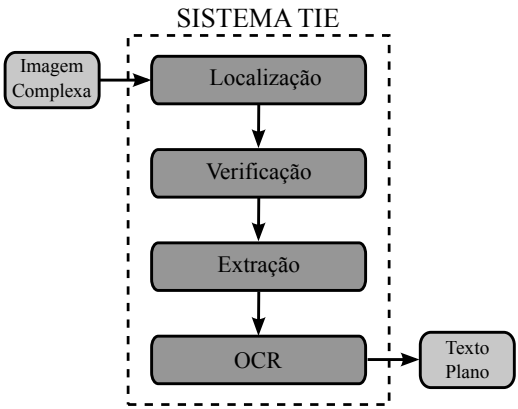


Fig. 1.10: Arquitetura de um sistema TIE.

nhecer um texto em uma imagem complexa sem previamente localizá-lo. Tal afirmação baseia-se no fato das imagens complexas possuírem texto disperso, com fontes de variados tamanhos, cores e orientações em um plano de fundo texturizado. Tais características tornam ineficiente a tentativa de separação dos caracteres do plano de fundo, transformando-a em uma imagem binária, sem antes localizá-los.

O sucesso da separação dos caracteres do plano de fundo em imagens complexas, como apresentada na Fig. 1.9, é dependente da localização prévia das regiões candidatas a texto. Tal processo restringe a imagem a pequenas regiões a serem exploradas, reduzindo os problemas relacionados à grande variedade de textura, objetos e cores presentes em uma imagem complexa.

A região identificada como candidata a texto é delimitada por uma caixa limítrofe (retangular), comumente conhecida como *bounding box* (BB). Os algoritmos de localização recebem uma imagem complexa como entrada e, dependendo do algoritmo de localização utilizado, podem retornar áreas delimitadas por BBs contendo um conjunto de palavras (linhas de texto), palavras ou caracteres. Para melhor exemplificar o processo de localização, a Fig. 1.11(a), denominada neste trabalho de *imagem-exemplo*, foi submetida a um algoritmo de localização que delimita os caracteres individualmente, como apresentado na Fig. 1.11(b).

1.4.2 Verificação

Uma vez que os algoritmos de localização geralmente utilizam poucas características textuais para selecionar as regiões da imagem candidatas a

(a) (b)

(c) (d)

Fig. 1.11: Seqüência de resultados das 3 primeiras etapas de um sistema TIE. (a) Imagem original. (b) Resultado da etapa de localização (BBs azuis representam os caracteres corretamente localizados, enquanto os BBs amarelos representam os falsos positivos). (c) Resultado após a etapa de verificação. (d) Imagem binária após a etapa de extração.

texto, após a etapa de localização, diversas regiões que não representam caracteres estão delimitadas por BBs (falsos positivos) [Fig. 1.11(b)]. A etapa de verificação tem como objetivo fazer uma seleção refinada dos BBs obtidos na etapa de localização, com o propósito de responder a seguinte pergunta: “quais áreas selecionadas possuem realmente caracteres?”.

Esta pergunta somente pode ser respondida se possuirmos características, extraídas de cada região delimitada por um BB, capazes de diferenciar entre regiões textuais e não-textuais. Dessa forma, a etapa de verificação extrai diversos atributos de cada BB obtido na etapa de localização e, mediante a avaliação desses atributos, o classifica como textual ou não-textual.

A etapa de verificação possui como entrada o conjunto de BBs advindos da etapa de localização e retorna o conjunto de BBs classificados como textuais. Assim, a etapa de verificação pode ser vista como um processo de

filtragem dos BBs localizados mediante a avaliação de características textuais capazes de diferenciar caracteres de outros objetos. A Fig. 1.11(c) apresenta o resultado da etapa de verificação, em que os BBs remanescentes foram os classificados como textuais.

Muitos autores consideram a verificação como parte integrante da etapa de localização, visto que as regiões localizadas e consideradas textuais só são conhecidas após a etapa de verificação. No entanto, neste trabalho considera-se a subdivisão em duas etapas (localização e verificação) visando facilitar a compreensão e destinar uma maior atenção aos algoritmos e atributos textuais envolvidos na etapa de verificação.

1.4.3 Extração

Após as etapas de localização e verificação, as regiões consideradas textuais estão delimitadas por BBs [Fig. 1.11(c)]. Dessa forma, tudo que está externo a tais áreas é considerado plano de fundo. Contudo, as áreas delimitadas por BBs ainda possuem pixels que representam o plano de fundo e pixels que representam caracteres.

A extração visa responder a seguinte pergunta: "quais pixels pertencem aos caracteres e quais pertencem ao plano de fundo?". A etapa de extração é fundamental para o reconhecimento dos caracteres pelo sistema de OCR devido à exigência deste último em obter como entrada caracteres com alto contraste sobre um plano de fundo uniforme e com o traço bem definido e sem rupturas. A etapa de extração possui como entrada um conjunto de BBs que delimitam as possíveis regiões textuais e retorna para cada BB uma imagem binária em que os pixels representando os caracteres possuem o valor binário '0' (preto) e os pixels do plano de fundo o valor binário '1' (branco), como ilustrado na Fig. 1.11(d).

1.4.4 OCR

Após as três primeiras etapas do sistema TIE, a imagem apresenta-se na forma binária com os caracteres sobre um plano de fundo homogêneo [Fig. 1.11(d)]. A última etapa geralmente é constituída de um OCR convencional e tem como objetivo responder a pergunta: "o que está escrito na imagem?".

O sistema de OCR possui como entrada uma imagem binária ou em níveis de cinza da região da palavra[4], retornando os caracteres em texto plano

[4]Quando o algoritmo de localização identifica os caracteres individualmente, faz-se necessário agrupar os caracteres para encontrar a região das palavras, visto que os sistemas de OCR utilizam dicionários para melhorar a taxa de reconhecimento.

Fig. 1.12: Transformação da imagem binária contendo caracteres em texto plano (ASCII) (o lado esquerdo apresenta as regiões após a etapa de extração, enquanto o lado direito representa a saída do OCR).

(geralmente ASCII ou HTML), como mostrado na Fig. 1.12.

No presente trabalho foi utilizado o OCR Tesseract (código aberto) [34] atualmente mantido pela empresa Google Inc.

1.5 Objetivos e Contribuições do Trabalho

Este trabalho propõe um sistema TIE genérico para imagens coloridas, estáticas[5] e complexas. Uma vez que este trabalho propõe um sistema TIE de propósito geral, diversos problemas surgem em praticamente todas as etapas descritas anteriormente. Tais problemas são devido ao grande número de fatores que variam em imagens sem qualquer tipo de restrição, tais como dimensão, fonte e orientação dos caracteres, textura do plano de fundo, dimensões da imagem, dentre outros.

Esta seção apresenta os diversos problemas relacionados ao reconhecimento textual em imagens complexas como também as contribuições do trabalho que visam solucioná-los.

1.5.1 Etapa de Localização

A construção de um sistema TIE de propósito geral é desafiadora devido ao completo desconhecimento da imagem de entrada. Como conseqüência, o sistema deve localizar as regiões textuais por meio de características menos dependentes dos parâmetros variantes em uma imagem genérica. Além disso, para contemplar a diversidade de imagens, o algoritmo de localização deve possuir o menor número de heurísticas possíveis.

O método de localização proposto explora a característica de contraste existente entre os pixels de contorno e o plano de fundo nos caracteres *legíveis*. Tal característica é invariante ao tipo de fonte e dimensão dos caracteres, permitindo que o método de localização obtenha êxito independente do

[5] Não inclui o reconhecimento de texto em vídeo.

tipo de caractere presente na imagem. Dessa forma, o método de localização baseia-se na obtenção dos contornos da imagem, selecionando aqueles que possuem características textuais, visto que os caracteres são completamente descritos pelos seus contornos.

O método de localização proposto é baseado no trabalho de Liu et al. [43], no qual os autores consideram que, para facilitar a legibilidade, os pixels do contorno dos caracteres possuem altos valores de magnitude do gradiente, quando comparado a outros objetos da imagem. No entanto, diferentemente do trabalho de Liu et al. [43], a detecção de bordas do método proposto tira proveito da análise de componentes principais na obtenção da imagem em níveis de cinza de maior variância e aplica sobre tal imagem um filtro derivativo. Tal método possui complexidade computacional inferior[6] ao método de Liu et al. [43] no processo de detecção de bordas, uma vez que este último utiliza a detecção por campos vetoriais [38] aplicado diretamente à imagem colorida. No entanto, segundo Lee e Park [37], tal detecção de bordas, em alguns casos, é menos sensível ao ruído.

O Capítulo 2 demonstra a relativa independência do método de localização proposto neste trabalho quanto ao tipo de fonte, dimensões, cor e orientação dos caracteres, além de ser indiferente às dimensões da imagem de entrada. O método identifica em uma única varredura as possíveis regiões textuais, eliminando o processo de busca exaustiva mediante técnicas de multiresolução. Além disso, possui a vantagem de identificar os caracteres individualmente, eliminando heurísticas referentes à localização baseada na coesão espacial dos caracteres, tais como alinhamento, espaçamento, dimensões relativas e orientação dos caracteres. Discute-se ainda neste capítulo a ineficiência do método proposto na localização de caracteres fora de foco ou sobrepostos a um plano de fundo povoado por bordas evidentes, cuja magnitude do gradiente é equiparável ao dos caracteres da imagem.

1.5.2 Etapa de Verificação

O principal objetivo da etapa de verificação é aumentar a robustez na determinação das regiões candidatas a texto identificadas na etapa de localização. Uma vez que o método proposto de localização identifica as regiões candidatas a texto usando um método baseado em bordas, o método de verificação proposto utiliza métodos texturais e estruturais para a certificação de tais regiões como textuais. A utilização de diferentes abordagens promove robustez à identificação de regiões textuais, porém não inserindo grande aumento de complexidade computacional, visto que os métodos computacionalmente custosos são apenas aplicados às áreas previamente delimitadas na etapa de

[6] Devido ao menor número de operações efetuadas.

localização.

A etapa de verificação proposta neste trabalho desenvolveu-se mediante a busca de atributos, extraídos da imagem, capazes de classificar as regiões localizadas em textuais e não-textuais. O conjunto de atributos com o melhor desempenho classificatório foi então utilizado para treinar uma máquina de aprendizado denominada *support vector machine* (SVM), criando-se dessa maneira um classificador binário, texto e não-texto. Pode-se enumerar as seguintes contribuições providas por este trabalho para a etapa de verificação:

1. A realização de uma interface gráfica capaz de gerar um banco de dados de regiões de imagens divididas em 3 classes: *caractere*, *mais de um caractere* e *não-caractere*.

   A interface delimita automaticamente um conjunto de regiões candidatas a texto em uma imagem, permitindo que o usuário atribua a cada uma delas uma classe (*caractere*, *mais de um caractere* ou *não-caractere*), informações dos caracteres (qualidade, orientação e o símbolo que o representa) e não-caracteres (aparência), além de cadastrar automaticamente informações de posição e dimensão de cada região. Tais informações vão servir para avaliar quais atributos possuem poder discriminativo entre regiões textuais e não-textuais. Uma outra utilização de tal banco, ainda mais importante, é a avaliação de desempenho de sistemas TIE, visto que conhecemos, dentre as imagens cadastradas, as regiões que possuem caracteres e quais são eles.

2. A concepção de uma interface gráfica de avaliação da capacidade de discriminação dos atributos entre regiões textuais e não-textuais. Tal interface acessa o banco de dados de regiões de imagens e possui uma série de campos que permitem ao usuário selecionar regiões da imagem que apresentam certas características. Dessa forma, provê ao usuário uma maneira de selecionar os atributos que possuem o melhor desempenho de classificação (texto ou não-texto) para uma dada aplicação.

3. A proposta e a avaliação de 16 atributos, 11 estruturais e 5 texturais. A capacidade classificatória individual de cada atributo foi obtida utilizando a interface de avaliação de atributos descrita anteriormente. Tal avaliação é realizada sobrepondo os histogramas de cada classe (*caractere*, *mais de um caractere* e *não-caractere*) para um determinado atributo. Caso a maior parte das ocorrências da classe *caractere* e *mais de um caractere* se apresentem em uma região distinta da classe *não-caractere* no histograma, o atributo possui um limiar potencialmente capaz de diferenciar entre regiões textuais e não-textuais.

4. A proposta de um seletor de atributos que obtém, dentre todos os atributos, um subconjunto de atributos capaz de melhorar a precisão de predição de um classificador específico.

5. A implementação de uma máquina de aprendizado SVM, para melhorar a precisão em classificar as regiões em textuais e não-textuais, utilizando o melhor subconjunto de atributos obtido.

1.5.3 Etapa de Extração

A etapa de extração possui como principal objetivo a separação dos caracteres do plano de fundo em uma imagem binária. Uma vez que caracteres de dimensões reduzidas e contendo artefatos sempre foi um desafio durante a extração, o método de extração usado neste trabalho propõe um algoritmo iterativo, baseado na clusterização de cores, com o objetivo de melhorar a segmentação para caracteres de dimensões reduzidas, baixa densidade e contendo artefatos. O método de extração proposto utiliza informações da percepção do sistema visual humano para a clusterização, associado a um método iterativo de avaliação da segmentação nas regiões de borda (área mais afetada por artefatos após o processo de compressão).

Além da proposta de um novo método de extração, pode-se destacar as seguintes contribuições adicionais:

- Comparação de desempenho entre 7 algoritmos de extração (incluindo o método proposto);

- A realização de uma interface gráfica capaz de produzir imagens de referência (*ground-truth*) e avaliar o desempenho dos algoritmos de extração implementados por meio da medida de discrepância de *probabilidade de erro* [42].

1.5.4 Etapa OCR

Após a extração dos caracteres, a imagem apresenta-se na forma binária. Contudo, ainda é necessário um pré-processamento anterior à entrada ao OCR. Tal pré-processamento destina-se a agrupar e normalizar as dimensões dos caracteres para alimentar o sistema de OCR. Dessa forma, o sistema de OCR pode utilizar um dicionário de palavras para uma melhoria na taxa de reconhecimento de caracteres.

Este trabalho propõe um método de clusterização hierárquica dos caracteres alinhados horizontalmente independente das dimensões dos caracteres e da imagem. Tal método é utilizado como uma etapa de pré-processa-

mento à etapa de reconhecimento dos caracteres pelo sistema de OCR Tesseract [34].

1.6 Estrutura da Dissertação

Esta dissertação está organizada como segue. O Capítulo 2 apresenta as principais técnicas de localização utilizadas atualmente, evidenciando as qualidades e os principais problemas relacionados a cada uma delas, bem como apresentando uma descrição detalhada do método de localização proposto. O Capítulo 3 descreve os atributos capazes de diferenciar regiões textuais de não-textuais como também a interface gráfica utilizada na avaliação de tais atributos. Após a avaliação da capacidade classificatória de cada atributo (texto e não-texto), descreve-se o modelo de verificação proposto baseado no treinamento da máquina de aprendizado SVM, utilizando o conjunto de atributos com melhor taxa de classificação. No Capítulo 4, é apresentado o método de extração desenvolvido, contendo uma descrição detalhada do método de clusterização de cor proposto, além de comparar o seu desempenho com diversos algoritmos comumente utilizados no processo de extração (binarização). Esse capítulo ainda descreve a interface gráfica concebida para gerar um banco de imagens *ground-truth* para a extração e permitir a avaliação de desempenho de diversos algoritmos extratores. O Capítulo 5 apresenta o método de clusterização hierárquica proposto para agrupar os caracteres extraídos e alinhados horizontalmente. Apresenta ainda a interface gráfica realizada para facilitar a comparação de sistemas TIE completos, bem como os resultados obtidos pelo sistema TIE proposto, utilizando o OCR Tesseract [34]. O Capítulo 6 apresenta os comentários e conclusões finais deste trabalho.

1.7 Notação

Todas as unidades, símbolos, operadores e abreviações presentes nesse trabalho seguem o padrão ISO 31/X1 [5].

Capítulo 2

Etapa de Localização

A localização e reconhecimento de texto em imagens (sistema TIE) é a área de pesquisa cujo objetivo é desenvolver sistemas com a habilidade de automaticamente ler o conteúdo do texto que está embutido em imagens. Para que tal sistema seja eficiente, o primeiro passo é determinar as regiões da imagem com maior probabilidade de possuir texto, sem necessariamente reconhecer os caracteres. Essa etapa, denominada *localização*, delimita as regiões da imagem que possuem características textuais através de BBs. Tal etapa reduz a área de processamento, inicialmente caracterizada por toda a imagem, para apenas um conjunto de regiões, tornando-se fundamental para a redução de complexidade computacional e sucesso da etapa de *extração*.

Os algoritmos de *localização* utilizam características extraídas da imagem para determinar as possíveis regiões textuais. Quanto às características utilizadas, os métodos de localização de texto podem ser divididos em duas categorias: baseados em *região* e baseados em *textura*. A Fig. 2.1 apresenta um diagrama da classificação dos métodos de localização, cujas características são discutidas a seguir.

2.1 Métodos Baseados em Região

Os textos legíveis apresentam contraste de crominância (imagem colorida) ou luminância (imagem em níveis de cinza) com o plano de fundo. Os métodos baseados em região consideram que a maioria dos caracteres possuem cores (ou níveis de cinza) perceptualmente distintas do plano de fundo

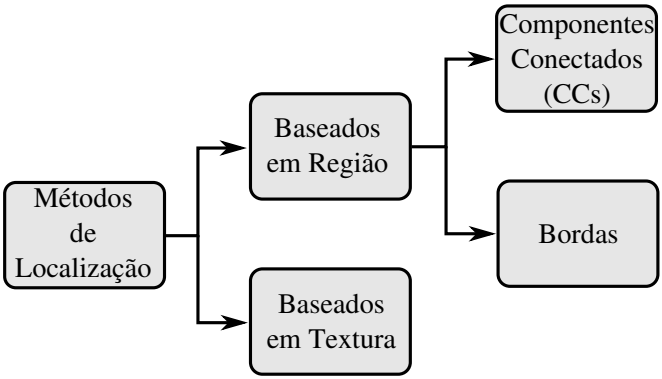


Fig. 2.1: Diagrama dos métodos de localização quanto às características utilizadas.

e utiliza tal característica para segmentar a imagem em regiões textuais e não-textuais (plano de fundo).

Os métodos baseados em região podem ser subdivididos em duas categorias: métodos baseados em componentes conectados (CC) (Apêndice A.1) e métodos baseados em *bordas*. Essas duas abordagens trabalham de maneira *bottom-up*, o que significa que tais métodos identificam subestruturas, tais como bordas e CCs, e as agrupam para delimitar as possíveis regiões que possuem texto.

2.1.1 Métodos Baseados em CC

Os métodos baseados em componentes conectados segmentam a imagem em um conjunto de CCs. Uma região da imagem que possui CCs com características geométricas similares e estão dispostos espacialmente sobre um eixo de alinhamento geralmente contém caracteres. Dessa maneira, os métodos baseados em CCs avaliam tais características, descartando os CCs considerados não-textuais e agrupando os textuais, até que todos os CCs gerados tenham sido avaliados. Ao final do processo, os CCs caracterizados como textuais estão agrupados e delimitados por BBs. Os agrupamentos são realizados mediante a avaliação de heurísticas referentes às restrições geométricas dos CCs, tais como razão de aspecto, densidade[1], regularidade na altura e espaçamento, alinhamento, dentre outros.

De maneira geral, os métodos baseados em CCs possuem 4 estágios de processamento:

[1]Razão entre o número de pixels que representa o CC e o número de pixels interno ao BB que delimita o CC.

(a) (b) (c) (d)

Fig. 2.2: Métodos baseados em CCs - geração de CCs. (a) Imagem original com caracteres de baixa densidade e artefatos incluídos durante o processo de compressão. (b) Extração dos CCs da imagem original, em que cada CC está representado por uma cor e delimitado por um BB em preto. (c) Geração correta do CCs da imagem. (d) Agrupamento dos CCs.

1. Pré-processamento, tais como clusterização de cor e redução de ruído.
2. Geração dos CCs.
3. Filtragem dos CCs não textuais.
4. Agrupamento dos CCs.

Os métodos baseados em CCs são computacionalmente menos complexos e relativamente mais fáceis de implementar do que os métodos baseados em textura. No entanto, são menos robustos à localização de texto sobre planos de fundo complexos, caracteres de baixa densidade e dimensões reduzidas.

Os métodos baseados em CCs possuem dois problemas principais, a saber: a segmentação para a geração dos CCs e o agrupamento dos componentes. Durante o processo de geração de CCs, um caractere de baixa densidade, com variação de cor e iluminação pode ser segmentado em vários CCs. Tal segmentação, de um único caractere em diversos CCs, prejudica a avaliação das características geométricas e espaciais contidas no processo de filtragem e agrupamento dos CCs, uma vez que a relação entre as dimensões, espaçamento e alinhamento dos componentes são comumente utilizados para determinar se um conjunto de CCs representa texto. A Fig. 2.2 apresenta a geração de CCs de uma imagem com caracteres de baixa densidade contendo artefatos [Fig. 2.2(a)]. Nota-se na Fig. 2.2(b) a fragmentação dos caracteres em vários CCs, em que cada CC está delimitado por um BB. Tal ocorrência prejudica todo o processo de classificação de tais componentes como textuais, visto que esses não possuem uma disposição espacial alinhada e são distintos geometricamente. A Fig. 2.2(c) ilustra a geração correta dos CCs (cada caractere transforma-se em um único CC), cujos BBs que os delimitam possuem alinhamento, espaçamento e alturas similares; o que permite inferir que os quatro CCs da Fig. 2.2(c) tratam-se de uma única palavra e devem ser agrupados [Fig. 2.2(d)].

A última etapa durante o processo de localização por métodos baseados em CCs é o agrupamento de componentes. Devido à complexidade em

agrupar caracteres em qualquer direção, muitos trabalhos assumem que os textos embutidos em imagens estão alinhados horizontalmente. Dessa forma, impõe-se aos algoritmos apenas a busca por componentes alinhados horizontalmente, eliminando qualquer possibilidade de localizar textos dispostos em outra direção. Logo, um grande obstáculo a ser superado é o agrupamento dos CCs em qualquer direção sem obter um aumento considerável de complexidade computacional e falsos alarmes.

Além disso, os métodos baseados em CCs geralmente utilizam diversos limiares heurísticos para determinar quais conjuntos de CCs representam texto ou não-texto, limitando-os a uma aplicação ou base de dados específica.

2.1.1.1 Revisão Bibliográfica dos Métodos Baseados em CC

Jain e Yu [27] iniciam a localização textual em imagens coloridas através da redução do número de cores da imagem. Tal redução é obtida considerando apenas os 2 primeiros bits mais significativos de cada plano RGB, transformando assim, uma imagem de 24-bits em uma de 6-bits. Essa técnica, denominada *bit-dropping*, é capaz de reduzir uma imagem que possui 2^{24} cores para uma imagem de apenas 64 cores. A Fig. 2.3 apresenta a seqüência de operações realizadas pelo método de Jain e Yu, em que uma imagem original [Fig. 2.3(a)] composta por dezenas de milhares de cores é reduzida para dezenas de cores utilizando o *bit-dropping* [Fig. 2.3(b)].

Após a redução do número de cores, uma clusterização hierárquica *single-link* [31] é realizada sobre a imagem com número reduzido de cores. Tal clusterização funde as duas cores com menor proximidade no espaço RGB e a cor resultante adquire, dentre as cores fundidas, aquela de maior ocorrência no histograma. A operação de clusterização continua até que o número de cores seja reduzido a 2 ou até que a menor distância na matriz de proximidade [31] seja maior do que 1. A Fig. 2.3(c) apresenta a imagem após a clusterização apresentando apenas 5 cores.

Jain e Yu assumem que os caracteres de uma mesma palavra possuem cores similares e como conseqüência são agrupados em um mesmo cluster. Com base nessa premissa, criam-se $n + 1$ imagens binárias, em que n é o número de clusters gerados. Tais clusters definem o número de cores restantes na imagem após o processo de clusterização. Para cada cor da imagem, uma imagem binária é criada atribuindo-se o valor ‘1’ para pixels de determinada cor e ‘0’ para os demais. Desse modo, n imagens binárias são criadas. Além disso, uma imagem binária adicional é gerada atribuindo-se o valor ‘0’ para as duas cores com maior número de pixels e o valor binário ‘1’ para todas as outras cores. Utilizando tal artifício, o autor visa contemplar a localização de palavras constituídas de caracteres com cores perceptualmente diferentes.

(a) (b)

(c) (d)

Fig. 2.3: Seqüência de operações realizadas pelo método de localização proposto por Jain e Yu. (a) Imagem original composta por 46705 cores. (b) Imagem após o *bit-dropping* apresentando apenas 23 cores. (c) Imagem após a clusterização das cores via método *single-link*, resultando em uma imagem composta de apenas 5 cores. (d) Resultado da localização (BBs em azul).

Uma vez que as imagens binárias tenham sido criadas, agrupam-se os CCs de cada imagem que estão alinhados horizontalmente e possuem características geométricas semelhantes. Após o agrupamento, a avaliação por projeções de perfil [27] é realizada visando eliminar possíveis CCs não-textuais agrupados. Todas as regiões identificadas como textuais em cada imagem binária são delimitadas na imagem original como resultado final do processo de localização [Fig. 2.3(d)].

A técnica proposta por Jain e Yu possui baixa complexidade compu-

tacional, porém contempla apenas a localização de palavras que possuem um número maior ou igual a 4 caracteres e estão alinhados horizontalmente. O *bit-dropping* reduz o número de cores de imagens cuja ordem de grandeza é de 10^7 para 10^2. Como conseqüência, caracteres com pouco contraste com o plano de fundo são segmentados em vários CCs ou até mesmo fundidos com o plano de fundo em uma única cor. A redução de cores via quantização torna o método ineficiente na localização de caracteres de dimensões reduzidas ou de baixa densidade. Contudo, o autor enfatiza que o método é eficiente na busca das informações mais importantes da imagem, geralmente contidas em caracteres de maior dimensão e alinhados horizontalmente.

Um outro método, proposto por Messelodi e Modena [47], consiste de 3 estágios: (i) extração dos objetos elementares; (ii) filtragem dos objetos; e (iii) seleção das linhas de texto. A extração dos objetos elementares exige um pré-processamento de normalização da intensidade da imagem em níveis de cinza original, como mostrado nas Figs. 2.4(a) e (b). O passo seguinte é a criação de duas imagens binárias mediante o uso de dois limiares globais (l_1 e l_2) aplicados sobre a imagem de intensidade normalizada[2], apresentadas nas Figs. 2.4(c) e (d). Após a geração das imagens binárias, obtém-se os CCs. Vários filtros baseados nas *características internas*, incluindo área, dimensões relativas, razão de aspecto, densidade e contraste são aplicados para eliminar componentes não-textuais. Finalmente, a seleção da linha de texto se inicia em uma única região e recursivamente se expande, até que um critério de parada seja satisfeito. Tal critério utiliza limiares sobre as *características externas*, tais como regularidade, alinhamento e similaridade de altura. Apesar de o método proposto por Messelodi e Modena possuir um sistema de localização capaz de identificar caracteres em diferentes orientações[3], os próprios autores sinalizam que a seleção dos filtros e seus limiares são altamente dependentes da aplicação e dimensões da imagem de entrada.

2.1.2 Métodos Baseados em Bordas

Diversas características textuais são utilizadas para a localização de texto em imagens, no entanto, os métodos baseados em bordas (*edge-based*) exploram o alto contraste entre o texto e o plano de fundo. Atuando de maneira *bottom-up*, os métodos baseados em bordas utilizam algum operador de detecção de borda (Canny, Sobel, e outros [23].) sobre a imagem. Posteriormente empregam filtros heurísticos para seleção das bordas textuais e as fundem por

[2] $l_1 = m - \frac{d_\mathrm{l}}{2}$ e $l_2 = m + \frac{d_\mathrm{r}}{2}$, em que m é o modo do histograma e d_l e d_r são os desvios esquerdo e direito do modo m, respectivamente.

[3] Os métodos baseados em CCs geralmente assumem que o texto presente na imagem está na direção horizontal.

(a) (b)

(c) (d)

Fig. 2.4: Seqüência de operações do método proposto por Messelodi e Modena. (a) Imagem em escala de cinzas original. (b) Imagem após a normalização da intensidade. (c) Imagem binária resultante da limiarização da imagem normalizada cujos pixels possuem níveis de cinza menores do que o limiar l_1. (d) Imagem binária resultante da limiarização da imagem normalizada cujos pixels possuem níveis de cinza maiores do que o limiar l_2.

meio de operadores morfológicos [58] ou de suavização para a delimitação das regiões textuais (BBs).

Chen et al. [12] geram inicialmente duas imagens de bordas, uma de bordas verticais e outra horizontais, por meio da detecção Canny [51]. As bordas textuais geralmente possuem coesão espacial, logo a densidade de bordas na direção vertical e horizontal em regiões de texto é superior a do plano de fundo. Baseando-se nessa característica, a operação morfológica de dilata-

ção [58] é utilizada visando conectar o conjunto de bordas transformando-as em *clusters*. De acordo com o tipo da imagem de bordas (vertical ou horizontal), diferentes operadores de dilatação são utilizados. A imagem de bordas verticais é dilatada por meio de um elemento estruturante de 1×5, visando fundir as bordas verticais na direção horizontal. A imagem de bordas horizontais é dilatada por meio de um elemento estruturante 6×3, fundindo as bordas horizontais na direção vertical. As duas imagens dilatadas são então submetidas a uma operação binária *AND*, com o objetivo de destacar as regiões densamente povoadas por bordas em ambas as direções. Tais regiões são então classificadas como textuais ou não-textuais utilizando um classificador SVM.

O método de Chen et al. apesar de propor uma etapa robusta de verificação, é capaz apenas de localizar texto cujas dimensões são menores do que os elementos estruturantes dos operadores morfológicos [58]. O autor não utiliza qualquer técnica multi-resolução para identificação de texto em diferentes dimensões, o que restringe o método a aplicações específicas.

Liu et al. [43] baseiam-se em um modelo híbrido, em que a localização é realizada mediante a identificação de bordas da imagem e a verificação de tais regiões mediante métodos texturais. Liu et al. iniciam o pré-processamento utilizando um filtro de mediana, visando eliminar o ruído presente nas imagens, seguido de um detector de borda em campos vetoriais [38]. Um limiar adaptativo seleciona os contornos mais evidentes. Em seguida, tais contornos são filtrados por meio das suas características estruturais e texturais (via, por exemplo, a transformada *wavelet*). Os contornos remanescentes são então caracterizados como textuais e delimitados por BBs.

2.2 Métodos Baseados em Textura

A característica mais intuitiva de um texto é a sua regularidade [67]. O texto é constituído por caracteres com aproximadamente o mesmo tamanho, mesma espessura de traço (corpo do caractere) e localizados a uma distância regular uns dos outros. Tais regularidades vêm sendo exploradas implicitamente em [29], [59] ao considerar que as regiões textuais possuem um certo tipo de textura cujos componentes freqüenciais são distintos dos outros objetos da imagem. Tal consideração é válida, visto que na direção de escrita flutuações periódicas de cor (ou níveis de cinza) podem ser observadas.

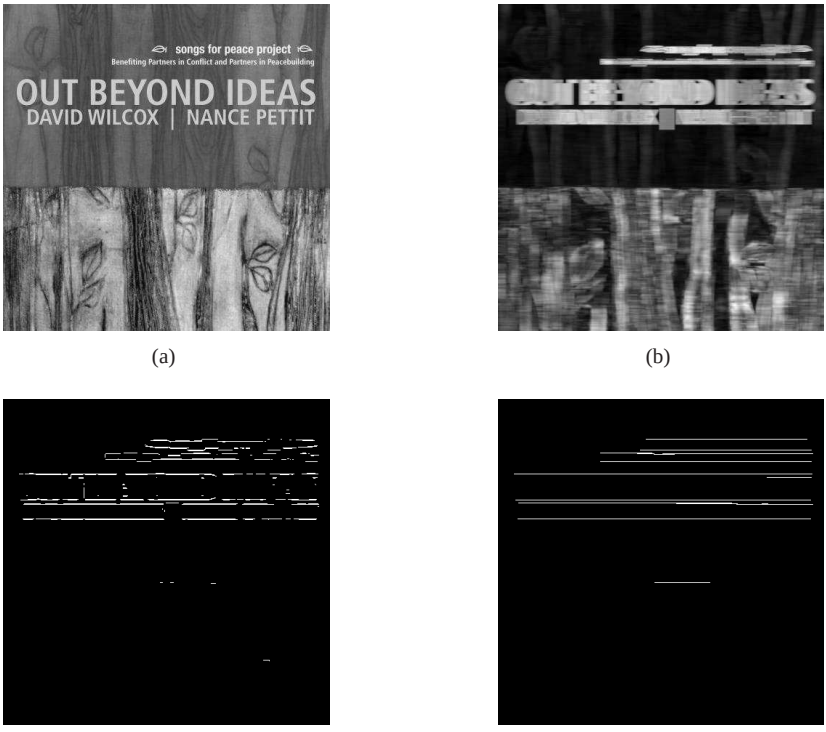
Baseado na premissa de que o texto possui propriedades texturais distintas do plano de fundo, qualquer técnica capaz de identificar regiões constituídas por diferentes componentes freqüenciais pode ser utilizada para segmentar e identificar regiões de texto em uma imagem, tais como: filtros de Gabor [26], *wavelet* [22], [55], *fast Fourier transform* (FFT) [59] e variância espacial [70].

A maior desvantagem dos métodos baseados em textura é a complexidade computacional envolvida no estágio de classificação textural, que é superior a dos métodos descritos anteriormente. A filtragem baseada em textura, para ser eficiente, requer uma varredura da imagem de entrada em diversas resoluções. Além disso, caracteres com ascendência ou descendência geralmente não são localizados completamente devido à ausência de textura fora da região de alinhamento dos caracteres.

2.2.1 Revisão bibliográfica dos métodos baseados em textura

Wu et al. [66], [67] propuseram um sistema de extração textual baseado em um esquema de segmentação de textura multi-escala. O método assume que textos possuem uma textura diferenciada do plano de fundo e de outros objetos da imagem. Conseqüentemente, o texto pode ser extraído mediante a utilização de técnicas de segmentação de textura. As potenciais áreas textuais são obtidas filtrando a imagem por meio de três filtros derivativos gaussianos sob três diferentes escalas. Uma transformação não-linear é aplicada a cada uma das 9 imagens resultantes do processo de filtragem. Para cada uma das 9 imagens, estima-se a energia local utilizando a saída da transformação não-linear. A estimativa de energia dos pixels correspondentes das 9 imagens formam um vetor. Tais vetores são então clusterizados utilizando o algoritmo *K-means* ($K = 3$). O processo é referido como segmentação de textura. O passo seguinte, rotulado pelo autor de *chip generation*, visa extrair, filtrar e agregar os traços dos caracteres presentes nas regiões identificadas após a clusterização. A segmentação de textura e extração de caracteres descrita anteriormente é realizada em múltiplas escalas para a detecção de texto de diferentes dimensões e, então, mapeadas sobre a imagem original.

Zhong et al. [70] assume que a variância das regiões textuais é maior do que o plano de fundo devido à grande variação de intensidade entre caracteres de uma palavra. Zhong et al. propuseram a identificação de regiões textuais sobre a imagem em níveis de cinza [Fig. 2.5(a)] mediante o cálculo da variância de intensidade dos pixels na direção horizontal. Através de uma máscara horizontal de 1×21, Zhong et al. calcularam a variância espacial dos pixels em uma vizinhança local. Isso gera uma imagem-variância em que os pixels localizados em uma região de alta variância possuem um valor de intensidade maior (mais próxima da cor branca), como ilustrado na Fig. 2.5(b). Extraem então as bordas da imagem-variância mediante a detecção Canny [9], como ilustrado na Fig. 2.5(c). Todavia, tais bordas, que representam os limites superior ou inferior de regiões textuais, não possuem continuidade (estão fragmentadas em diversos pontos) [Fig. 2.5(c)]. Uma filtragem utilizando heurísticas



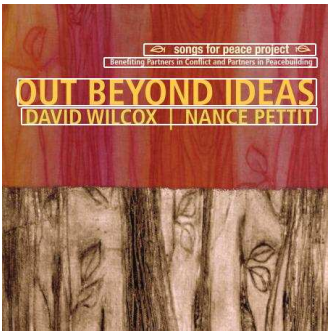
(a) (b)

(c) (d)

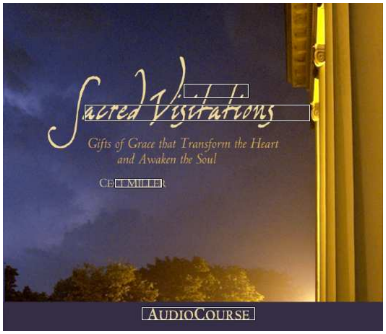
Fig. 2.5: Seqüência de operações do método baseado em textura desenvolvido por Zhong et al.. (a) Imagem em escala de cinzas original. (b) Imagem-variância. (c) Imagem após a detecção de bordas Canny. (d) Imagem após filtragem das bordas verticais e agrupamento das bordas alinhadas na direção horizontal.

de alinhamento e espaçamento visam selecionar as bordas que representam as regiões textuais e as fundem formando longas linhas, como mostrado na Fig. 2.5(d). Pares de linhas de dimensões similares são agrupadas e define-se como região textual a área entre duas linhas que possuem direções opostas do gradiente. Cada par de linhas identificado corresponde ao limite superior e inferior de uma região textual. O resultado é então exposto por meio de BBs, como apresentado na Fig. 2.6(a).

O maior problema do método de Zhong et al. está na incapacidade de localização de caracteres ascendentes ou descendentes como ilustrado na Fig. 2.6(b), em que as letras ‘S’, ‘d’, ‘V’, ‘t’ e ‘t’ das palavras ‘**S**acre**d** **V**isi**t**a-**t**ions’ não foram localizadas corretamente.



(a)



(b)

Fig. 2.6: Exemplos de resultados do método de localização proposto por Zhong et al.. (a)Resultado da localização do método proposto por Zhong et al.. (f) Imagem que caracteriza a incapacidade do método na localização de palavras que possuem caracteres ascendentes ou descendentes.

2.3 Método de Localização Proposto

Imagens complexas são tipicamente de baixa resolução, coloridas e apresentam artefatos incluídos durante o processo de compressão. Exibem textos, em sua maioria, esparsos, possuindo uma quantidade reduzida de caracteres. Além disso, possuem caracteres de fontes variadas podendo apresentar efeitos, tais como sombreamento e transparência sobre um plano de fundo texturizado. Assim, é importante que o método de localização seja o menos dependente possível da regularidade, textura, fonte e dimensão dos caracteres.

Um método de localização de texto para imagens complexas deve atingir os seguintes objetivos:

1) Ser independente em relação às dimensões da imagem de entrada e aparência do texto, tais como caracteres de variadas fontes, dimensões, espaçamento, alinhamento, orientação, contraste[4] e cor[5].

2) Capacidade de localização individual de caracteres, proporcionando a identificação de caracteres isolados.

3) Ser capaz de identificar caracteres embutidos em planos de fundo complexos.

[4]Caracteres mais escuros do que o plano de fundo e vice-versa.

[5]Superfícies curvas, objetos 3D e texto com distorções de perspectiva não recebem nenhum tratamento adicional por serem considerados menos freqüentes.

(a) (b) (c)

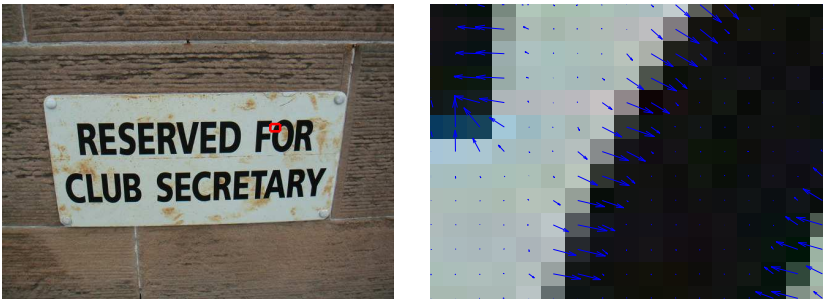
Fig. 2.7: Características dos caracteres legíveis. (a) Caracteres policromáticos cujas cores possuem contraste com o plano de fundo. (b) Contorno extraído dos caracteres policromáticos da Fig. 2.7(a) por meio da avaliação da magnitude do gradiente. (c) Caractere representado apenas pelo seu contorno.

Para que o método de localização satisfaça o objetivo (1) supracitado, é necessário utilizar alguma característica presente nos caracteres que independa das variações de fonte, espaçamento, orientação, cor e dimensão dos caracteres. Uma característica invariante a tais parâmetros, presente na maioria dos caracteres legíveis, é a variação abrupta de cor ou de intensidade existente no limite entre os pixels de contorno dos caracteres (bordas) e o plano de fundo. Quando tal contraste existe, o contorno dos caracteres, mesmo sendo policromático, pode ser completamente extraído, como ilustrado na Fig. 2.7. Pode-se afirmar ainda que o caractere é completamente caracterizado pelo seu contorno, como pode ser observado na Fig. 2.7(c).

Visto que a maioria dos caracteres legíveis possuem uma variação abrupta de cor (imagem colorida) ou intensidade (imagem em níveis de cinza) na região entre o seu contorno e o plano de fundo, o método de localização proposto mede tal variação calculando o gradiente de intensidade dos pixels da imagem em níveis de cinza. Liu et al. [43] mostram que a magnitude do gradiente de intensidade nos pixels de borda dos caracteres é superior à maioria dos pixels do plano de fundo da imagem[6]. A Fig. 2.8 apresenta uma região em destaque da *imagem-exemplo* e os vetores gradiente de intensidade correspondentes a cada pixel da região, em que as setas azuis indicam a magnitude (proporcional ao tamanho do segmento da seta) e direção do gradiente. Sendo assim, o algoritmo de localização proposto possui como pilar fundamental a detecção e seleção dos contornos (pixels de borda). Os contornos considerados textuais são aqueles cujos pixels possuem a magnitude do gradiente superior a um limiar calculado especificamente para a imagem. Tais contornos são então delimitados por BBs e as regiões delimitadas tornam-se candidatas a texto.

A utilização de bordas como principal característica para a localização de caracteres possui como vantagem, sobre os métodos baseados em textura, a

[6]Um estudo comparativo da magnitude do gradiente dos pixels de borda dos caracteres e não-caracteres é apresentado no Capítulo 3, cujos resultados corroboram a afirmação de Liu et al. [43].



(a) (b)

Fig. 2.8: Exemplo dos vetores gradiente em uma região da imagem-exemplo. (a) Imagem-exemplo com uma região destacada para a apresentação dos vetores gradiente na Fig. 2.8(b). (b) Apresentação dos vetores gradiente de intensidade de cada pixel da região destacada pelo BB vermelho na Fig. 2.8(a).

capacidade de localização de caracteres isolados [objetivo (2)]. Uma vez que a localização proposta depende apenas da magnitude do gradiente de intensidade dos pixels de borda, seu uso é capaz de identificar caracteres isolados, de qualquer fonte e dimensão.

Os métodos baseados em textura geralmente utilizam janelas de avaliação que varrem toda a imagem em busca de caracteres, porém são ineficientes em localizar caracteres isolados, visto que um único caractere geralmente não possui regularidade suficiente para caracterizar textura. Além disso, a eficiência do método é dependente do tamanho e espaçamento das fontes relativo à janela de avaliação, em que técnicas multi-resolução [39], [41], [66], [67] são comumente utilizadas para minorar tal problema. Nesse aspecto, o método proposto é computacionalmente vantajoso sobre os métodos texturais, devido à capacidade de identificar texto de variadas dimensões sem a utilização de múltiplas resoluções.

Uma outra vantagem relacionada ao método proposto é a ausência de heurísticas durante o processo de localização relacionadas ao alinhamento, espaçamento e proporções geométricas comumente utilizados nos métodos baseados em CCs. Tais heurísticas são incapazes de contemplar a variedade de orientações, estilos e dimensões das fontes. Além disso, a utilização de bordas para identificação permite a localização de caracteres policromáticos e de baixa densidade, visto que os contornos de tais caracteres são extraídos de forma independente do número de cores presente no corpo dos caracteres, como ilustrado na Fig. 2.7.

Para extrair as regiões candidatas a texto, o algoritmo de localização

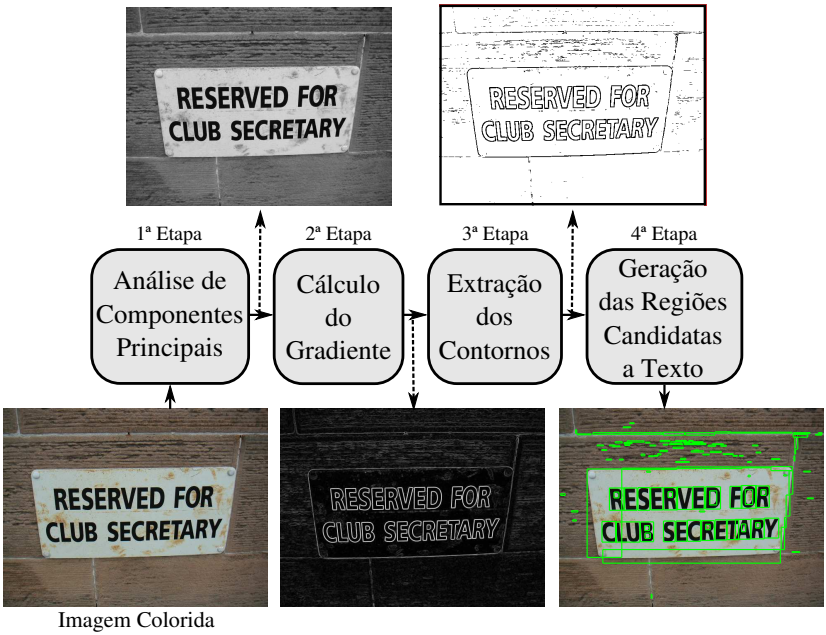


Fig. 2.9: Diagrama de fluxo do algoritmo de localização proposto.

proposto utiliza quatro etapas, mostradas na Fig. 2.9.

A primeira etapa visa transformar a imagem colorida (RGB) em uma imagem em níveis de cinza por meio da *principal component analysis* (PCA) [19]. Uma vez obtida a imagem em níveis de cinza, a segunda etapa calcula a magnitude e direção do gradiente de intensidade para cada pixel da imagem. Produz-se assim uma imagem que representa a magnitude do gradiente de intensidade de cada pixel, em que os pixels de maior magnitude do gradiente estão mais próximas da cor branca, como ilustrado na saída da segunda etapa na Fig. 2.9. A terceira etapa gera um limiar adaptativo L_{M} para a imagem a partir da avaliação da magnitude e direção do gradiente de intensidade dos pixels da imagem. Os pixels cuja magnitude do gradiente de intensidade é maior do que o limiar L_{M} recebem o valor binário '0' (preto), enquanto todos os outros, o valor binário '1' (branco). A imagem binária resultante da terceira etapa é composta pelos contornos de caracteres e objetos de grande contraste. A quarta etapa encarrega-se de transformar os contornos de tal imagem binária em CCs, filtrá-los e delimitá-los como regiões candidatas a texto através de BBs. As seções seguintes descrevem cada etapa detalhadamente.

(a)

(b)

(c)

(d)

Fig. 2.10: Imagem-exemplo em RGB e os planos R, G e B que a compõe. (a) Imagem-exemplo em RGB. (b) Plano R da imagem-exemplo. (c) Plano G da imagem-exemplo. (d) Plano B da imagem-exemplo. Os coeficientes de correlação da imagem-exemplo são $\rho_{\mathrm{rg}} = 0.97274$ entre os planos R e G, $\rho_{\mathrm{rb}} = 0.93498$ entre os planos R e B e $\rho_{\mathrm{gb}} = 0.9865$ entre os planos G e B.

2.3.1 Análise de Componentes Principais - PCA

O êxito do método de localização proposto está vinculado ao sucesso da detecção de bordas da imagem. No entanto, as técnicas de detecção de bordas para imagens coloridas possuem complexidade computacional elevada quando comparadas às imagens em níveis de cinza. Para reduzir tal complexidade, o método proposto tira proveito da alta correlação existente entre os canais R, G e B da imagem colorida (veja Fig. 2.10) para obter uma imagem em níveis de cinza sem perda considerável de informação. Tal imagem em níveis de cinza é obtida utilizando a técnica PCA, conhecida também como *Karhunen-Loève transform* (KLT) ou transformada de Hotelling [19]. A PCA transforma os dados da imagem colorida (eixos R, G e B) para uma nova base (componentes principais) [50], cujo eixo do primeiro componente principal está localizado na direção da máxima variância dos dados originais e os novos componentes estão estatisticamente descorrelacionados, como ilustrado na Fig. 2.11. A projeção dos dados sobre o primeiro componente principal gera uma imagem em níveis de cinza ótima em relação ao *mean square error* (MSE) [19], referenciada neste trabalho como $\boldsymbol{P_1}$.

(a) (b) (c)

Fig. 2.11: Análise de componentes principais aplicada à imagem-exemplo [Fig. 2.10(a)]. (a) Dados projetados sobre o primeiro componente principal - imagem $\boldsymbol{P_1}$. (b) Dados projetados sobre o segundo componente principal - imagem $\boldsymbol{P_2}$. (c) Dados projetados sobre o terceiro componente principal - imagem $\boldsymbol{P_3}$.

A imagem-exemplo projetada sobre os componentes principais ($\boldsymbol{P_1}$,$\boldsymbol{P_2}$ e $\boldsymbol{P_3}$) está ilustrada na Fig. 2.11, onde nota-se que a maior parte da informação está contida no primeiro componente principal (imagem $\boldsymbol{P_1}$). Devido ao alto índice de correlação existente entre os planos R, G e B, um baixo conteúdo de informação está presente no segundo e terceiro componentes principais [Figs. 2.11(b) e (c)]. Dessa forma, este trabalho, visando reduzir complexidade computacional durante o processo de detecção de bordas, utiliza apenas a imagem em níveis de cinza $\boldsymbol{P_1}$ para as etapas posteriores.

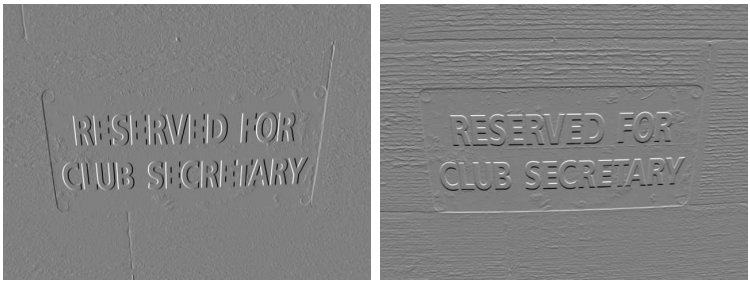
2.3.2 Cálculo do Gradiente de Intensidade

O método de localização proposto, identifica as regiões candidatas a texto delimitando as áreas de variação abrupta de intensidade, uma vez que a legibilidade dos caracteres está relacionada ao contraste dos caracteres com plano de fundo da imagem. Sendo assim, uma vez obtida a imagem em níveis de cinza $\boldsymbol{P_1}$, o método proposto calcula o gradiente de intensidade da imagem $\boldsymbol{P_1}$ por meio dos filtros derivativos horizontal $\boldsymbol{H_h}$ e vertical $\boldsymbol{H_v}$ de Prewitt [7], visando identificar a direção de maior variação de intensidade para cada pixel da imagem. A *point spread function* (PSF)[7] do filtro derivativo de Prewitt no domínio espacial ao longo da direção vertical e horizontal é, respectivamente, dada por

$$\boldsymbol{H_v} = \frac{1}{3}\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \text{e} \quad \boldsymbol{H_h} = \frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}. \qquad (2.1)$$

Convolvendo os filtros $\boldsymbol{H_v}$ e $\boldsymbol{H_h}$ com a imagem em níveis de cinza $\boldsymbol{P_1}$

[7] A resposta ao impulso no domínio espacial definida em processamento de imagens.



(a) (b)

Fig. 2.12: Imagens referentes às derivadas parciais da imagem $\boldsymbol{P_1}$. (a) Imagem $\boldsymbol{G_x}$ resultante da convolução entre $\boldsymbol{H_v}$ e $\boldsymbol{P_1}$. (b) Imagem $\boldsymbol{G_y}$ resultante da convolução entre $\boldsymbol{H_h}$ e $\boldsymbol{P_1}$.

no domínio da seqüência, obtém-se as imagens $\boldsymbol{G_x}$ e $\boldsymbol{G_y}$, respectivamente, como ilustrado na Fig. 2.12. Dado que as imagens em níveis de cinza $\boldsymbol{G_x}$ e $\boldsymbol{G_y}$ podem ser consideradas as derivadas parciais da imagem $\boldsymbol{P_1}$ [23], o gradiente de intensidade da imagem $\boldsymbol{P_1}$ pode ser representado da seguinte forma:

$$\nabla \boldsymbol{P_1} = \begin{bmatrix} \boldsymbol{G_x} \\ \boldsymbol{G_y} \end{bmatrix} = \begin{bmatrix} \frac{\partial \boldsymbol{P_1}}{\partial x} \\ \frac{\partial \boldsymbol{P_1}}{\partial y} \end{bmatrix}. \tag{2.2}$$

Logo, a magnitude $G_{\mathrm{mag}}(i,j)$ e direção $G_\theta(i,j)$ do vetor gradiente para cada pixel (i,j) da imagem $\boldsymbol{P_1}$ são dadas por

$$G_{\mathrm{mag}}(i,j) = \sqrt{G_{\mathrm{x}}(i,j)^2 + G_{\mathrm{y}}(i,j)^2} \tag{2.3}$$

e

$$G_\theta(i,j) = \arctan\left(\frac{G_{\mathrm{y}}(i,j)}{G_{\mathrm{x}}(i,j)}\right). \tag{2.4}$$

Uma vez obtida $\boldsymbol{G_{\mathbf{mag}}}$ e $\boldsymbol{G_\theta}$, sabe-se a magnitude e direção da variação de intensidade para cada pixel da imagem, como ilustrado na Fig. 2.13(a), em que setas azuis representam a magnitude (proporcional ao tamanho do segmento das setas) e direção do gradiente de intensidade. A Fig. 2.13(b) ilustra a magnitude do gradiente de intensidade sob a forma de imagem ($\boldsymbol{G_{\mathbf{mag}}}$), em que os pixels de maior magnitude do gradiente estão mais próximos da cor branca.

2.3.3 Extração dos Contornos Candidatos a Texto

Os pixels de borda (contornos) de uma imagem são aqueles que representam máximos locais de magnitude do gradiente em uma dada direção [43]. Uma vez que estamos interessados apenas nos *pixels de borda textuais*[8] e tais pixels geralmente possuem valores de magnitude do gradiente superiores à maioria dos pixels da imagem [veja Fig. 2.13(b)], faz-se primeiro a identificação dos pixels de borda e, posteriormente, selecionam-se aqueles cuja magnitude do gradiente é superior a um limiar L_{M}.

Assumindo que a média da magnitude do gradiente dos *pixels de borda textuais* são superiores à maioria dos pixels da imagem [43], um pixel (i,j) é aceito como um possível *pixel de borda textual* caso obedeça às seguintes restrições:

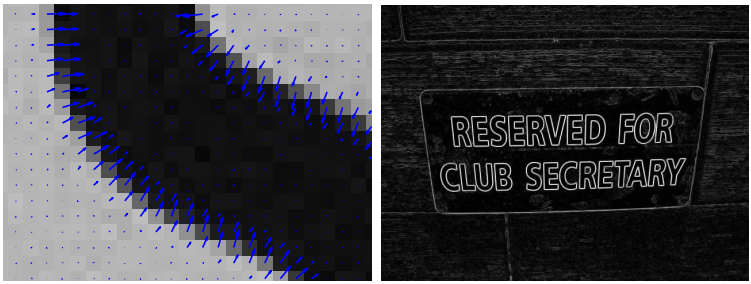
1. *A magnitude do gradiente do pixel* (i,j) *deve ser maior do que a magnitude de ambos os pixels da vizinhança-8 -* $(i^{'},j^{'})$ *e* $(i^{''},j^{''})$ *- relacionados à direção do gradiente do pixel* (i,j). Assim,

$$G_{\mathrm{mag}}(i,j) > G_{\mathrm{mag}}(i',j') \qquad \text{e} \qquad G_{\mathrm{mag}}(i,j) > G_{\mathrm{mag}}(i'',j'').$$

   Os (dois) pixels da vizinhança-8, cuja magnitude do gradiente é comparada ao pixel (i,j), são definidos de acordo com ângulo do gradiente do pixel sob avaliação (i,j), como mostrado na Fig. 2.14. Tal figura apresenta um círculo de correspondência de cores entre ângulos do gradiente do pixel (i,j) e os pixels da vizinhança-8 (matriz de pixels 3×3).

   Para exemplificar, dois casos são mostrados na Fig. 2.15. No primeiro

[8] Pixels que representam os contornos dos caracteres.



(a) (b)

Fig. 2.13: Gradiente da imagem-exemplo. (a) Apresentação dos vetores gradiente de cada pixel (setas azuis) em uma região da imagem-exemplo. (b) Imagem representando a magnitude do gradiente $\boldsymbol{G_{\mathrm{mag}}}$ (quanto maior a magnitude, mais próximo da cor branca os pixels se apresentam).

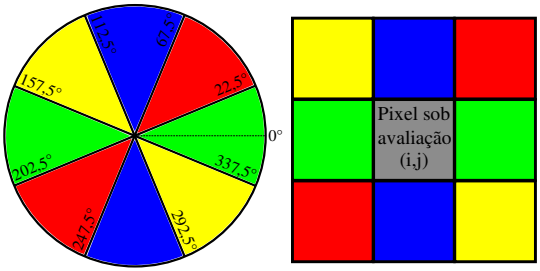


Fig. 2.14: Círculo de correspondência de cores entre os ângulos do gradiente do pixel (i,j) e os pixels da vizinhança-8 $(i^{'},j^{'})$ e $(i^{''},j^{''})$.

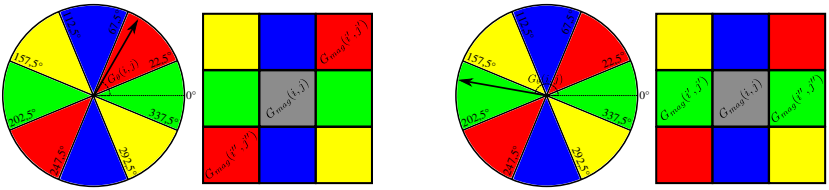


Fig. 2.15: Exemplos da correspondência entre o ângulo do vetor gradiente do pixel (i,j) sob avaliação e os pixels da vizinhança-8 $(i^{'},j^{'})$ e $(i^{''},j^{''})$. (a) Ilustração do primeiro caso. (b) Ilustração do segundo caso.

caso [Fig. 2.15(a)], o gradiente do pixel sob avaliação (i,j) (centro da matriz 3×3 de pixels) possui um ângulo (G_θ) igual a 60°, como indicado no círculo de cores. Assim, o vetor gradiente do pixel sob avaliação (i,j) encontra-se sobre a região vermelha do círculo (entre 22,5°e 67,5°); isso determina que os pixels da vizinhança-8 $(i^{'},j^{'})$ e $(i^{''},j^{''})$ correspondentes ao referido ângulo são os diagonais destacados na mesma cor da região do círculo.

No segundo caso [Fig. 2.15(b)], o vetor gradiente ($G_\theta = 170°$) encontra-se sobre a área em verde. Logo, os pixels da vizinhança-8 $(i^{'},j^{'})$ e $(i^{''},j^{''})$, cuja magnitude G_{mag} são comparadas ao pixel sob avaliação (i,j), são os destacados em verde na matriz de pixels 3×3. Contudo, entre os pixels da vizinhança-8 $(i^{'},j^{'})$ e $(i^{''},j^{''})$, compara-se apenas aqueles cujo vetor gradiente encontra-se na mesma região do círculo do pixel (i,j). Caso nenhum pixel da vizinhança-8 seja apto à comparação, o pixel sob avaliação (i,j) continua candidato a um pixel de borda textual.

2. A segunda exigência é proposta por Liu et al. [43], em que a magnitude do gradiente do pixel (i,j) deve ser maior do que um limiar L_{M} obtido para cada imagem. Tal exigência visa selecionar as bordas de maior magnitude do gradiente da imagem, visto que essas possuem maior chance de representarem bordas textuais. O limiar L_{M} é determinado pela seguinte equação:

$$L_{\mathrm{M}} = \frac{\sum\limits_{(i,j)\in \boldsymbol{P_1}} (G_{\mathrm{mag}}(i,j) \cdot |G_{\mathrm{dif}}(i,j)|)}{\sum\limits_{(i,j)\in \boldsymbol{P_1}} |G_{\mathrm{dif}}(i,j)|} \tag{2.5}$$

e

$$G_{\mathrm{dif}}(i,j) = G_{\mathrm{mag}}(i^{'},j^{'}) - G_{\mathrm{mag}}(i^{''},j^{''}) \tag{2.6}$$

onde $\boldsymbol{P_1}$ é a imagem em níveis de cinza; $G_{\mathrm{mag}}(i^{'},j^{'})$ e $G_{\mathrm{mag}}(i^{''},j^{''})$ são a magnitude do gradiente dos pixels da vizinhança-8 do pixel (i,j), como ilustrado na Fig. 2.15.

A primeira condição busca os pixels que contêm máximos locais de magnitude em uma determinada direção. Tal operação atribui aos pixels que atendem a primeira condição o valor binário '0' (preto) e a todos os outros o valor binário '1' (branco). Assim, uma imagem binária é criada destacando-se os pixels candidatos a bordas textuais (em preto), como ilustrado na Fig. 2.16(a). Uma segunda imagem binária, relacionada à segunda condição, é criada [Fig. 2.16(b)] mediante a limiarização da imagem de magnitude $\boldsymbol{G_{\mathrm{mag}}}$ [Fig. 2.13(b)] utilizando o limiar L_{M}. Atribui-se aos pixels de $\boldsymbol{G_{\mathrm{mag}}}$, cuja magnitude do gradiente é superior a L_{M}, o valor binário '0' e a todos os outros o valor '1'. Uma operação lógica *OR* é então realizada entre as duas imagens binárias [Figs. 2.16(a) e (b)] para selecionar os pixels que atendem as duas exigências. A Fig. 2.16(c) apresenta o resultado da extração das bordas consideradas textuais da imagem-exemplo.

2.3.4 Geração das Regiões Candidatas a Texto

Após a extração das bordas consideradas textuais [Fig. 2.16(c)] faz-se necessário delimitar as regiões da imagem candidatas a texto para posterior verificação. Tais regiões são obtidas através da transformação de cada borda extraída em um CC, como mostrado na Fig. 2.17(a), em que cada CC possui uma cor. Após a transformação de cada borda em um CC, como mostrado na Fig. 2.17(b), cada CC é delimitado através de um BB. A região interna de cada BB, para ser considerada candidata a texto, deve atender a algumas restrições quanto as suas dimensões:

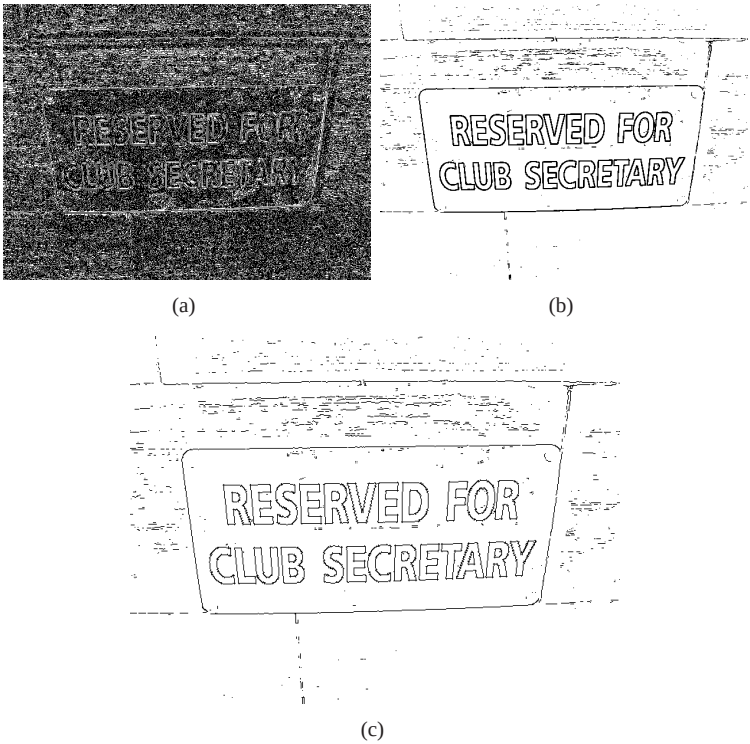
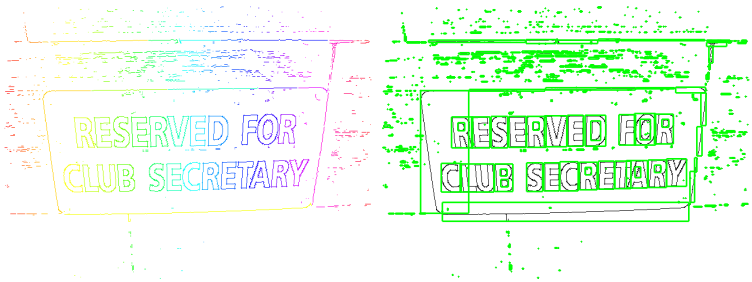


Fig. 2.16: Extração das bordas consideradas textuais da imagem-exemplo. (a) Imagem binária em que os pixels em preto representam os máximos locais em uma determinada direção. (b) Imagem binária resultante da limiarização da imagem magnitude $\boldsymbol{G}_{\mathbf{mag}}$ por meio do limiar L_{M}. (c) Resultado da extração das bordas da imagem consideradas textuais.

1. **Área Mínima**. Os BBs devem possuir uma área ($\mathrm{BB}_{\mathrm{area}}$) maior do que 20 pixels. Tal restrição é utilizada pela pequena quantidade de caracteres existentes com tais dimensões e a incapacidade de reconhecimento desses pelos sistemas OCR. Tal restrição foi proposta por Messelodi et al. [47], em que o autor ressalta a possibilidade do não reconhecimento das pontuações como regiões candidatas a caracteres. Assim,

$$\mathrm{BB}_{\mathrm{area}} > 20 \ \text{ pixels}.$$

2. **Área Máxima**. A área de um único BB ($\mathrm{BB}_{\mathrm{area}}$) não deve ser maior do que 40% da área da imagem $\mathrm{I}_{\mathrm{area}}$. Visto que o método de localização proposto identifica caracteres individuais, é improvável que um BB cor-



(a) (b)

Fig. 2.17: Transformação de bordas candidatas a texto em CCs e delimitadas por BBs. (a) Imagem de bordas transformadas em CCs, em que cada CC está representado em uma cor. (b) Imagem de CCs delimitados por BBs.

respondente a um único caractere possua uma área maior do que 40% da área da imagem. Então,

$$\mathrm{BB}_{\mathrm{area}} \leq 0{,}4 \cdot \mathrm{I}_{\mathrm{area}}.$$

3. **Dimensões Relativas**. A largura e altura máxima de um BB ($\mathrm{BB}_{\mathrm{largura}}$ e $\mathrm{BB}_{\mathrm{altura}}$) não deve ser superior a 80% da largura ($\mathrm{I}_{\mathrm{largura}}$) e altura ($\mathrm{I}_{\mathrm{altura}}$) da imagem. Assim,

$$\mathrm{BB}_{\mathrm{largura}} \leq 0{,}8 \cdot \mathrm{I}_{\mathrm{largura}}$$

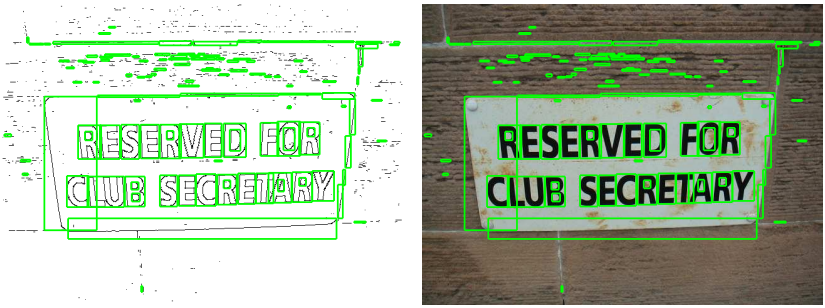
e

$$\mathrm{BB}_{\mathrm{altura}} \leq 0{,}8 \cdot \mathrm{I}_{\mathrm{altura}}.$$

Após a determinação das regiões candidatas a texto, o processo de localização é finalizado e os BBs que delimitam tais regiões são sobrepostos à imagem original. A Fig. 2.18 apresenta o conjunto de BBs da imagem-exemplo que delimitam as regiões consideradas textuais sobrepostos à imagem de bordas [Fig. 2.18(a)] e à imagem original [Fig. 2.18(b)].

2.4 Considerações Finais

O método proposto visa definir as regiões textuais da imagem mediante uma abordagem híbrida, em que a etapa de localização utiliza um método baseado em região e a etapa de verificação (certificação) de tais áreas é realizada por meio de métodos texturais e estruturais, promovendo assim, uma maior



(a) (b)

Fig. 2.18: Resultado da localização. (a) Conjunto de BBs considerados textuais sobrepostos à imagem de bordas. (b) Conjunto de BBs considerados textuais sobrepostos à imagem original.

robustez na localização dos caracteres. O método de localização delimita, na imagem, um conjunto de regiões com propriedades textuais. Essa delimitação das regiões candidatas a texto durante a localização provê uma redução de custo computacional para a etapa posterior de verificação, submetendo apenas tais regiões a uma análise estrutural e textural, classificando-as como textuais ou não-textuais.

O método proposto de localização assume que a magnitude do gradiente de intensidade dos pixels de borda dos caracteres é superior à maioria dos pixels do plano de fundo. Uma média ponderada da magnitude do gradiente de intensidade dos pixels resulta em um limiar L_{M} para cada imagem, sendo este último responsável por selecionar as bordas consideradas textuais. Tal proposta permite a localização individual dos caracteres e reduz a grande quantidade de heurísticas impostas pelos métodos baseados em CCs na determinação das áreas textuais, tais como limiares de alinhamento, espaçamento, orientação, similaridade de dimensões e cores dos caracteres.

Uma vez que a proposta de localização seleciona as bordas candidatas a texto utilizando o gradiente de intensidade dos pixels, o método é capaz de localizar caracteres isolados, além de ser indiferente ao tipo de fonte e dimensões. Devido à mesma razão, o método é independente quanto à orientação, distorção de perspectiva, espaçamento, alinhamento, cor[9], tipo de contraste[10], ascendência e descendência dos caracteres, como ilustrado nas Figs. 2.19 e 2.20. Tal abordagem possui como vantagem adicional a inferior comple-

[9]Caracteres multicoloridos ou palavras em que cada caractere possui uma cor distinta dos demais.

[10]Caracteres mais escuros do que o plano de fundo (contraste positivo) ou mais claros (contraste negativo).

xidade computacional quando comparada aos métodos multi-resolução [39], [41], [66], [67].

É mostrado na Fig. 2.19(a) uma imagem corretamente localizada com caracteres de diferentes dimensões, fontes variadas, possuindo ascendência e descendência. A Fig. 2.19(b) exemplifica a capacidade do método proposto em localizar caracteres com distorção de perspectiva.

Apresenta-se na Fig. 2.20(a) uma *imagem-artificial* com caracteres não alinhados e palavras possuindo diferentes orientações, em que o método proposto obteve êxito devido à proposta de localização individual dos caracteres. A Fig. 2.20(b) ilustra uma *imagem-cena* em que os caracteres da palavra 'ALIEN' possuem um espaçamento entre caracteres maior do que a largura do caractere, o que impossibilitaria muitos métodos baseados em textura (ou densidade de bordas) em localizar corretamente a palavra. Contudo, o método proposto, por basear-se na localização individual dos caracteres, possui independência quanto ao espaçamento.

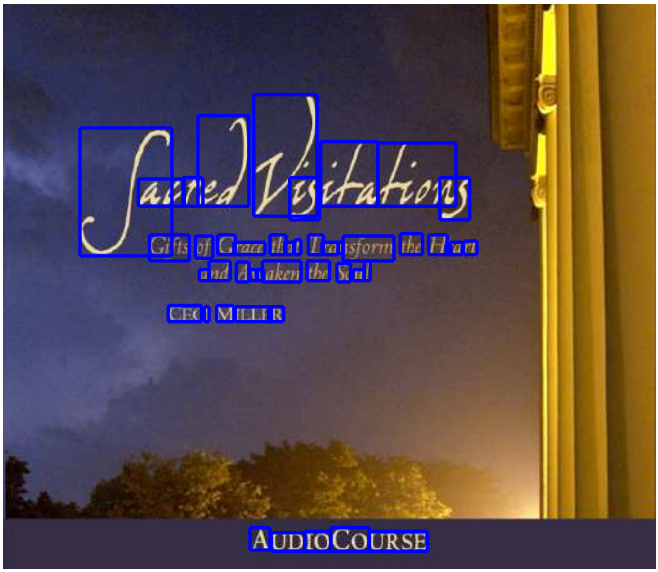
Um outro fator vantajoso é a independência do método às dimensões da imagem de entrada. Diversos métodos de localização baseados em textura ou que utilizam operadores morfológicos são dependentes das dimensões da imagem de entrada [12], [13], [70].

A localização individual dos caracteres ainda permite uma melhoria de desempenho no processo de extração (binarização). Visto que o método proposto delimita os caracteres individualmente e não um conjunto de caracteres (palavras), a região delimitada geralmente possui um plano de fundo mais uniforme como também menor variação de iluminação.

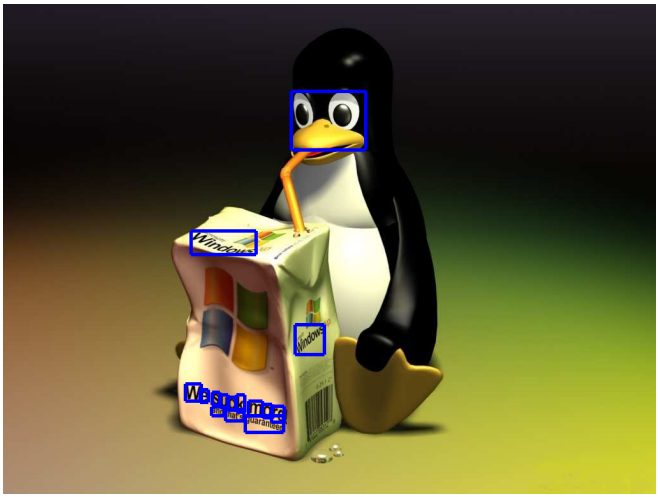
A principal limitação do método proposto está na localização de caracteres sobrepostos a um plano de fundo contendo variação abrupta de cor (bordas evidentes). Tal plano de fundo possui pixels cuja magnitude do gradiente é da mesma ordem dos pixels de borda dos caracteres. Tal fato impossibilita a localização correta devido à junção dos contornos do plano de fundo ao dos caracteres, em que o limiar L_{M} é incapaz de filtrar as bordas evidentes do plano de fundo, como ilustrado na Fig. 2.21. Mostra-se na Fig. 2.21(a) uma imagem contendo os caracteres 'r', 'c', 'w' e 's' (BBs em vermelho) que foram considerados não-textuais por estarem sobrepostos a um plano de fundo com variação abrupta de cor (violeta e preto). Os pixels do plano de fundo no limite entre as cores violeta e preto possuem magnitude do gradiente superior ao limiar L_{M}, como conseqüência, tal borda referente ao plano de fundo não é filtrada. Apresenta-se na Fig. 2.21(b) a ampliação da imagem de bordas na região em que os caracteres não são localizados. Nota-se que houve a fusão das bordas do plano de fundo com as bordas dos caracteres 'r', 'c', 'w' e 's', impossibilitando o algoritmo de verificação (Capítulo 3) em classificar tal contorno como textual.

O limiar L_{M}, utilizado para a seleção das bordas candidatas a texto, pode acarretar a eliminação das bordas de caracteres de pouco contraste com o plano de fundo. Isso ocorre quando, em uma mesma imagem, caracteres de alto e baixo contraste coexistem. A existência de caracteres de alto contraste eleva o valor do limiar L_{M}, ocasionando a eliminação das bordas textuais de menor contraste, como ilustra a Fig. 2.22(a).

O método limita-se a reconhecer caracteres que estão no foco na imagem [Fig. 2.22(b)]. Regiões desfocadas da imagem possuem a energia concentrada nos componentes de baixa freqüência. Sendo assim, os caracteres desfocados apresentam baixa magnitude do gradiente nos pixels de borda (contorno), tornando o método proposto ineficiente em selecionar tais caracteres como regiões candidatas a texto.

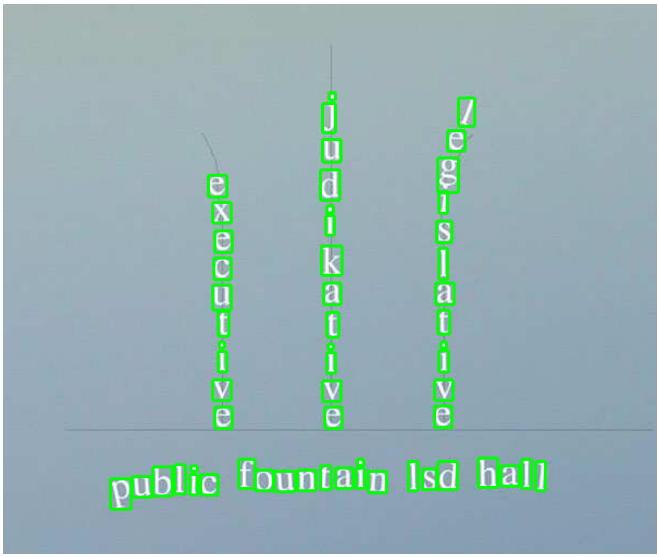


(a)

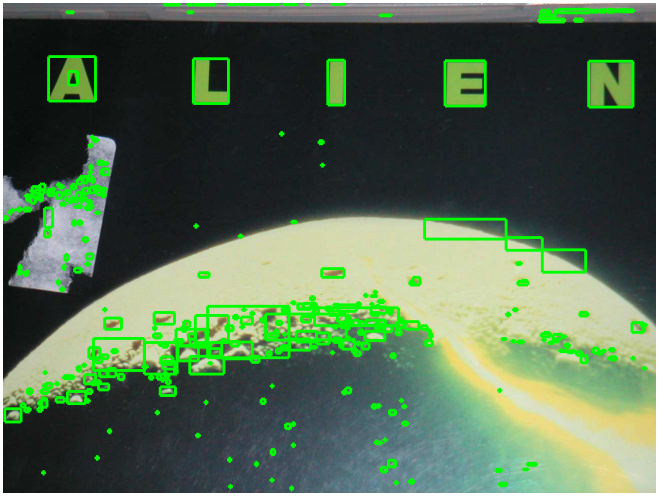


(b)

Fig. 2.19: Imagens exemplificando as vantagens do método proposto de localização (BBs em azul). (a) *Imagem-artificial* com caracteres de fontes variadas, possuindo ascendência, descendência e diferentes dimensões. (b) *Imagem-artificial* possuindo caracteres com distorção de perspectiva.



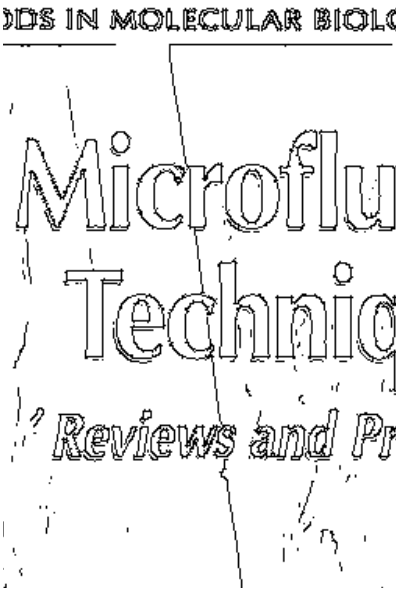
(a)



(b)

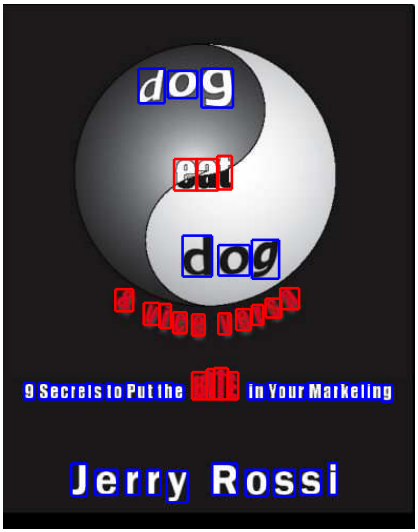
Fig. 2.20: Imagens exemplificando as vantagens do método proposto de localização. (a) *Imagem-artificial* possuindo palavras sem alinhamento e com diferentes orientações (vertical e horizontal). (b) *Imagem-cena* contendo uma palavra com grande espaçamento entre caracteres.

(a)



(b)

Fig. 2.21: Imagens exemplificando as limitações do método proposto de localização. Os caracteres localizados corretamente estão delimitados por BBs em azul, os falsos-positivos em amarelo e os não-localizados em vermelho. (a) Imagem possuindo caracteres sobrepostos a uma borda (diagonal) de plano de fundo de grande contraste. (b) Região ampliada da imagem de bordas produzida pelo método de localização.



(a)

(b)

Fig. 2.22: Imagens exemplificando as deficiências do método proposto de localização. Nesta imagem destacam-se os caracteres não-localizados (BBs em vermelho), os localizados (BBs em azul) e os falsos-positivos (BBs em amarelo). (a) Imagem contendo caracteres de baixo contraste (caracteres em vermelho em um plano de fundo preto) e alto contraste (caracteres em branco em um plano de fundo preto) com o plano de fundo coexistindo na mesma imagem. b) Imagem contendo caracteres fora de foco.

Capítulo 3

Etapa de Verificação

O algoritmo de localização, descrito no capítulo anterior, aproveita a baixa complexidade computacional dos métodos baseados em bordas para delimitar as regiões da imagem candidatas a texto. Uma vez que reduziu-se a imagem a um grupo de regiões candidatas a texto, métodos de maior complexidade computacional, tais como texturais, podem ser utilizados no processo de classificação de tais regiões como textuais ou não-textuais. O objetivo da etapa de *verificação* é classificar as áreas da imagem delimitadas por BBs, durante a etapa de localização, em duas classes: texto e não-texto. Após a classificação, as regiões da imagem rotuladas como não-textuais são descartadas. Sendo assim, a etapa de verificação pode ser vista como uma filtragem refinada das regiões obtidas durante a etapa de localização, aumentando a robustez na determinação das áreas textuais.

A principal deficiência dos métodos texturais de localização é a alta complexidade computacional. O método proposto, em vez de realizar uma análise textural sobre toda a imagem [52], [66], seleciona possíveis regiões textuais por meio de um método baseado em bordas (etapa de localização) e aplica técnicas estruturais e texturais apenas nesse conjunto de regiões, classificando-as como textuais ou não-textuais (etapa de verificação). A utilização de diferentes abordagens promove robustez na identificação das regiões textuais, porém não insere grande aumento de complexidade computacional, visto que os métodos computacionalmente custosos são apenas aplicados às áreas previamente delimitadas na etapa de localização.

A etapa de verificação deste trabalho visa solucionar um problema de

classificação. A classificação é um processo de aprendizado supervisionado[1], em que a partir de um conjunto de exemplos cuja classe é conhecida, treina-se um algoritmo de predição capaz de determinar a classe de exemplos desconhecidos.

Para realizar um aprendizado supervisionado, é necessário possuir um conjunto de exemplos (regiões de imagens) cujas classes (texto ou não-texto) são conhecidas. A esse conjunto de exemplos dá-se o nome de *conjunto de treinamento*. Uma vez de posse do conjunto de treinamento, extraem-se atributos (*features*) de cada região da imagem, tais como densidade de bordas, componentes frequenciais, etc. Essa etapa é comumente conhecida como *extração de atributos (feature extraction)*. No entanto, dentre todos os atributos extraídos, podem existir atributos irrelevantes ou redundantes que podem comprometer o treinamento do algoritmo de predição. Para selecionar o subconjunto de atributos que proporciona o melhor desempenho no aprendizado do algoritmo utiliza-se uma etapa conhecida como *seleção de atributos* [*feature selection (FS)*]. Os atributos selecionados (extraídos do conjunto de treinamento) associados a suas classes correspondentes alimentam o algoritmo de aprendizado. Baseado no conjunto de exemplos (*instances*) e suas classes correspondentes, o algoritmo de aprendizado cria uma regra de decisão capaz de prever exemplos desconhecidos, como mostrado na Fig. 3.1.

Existe uma grande diversidade de métodos de classificação que podem ser categorizados em: *artificial neural network* (ANN); métodos estatísticos; SVM; métodos estruturais e classificadores híbridos.

Desde o final da década de 80, ANN vêm sendo amplamente utilizadas para o reconhecimento de padrões devido à redescoberta do algoritmo *back-propagation* [53], [54] para o treinamento de redes, sendo capazes de classificar distribuições arbitrariamente complicadas. Em meados da década de 90, uma nova direção em reconhecimento de padrões foi introduzida com as máquinas de vetores de suporte (SVM) [16], [64]. A idéia central do algoritmo SVM é transformar os dados de entrada em um espaço de atributos de alta dimensionalidade e separar as classes através de uma superfície de decisão (hiperplano) nesse espaço. Diferentemente das redes neuronais, em que se minimiza o erro sobre o conjunto de dados (risco empírico), SVM é um algoritmo de minimização do risco estrutural [*structural risk minimization* (SRM)].

Para a verificação das regiões obtidas na etapa de localização, escolheu-se o algoritmo SVM devido à minimização do erro de generalização que este proporciona, tornando-o um algoritmo de predição do estado-da-arte para a classificação binária (2 classes). Para uma revisão detalhada do algoritmo SVM fundamentado na otimização de hiperplanos, consultar os trabalhos de

[1] As classificações não supervisionadas são geralmente referenciadas na literatura como clusterização.

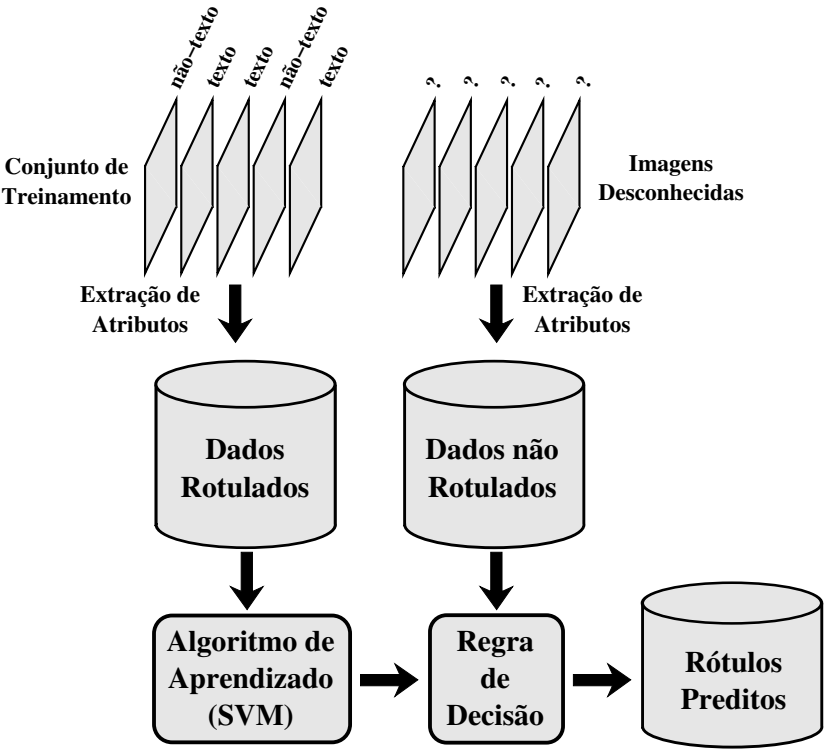


Fig. 3.1: Diagrama de blocos do processo de classificação utilizando SVM.

Vapnik [64] e Cristianini et al. [16].

O restante deste capítulo está organizado como segue. A Seção 3.1 descreve a etapa de extração e avaliação dos atributos utilizados para a classificação das regiões localizadas. A Seção 3.2 explica o método de seleção dos atributos que proporcionam a melhor predição durante a classificação. A Seção 3.3 expõe o processo de treinamento do algoritmo SVM com os atributos selecionados, detalhando a escolha dos parâmetros a esse associados. A Seção 3.4 apresenta os comentários e conclusões deste capítulo.

3.1 Extração de Atributos

O objetivo da extração de atributos é mapear padrões de entrada (regiões da imagem obtidas na etapa de localização) em pontos no espaço de atributos $\mathbb{X}$. Cada ponto no espaço de atributos é representado por um vetor

$\boldsymbol{x} = \{x_1, x_2, x_3, \ldots, x_{\mathrm{n}}\}$, em que n é a dimensão do espaço e cada elemento x_{k} do vetor $\boldsymbol{x}$ é um atributo. Uma vez que os atributos tenham sido extraídos das regiões da imagem, torna-se um problema de classificação (atribuição de um rótulo) do vetor $\boldsymbol{x}$ em texto ou não-texto.

O êxito da classificação está relacionado à capacidade de discriminação entre classes que um atributo ou associação de atributos possui. O classificador resulta em uma predição satisfatória apenas com a utilização de atributos que trazem consigo informações necessárias para a diferenciação entre classes.

Esta seção descreve o conjunto de atributos propostos para a classificação e avalia a capacidade discriminatória de cada atributo agindo individualmente. Para avaliar o desempenho individual de cada atributo, duas interfaces foram criadas: uma para a extração de regiões textuais e não-textuais de imagens, visando compor um banco de imagens rotuladas (Apêndice B.1) e outra para avaliação da capacidade classificatória de cada atributo sobre tais exemplos (Apêndice B.2).

A interface da base de dados visa facilitar o cadastro manual das regiões da imagem obtidas do localizador proposto (Seção 2.3) em 3 classes: classe caractere (C), quando o BB delimita apenas um caractere; classe mais de um caractere (C+), quando o BB delimita mais de um caractere; classe não-caractere (NC), quando o BB delimita uma região não-textual. É de fundamental importância que o algoritmo de localização esteja embutido na interface e seja responsável pela determinação das regiões da imagem que compõem o banco de dados. Isso permite o cadastramento de regiões não-textuais com relevância para o treinamento, uma vez que as regiões não-textuais são obtidas pelo próprio localizador do sistema TIE.

Para as regiões textuais (C e C+) de cada imagem são cadastradas as coordenadas de posição, a orientação dos caracteres[2], uma avaliação subjetiva da qualidade dos caracteres (bom ou ruim) e a *tag*, que são os símbolos que representam os caracteres. A interface ainda permite cadastrar manualmente as regiões textuais não identificadas pelo localizador.

Das regiões não-textuais (NC) coletam-se as coordenadas de posição, índices para aumentar a velocidade de processamento e o tipo de não-caractere (avaliação subjetiva da similaridade do não-caractere com um caractere). Não-caracteres considerados similares recebem o rótulo P, enquanto os não-similares o rótulo NP. A Fig. 3.2 apresenta amostras dos arquivos textos gerados pela interface de cadastramento. A Fig. 3.2(a) representa uma amostra do cadastro de regiões da imagem que pertencem a classe C, enquanto a Fig. 3.2(b) a classe NC. Além de um arquivo texto para cada classe, um arquivo texto adicional cadastra as coordenadas de posição e *tags* das palavras completas da imagem.

[2]Vertical (v), horizontal (h), ou outra direção (o).

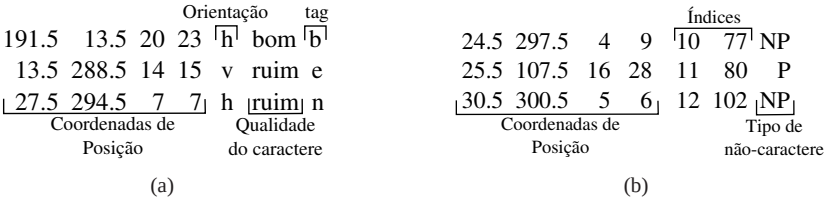


Fig. 3.2: Amostras dos arquivos de texto de cadastramento das regiões da imagem. (a) Amostra do cadastramento em arquivo texto da classe C. (b) Amostra do cadastramento em arquivo texto da classe NC.

A interface de avaliação de atributos extrai o valor de um dado atributo para cada imagem rotulada presente no banco de dados. Plota-se em seguida, para cada classe de imagem (C, C+ e NC), um histograma do atributo. Tais histogramas, plotados sobre um mesmo eixo, permitem avaliar a capacidade individual do atributo em discriminar regiões textuais de não-textuais.

As subseções seguintes apresentam a descrição e avaliação dos atributos propostos, que podem ser categorizados em texturais e estruturais. Os atributos capazes de retirar informações freqüenciais de regiões da imagem caracterizam os atributos texturais. Além disso, cada caractere possui uma estrutura (contorno) que o distingue de muitos objetos não-textuais. Os atributos capazes de extrair informações dos contornos dos objetos são categorizados como estruturais.

3.1.1 Atributos Estruturais

O contorno dos caracteres é composto por um conjunto de pixels que possuem vizinhança-8 e possui características estruturais que se refletem na magnitude e ângulo do gradiente de tais pixels. A magnitude dos pixels dos contornos textuais geralmente é superior a maioria dos pixels dos contornos de outros objetos [43]. Além disso, a distribuição de probabilidade da magnitude e do ângulo do gradiente dos contornos textuais possuem uma relativa diferença aos contornos não-textuais.

Observa-se da Fig. 3.3 que os caracteres possuem um contorno fechado. Clark e Mirmehdi [15] destacam que para cada pixel do contorno do caractere cujo gradiente aponta em uma dada direção, geralmente, existe um outro pixel do contorno do caractere cuja direção do gradiente é aproximadamente a mesma, porém em sentido oposto [Fig. 3.3(b)]. Baseado nessa característica da magnitude e dos ângulos do gradiente, onze atributos estruturais x_k são extraídos dos *pixels de contorno* da imagem:

- A média da magnitude do gradiente - x_1.
- A variância da magnitude do gradiente - x_2.
- O *skewness* da magnitude do gradiente - x_3.
- O *kurtosis* da magnitude do gradiente - x_4.
- A média da direção do gradiente - x_5.
- A variância da direção do gradiente - x_6.
- O *skewness* da direção do gradiente - x_7.
- O *kurtosis* da direção do gradiente - x_8.
- A máxima variação da direção do gradiente - x_9.
- A razão entre o número de pixels de contorno e as dimensões dos BB - x_{10}.
- A razão entre o número de pixels de contorno e a área do BB - x_{11}.

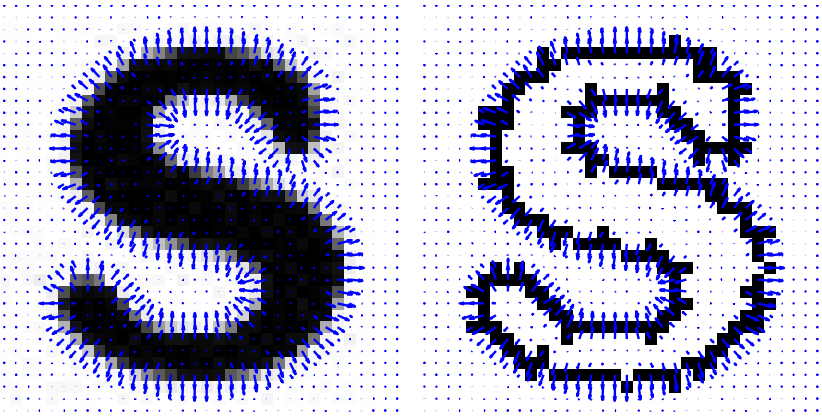
3.1.1.1 Média da Magnitude do Gradiente dos Pixels de Contorno - x_1

Liu et al. [43] identificaram que a média da magnitude do gradiente dos pixels de contorno (borda) dos caracteres é geralmente maior do que a dos outros objetos da imagem. Apresenta-se na Fig. 3.3 o gradiente de cada pixel sobreposto à imagem original e ao contorno correspondente, em que o vetor gradiente de cada pixel está representado por uma seta azul. A magnitude do vetor gradiente em cada pixel é proporcional ao tamanho do segmento da seta. Das Figs. 3.3(a) e (b), notam-se que os pixels de borda dos caracteres apresentam uma magnitude do gradiente maior do que o plano de fundo. Uma vez que a etapa de localização obtém um limiar adaptativo L_{M} para cada imagem e transforma cada contorno considerado textual em um CC, Liu et al. [43] propuseram o atributo de média da magnitude do gradiente dos pixels de contorno, cuja equação é dada por

$$M_{\mathrm{avg}} = \frac{\sum_{(i,j)\in CC} M(i,j)}{n \cdot L_{\mathrm{M}}} \tag{3.1}$$

em que n é o número de pixels do CC[3], L_{M} é o limiar adaptativo obtido para a imagem e $M(i,j)$ é a magnitude do gradiente do pixel (i,j) pertencente ao

[3]Na Fig. 3.3(b), o contorno, representado pelos pixels na cor preta, é um CC da imagem e n é número de pixels do contorno.



(a) (b)

Fig. 3.3: Representação do gradiente sobre os pixels: (a) da imagem original; (b) da imagem de bordas (contornos).

CC. Dessa forma, obtém-se para cada CC um valor M_{avg} que representa a razão entre a média da magnitude do gradiente dos pixels de contorno e o limiar L_{M}. Utilizando o conjunto de imagens rotuladas da base de dados, obtém-se o valor do atributo M_{avg} para cada uma das imagens. Uma vez que sabe-se a classe de cada região de imagem (C, C+, NC), um histograma do atributo para cada classe é obtido, como pode ser observado na Fig. 3.4, em que a abscissa representa o valor do atributo M_{avg} e a ordenada, a porcentagem de regiões que possuem determinado valor do atributo. Observa-se, na Fig. 3.4, que $42{,}81\%$ das regiões pertencentes a classe NC (vermelho) possuem valores para o atributo menores do que $1{,}33$ (região sombreada), enquanto regiões textuais apresentam apenas $0{,}76\%$ e $0{,}72\%$ para C (azul) e C+ (verde), respectivamente. Isso demonstra que o atributo, mesmo atuando individualmente, possui um limiar com capacidade classificatória entre regiões textuais e não-textuais. O atributo média dos pixels do contorno é referenciado no restante do trabalho como x_1.

3.1.1.2 Variância da Magnitude do Gradiente dos Pixels de Contorno - x_2

Os pixels de contorno dos caracteres apresentam, em geral, uma variação de magnitude do gradiente superior aos contornos não-textuais. Isso é resultado da característica estrutural dos contornos dos caracteres, em que

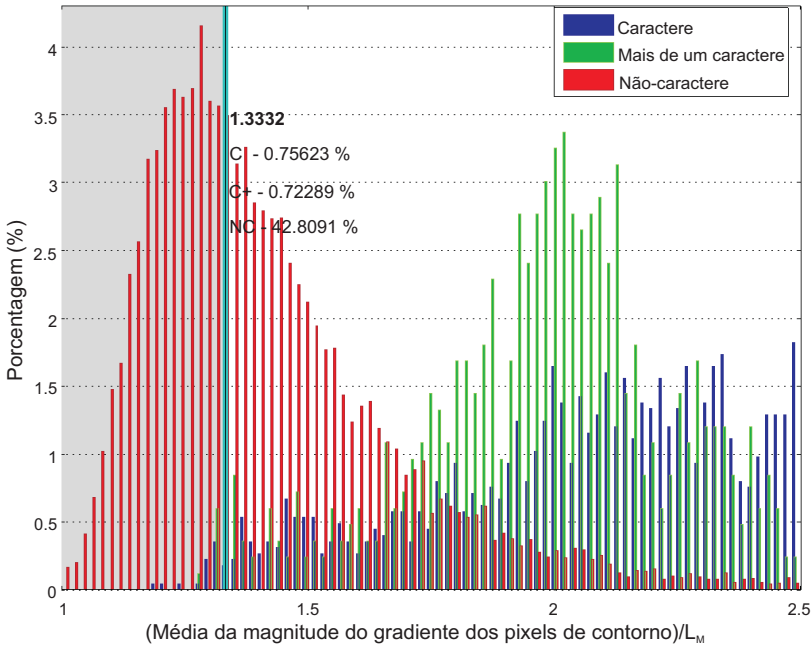


Fig. 3.4: Atributo x_1 - Média da magnitude do gradiente dos pixels de contorno.

esses são compostos por segmentos de reta seguidos de variações abruptas de direção (cantos) (por exemplo, letra 'H' mostrada na Fig. 3.5), ou segmentos de reta acoplados a curvas suaves (por exemplo, letra 'B' ilustrada na Fig. 3.5). Dessa forma, nos contornos textuais, os pixels dos segmentos de reta geralmente possuem maior magnitude do gradiente do que os pixels dos cantos e curvas. Pode-se observar que tais curvas possuem uma variação de cor menos abrupta do que os segmentos de reta, para que o observador não sinta o efeito 'tabuleiro de xadrez'. Como conseqüência, a magnitude do gradiente dos pixels de contorno com tal estrutura possui uma maior variabilidade. A Fig. 3.5(b) exemplifica a maior variância dos contornos textuais. Os pixels cuja magnitude é inferior à metade do desvio padrão da média do contorno são destacados em vermelho, enquanto os que superam duas vezes o desvio padrão estão destacados em azul. Nota-se que cada caractere possui uma quantidade significativa de pixels destacados em vermelho (cantos) e azul (segmentos de reta) relativamente aos contornos não-textuais da parte superior. Tal fato é um

(a) (b)

Fig. 3.5: Exemplo da variação da magnitude sobre contornos textuais e não-textuais. (a) Imagem original contendo caracteres (região inferior da imagem) e elementos não-textuais (região superior da imagem). (b) Imagem destacando os contornos (bordas) extraídos da imagem original. Os pixels destacados em vermelho correspondem aos pixels cujas magnitudes são inferiores à metade do desvio padrão da média do contorno a que eles pertencem, enquanto os pixels destacados em azul, correspondem ao dobro do desvio padrão da média do contorno.

indício de que os contornos textuais possuem uma variância de magnitude do gradiente maior do que os contornos não-textuais.

Utilizando o banco de dados contendo as classes C, C+ e NC, construiu-se um histograma da variância da magnitude do gradiente dos pixels de contorno para cada classe, como mostrado na Fig. 3.6. Observa-se que os contornos da classe NC estão concentrados nas regiões de baixa variância, enquanto quase a totalidade dos contornos textuais (C e C+) apresentam valores de variância acima do limiar que delimita a região sombreada (Fig. 3.6). Os exemplos textuais, por estarem localizados em áreas diferentes das não-textuais no histograma, indicam que o atributo de variância de magnitude do gradiente possui capacidade classificatória mesmo agindo individualmente. No restante do trabalho, tal atributo é referenciado como x_2.

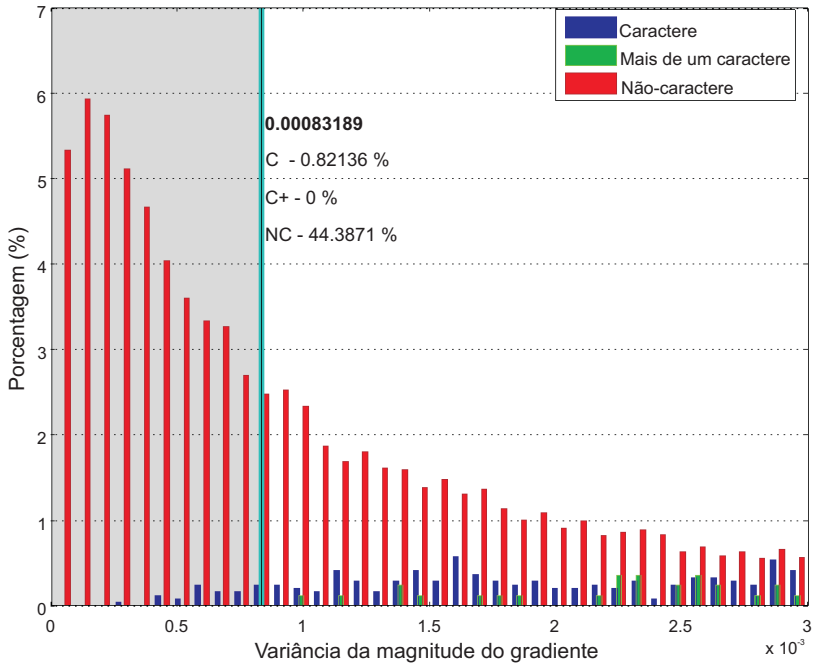


Fig. 3.6: Atributo x_2 - Variância da magnitude do gradiente dos pixels de contorno

3.1.1.3 *Skewness* da Magnitude do Gradiente dos Pixels de Contorno - x_3

O *skewness* é uma medida do grau de assimetria de uma distribuição de probabilidade. Neste trabalho o *skewness* é definido como

$$S = \frac{\mu_3}{\sigma^3} \tag{3.2}$$

onde μ_3 é o terceiro momento central e σ é o desvio padrão. As regiões textuais tendem a possuir uma assimetria na distribuição da magnitude do gradiente dos pixels de contorno. Os pixels localizados onde existe uma variação abrupta da direção do contorno tendem a possuir menor magnitude do gradiente. Uma vez que tais variações geralmente ocorrem em contornos textuais, a distribuição da magnitude dos caracteres tende a possuir uma cauda esquerda mais longa. Tal distribuição possui um *skewness* negativo. Pode-se inferir

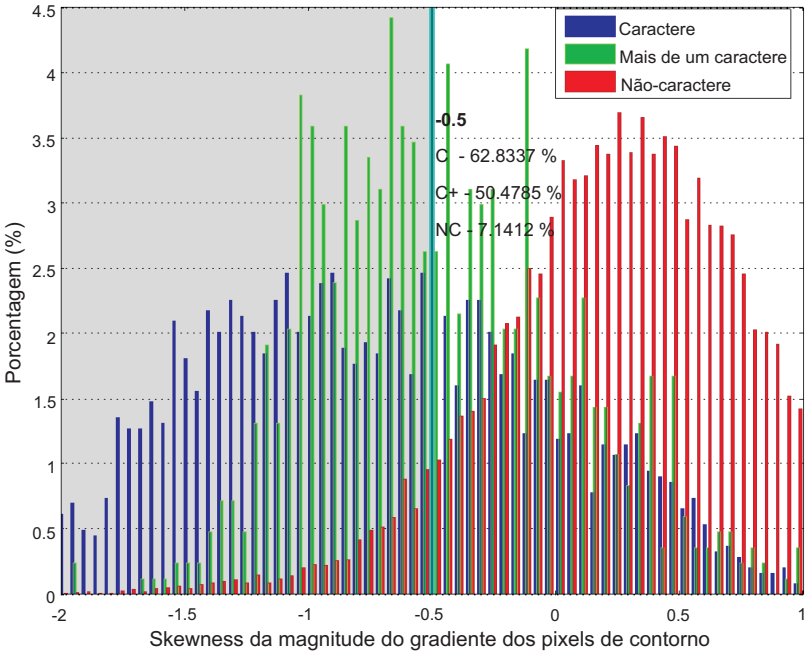


Fig. 3.7: Atributo x_3 - *Skewness* da magnitude do gradiente dos pixels de contorno.

ainda que os contornos não-textuais, por possuírem uma maior predominância de segmentos de reta, possuem uma assimetria cujo *skewness* é positivo. Para avaliar tal hipótese, um histograma de cada uma das classes C, C+ e NC do banco de imagens para o atributo foi realizado, como ilustrado na Fig. 3.7. Nota-se que a maior parte dos contornos textuais possuem valores de *skewness* abaixo de zero (*skewness* negativo), enquanto os contornos não-textuais possuem uma distribuição cujo modo reside na região positiva do *skewness*. Apresenta-se na Fig. 3.7 o histograma de cada classe juntamente com a quantidade de contornos textuais e não-textuais sob a região sombreada, cujo limiar é $-0{,}5$. Nota-se na Fig. 3.7 que existem aproximadamente 62,8% e 50,5% dos contornos textuais (C e C+) sob a região sombreada, enquanto apenas 7,14% dos não-textuais (NC). Tais valores demonstram que o *skewness* da magnitude do gradiente dos pixels de contorno, mesmo atuando individualmente, é um atributo com capacidade discriminatória e é referenciado no restante do trabalho como x_3.

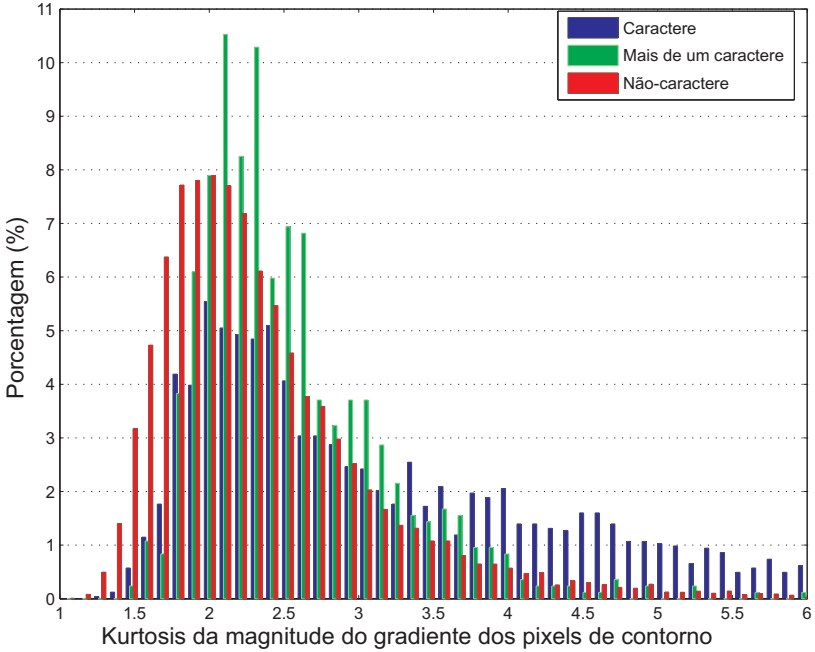


Fig. 3.8: Atributo x_4 - *Kurtosis* da magnitude do gradiente dos pixels de contorno.

3.1.1.4 *Kurtosis* da Magnitude do Gradiente dos Pixels de Contorno - x_4

O *kurtosis* é uma medida de dispersão que caracteriza o 'achatamento' da curva da função distribuição. Neste trabalho utilizamos a definição de *kurtosis* como a razão entre o quarto momento central e o desvio padrão elevada à quarta potência[4]. Assim,

$$S = \frac{\mu_4}{\sigma^4} \tag{3.3}$$

onde μ_4 é o quarto momento central e σ é o desvio padrão. Mesmo não possuindo qualquer evidência de que tal atributo seja significativo, um histograma contendo as classes textuais (C e C+) e não-textuais (NC) foi construído, como mostrado na Fig. 3.8. Observa-se que as três classes estão na mesma região do histograma, evidenciando assim, a incapacidade de discriminação de tal atributo agindo individualmente. No entanto, a incapacidade discriminatória individual não é condição suficiente para tornar o atributo irrelevante, uma

[4]Muitos autores subtraem o valor 3 do *kurtosis* de forma que a distribuição normal possua *kurtosis* 0. No presente trabalho, o *kurtosis* de uma distribuição normal possui valor 3.

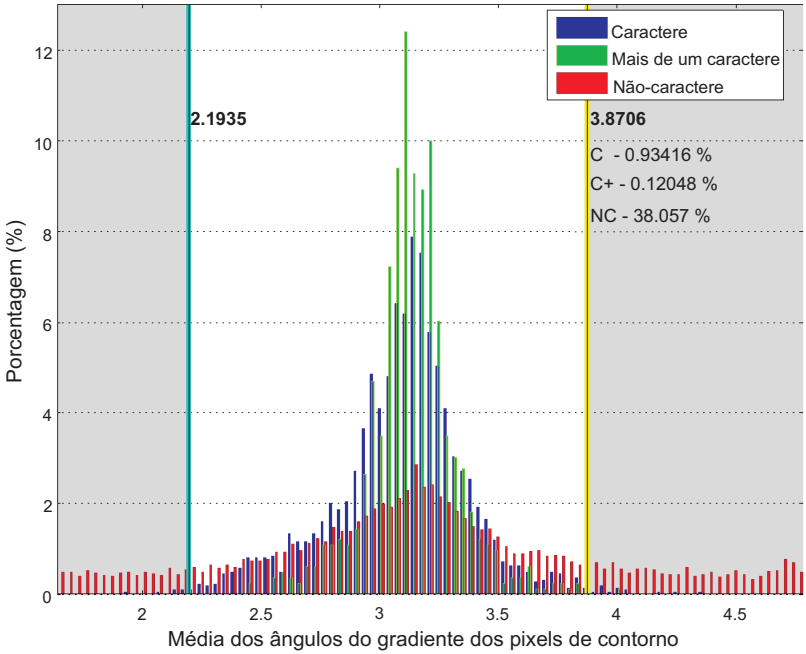


Fig. 3.9: Atributo x_5 - Média do ângulo do gradiente dos pixels de contorno (rad).

vez que tal atributo quando associado a outros, no espaço de atributos, pode contribuir de forma significativa na classificação dos dados. A avaliação de relevância do atributo *kurtosis* da magnitude do gradiente dos pixels de contorno é considerada na Seção 3.2, em que tal atributo é referenciado como x_4.

3.1.1.5 Média dos Ângulos do Gradiente dos Pixels de Contorno - x_5

A média dos ângulos (direção) do gradiente dos pixels de contorno de um caractere, tendem a se concentrar em π rad. Isso se deve ao caminho fechado que percorre um contorno de caractere, em que os ângulos dos vetores gradiente geralmente estão distribuídos entre 0 e 2π rad. Na Fig. 3.9, é apresentado o histograma da média dos ângulos do gradiente dos pixels de borda pertencentes às classes C (azul), C+ (verde) e NC (vermelho). Observa-se da Fig. 3.9 que as regiões textuais (C e C+) concentram-se próximo ao valor π rad, enquanto os contornos de NC possuem uma longa cauda. Isso indica

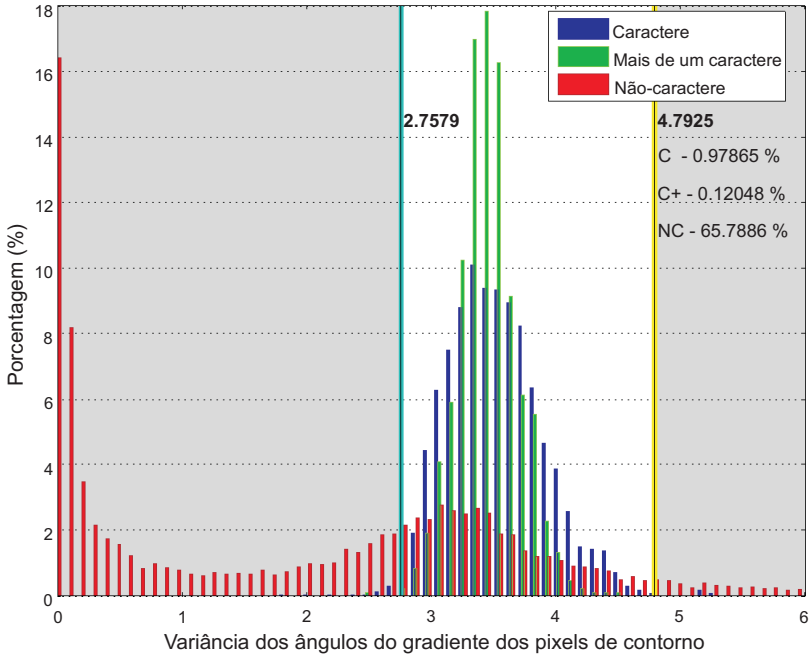


Fig. 3.10: Atributo x_6 - Variância dos ângulos do gradiente dos pixels de contorno.

que os contornos não-textuais possuem uma maior variabilidade em relação ao atributo média dos ângulos dos pixels de contorno. Os histogramas demonstram que, mesmo atuando individualmente, o atributo média dos ângulos do gradiente dos pixels de contorno possui poder classificatório. No restante do trabalho tal atributo é referenciado como x_5.

3.1.1.6 Variância dos Ângulos do Gradiente dos Pixels de Contorno - x_6

As regiões textuais são caracterizadas por possuírem contornos fechados [Fig. 3.3(b)], logo, os ângulos do gradiente dos pixels de contorno percorrem toda a faixa angular de 0 a 2π rad. Devido a tal característica estrutural dos contornos dos caracteres, pode-se inferir que a variância dos ângulos dos pixels de contornos textuais é superior aos contornos não-textuais. É mostrado na Fig. 3.10 o histograma dos valores de variância dos ângulos do gradiente dos pixels de contorno das classes C (azul), C+ (verde) e NC (azul). Nota-se

que o histograma referente à classe NC (vermelho) possui um pico cuja variância está próximo de zero. Tal fato é resultado de contornos não-textuais compostos preponderantemente por um único segmento de reta (situação atípica para contornos textuais), cujo ângulo do gradiente é aproximadamente constante para todos os pixels do contorno. Contudo, os contornos textuais (C e C+) estão concentrados em uma região de alta variância, tornando tal atributo individualmente atraente na discriminação entre texto e não-texto. Tal atributo é referenciado no restante do trabalho como x_6.

3.1.1.7 *Skewness* dos Ângulos do Gradiente dos Pixels de Contorno - x_7

A proposta do *skewness* como um atributo capaz de discriminar alguns contornos textuais de não-textuais parte da simetria existente na distribuição de probabilidade dos ângulos do gradiente dos contornos textuais. Tal simetria existe devido à grande quantidade de pixels cuja direção do gradiente é a mesma, porém em sentidos opostos (Fig. 3.3), como observado por Clark e Mirmehdi [15]. Apresenta-se na Fig. 3.11 o histograma dos ângulos do gradiente dos pixels de dois contornos textuais (lado esquerdo) e dois contornos não-textuais (lado direito). Nota-se que os contornos textuais, diferentemente dos não-textuais, apresentam uma distribuição quase simétrica, indicando que o valor do *skewness* para caracteres textuais deve concentrar-se próximo a zero. Um *skewness* negativo indica que a cauda esquerda da distribuição de probabilidade é maior, enquanto um *skewness* positivo indica uma maior cauda direita. Quando a distribuição é simétrica, o *skewness* é nulo (por exemplo, distribuição normal).

Para avaliar o poder discriminatório individual do atributo *skewness* dos ângulos do gradiente dos pixels de contorno, plotou-se um histograma do valor do *skewness* referente aos contornos das classes C (azul), C+ (verde) e NC (vermelho), como mostrado na Fig. 3.12. Observa-se do histograma da Fig. 3.12 que os contornos textuais apresentam uma distribuição aproximadamente simétrica, visto que os valores do *skewness* estão concentrados próximo ao valor zero. A região sombreada da Fig. 3.12 evidencia que quase a metade dos contornos não-textuais se apresentam nas caudas do histograma, enquanto menos de 2% são de regiões textuais. O atributo *skewness* da direção do gradiente dos pixels de contorno é referenciada no restante do trabalho como x_7.

3.1.1.8 *Kurtosis* dos Ângulos do Gradiente dos Pixels de Contorno - x_8

Os caracteres, de forma geral, possuem contornos cuja direção do gradiente dos seus pixels se distribuem de forma aproximadamente uniforme, como apresentam os histogramas angulares dos caracteres ‘o’ e ‘t’ nas Figs. 3.11(a) e (c). Tal característica indica que a distribuição dos ângulos dos pi-

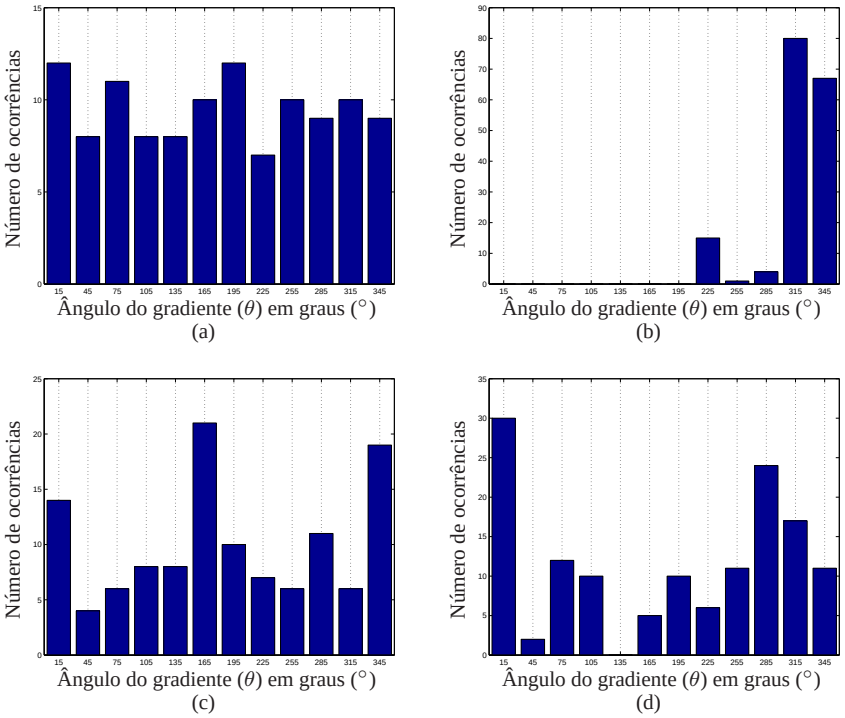


Fig. 3.11: Exemplos de histogramas dos ângulos do gradiente dos pixels de contorno de caracteres e não-caracteres. (a) Histograma do contorno do caractere da letra 'o'. (b) Histograma do contorno de um não-caractere. (c) Histograma do contorno do caractere da letra 't'. (d) Histograma do contorno de um não-caractere.

xels de contornos textuais é mais próxima da distribuição uniforme do que os contornos não-textuais.

Calculando-se o *kurtosis* para cada contorno das regiões da imagem do banco de dados, constrói-se um histograma para cada classe C, C+ e NC, como mostrado na Fig. 3.13. Nota-se, a partir da Fig. 3.13, que as regiões textuais (C e C+) situam-se em torno do valor $1{,}8$ no histograma. Tal fato demonstra que as regiões textuais possuem uma distribuição aproximadamente uniforme para os ângulos do gradiente dos pixels de contorno[5]. No entanto, as regiões não-textuais possuem uma longa cauda à direita no histograma, em que cerca de 41% das regiões da classe NC (região sombreada) possuem um *kurtosis* suficientemente diferente das regiões das classes C e C+, comprovando a ca-

[5]O valor do *kurtosis* para a distribuição uniforme é $1{,}8$.

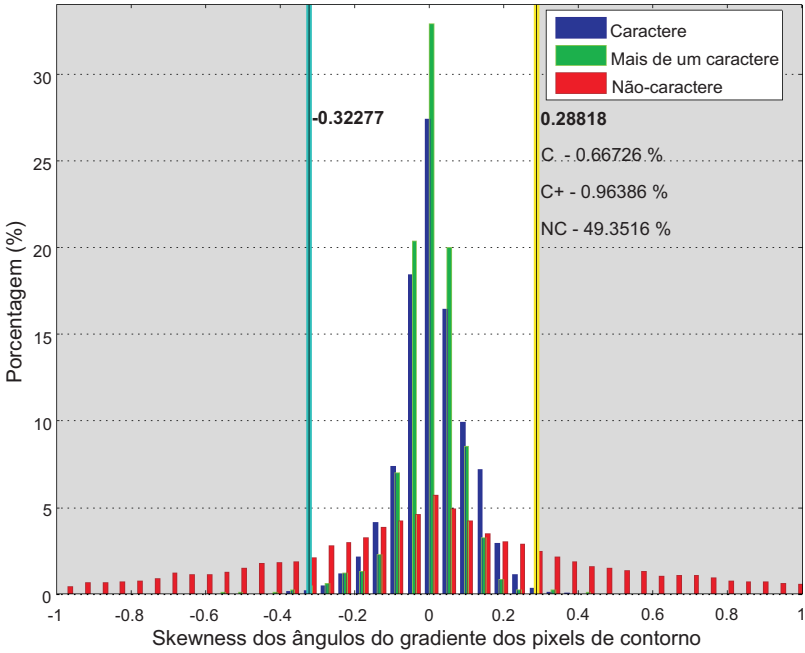


Fig. 3.12: Atributo x_7 - *skewness* da direção do gradiente dos pixels de contorno.

pacidade classificatória do atributo. Tal atributo é referenciado no restante do trabalho como x_8.

3.1.1.9 Máxima Variação da Direção do Gradiente dos Pixels de Contorno - x_9

Um atributo estrutural proposto por Liu et al. [43] é o da máxima variação da direção do gradiente. Uma vez que as bordas dos caracteres são contornos fechados, a máxima diferença da direção do gradiente nos pixels de contorno deve estar próxima de 2π rad. Visto que muitos contornos de objetos não-textuais não percorrem um percurso fechado, a máxima variação da direção do gradiente dos pixels de contorno torna-se capaz de classificar uma grande quantidade de imagens, como ilustrado pelo conjunto de histogramas das classes C, C+ e NC na Fig. 3.14. Tal atributo é referenciado no restante do trabalho como x_9.

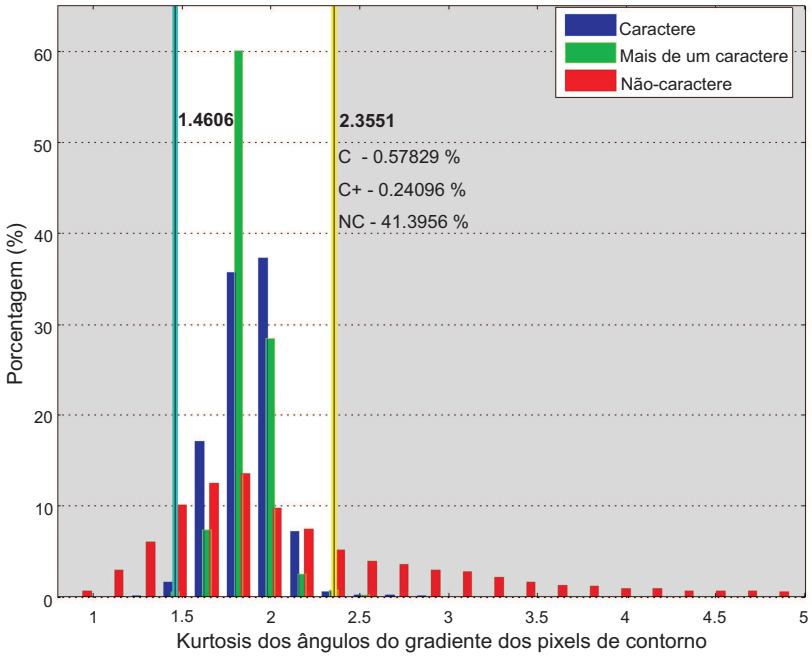


Fig. 3.13: Atributo x_8 - *kurtosis* dos ângulos do gradiente dos pixels de contorno.

3.1.1.10 Razão entre o Número de Pixels de Contorno e as Dimensões dos BBs - x_{10}

Liu et al. [43] observaram que o número de pixels de um contorno textual geralmente supera a máxima dimensão do BB (altura ou largura). Isso se deve ao fato dos caracteres possuírem contornos fechados, levando o contorno a possuir um número de pixels, no mínimo, duas vezes maior do que a máxima dimensão do BB que o delimita. Além disso, os contornos não-textuais geralmente não percorrem um percurso ou são simples segmentos de reta, tornando a comparação da dimensão do BB com o número de pixels de contorno um atributo classificatório. Utilizando o banco de imagens rotuladas, construiu-se um histograma para cada classe, como mostrado na Fig. 3.15. Nota-se, na Fig. 3.15 que apenas 0,12% das regiões textuais estão sob a região sombreada cujo limiar é duas vezes a máxima dimensão do BB que delimita o contorno. No entanto, 59,7% dos contornos não-textuais estão nessa região do histograma, demonstrando que tal atributo possui poder discriminatório entre re-

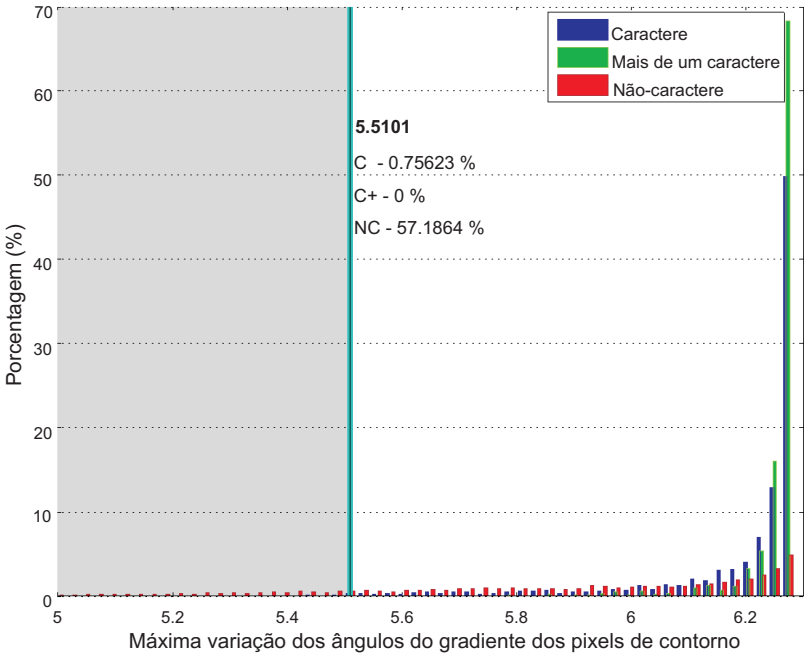


Fig. 3.14: Atributo x_9 - Máxima variação da direção do gradiente dos pixels de contorno (rad).

giões textuais e não-textuais. O atributo é referenciado no restante do capítulo como x_{10}.

3.1.1.11 Razão entre o Número de Pixels de Contorno e a Área do BB - x_{11}

Seguindo a mesma linha de raciocínio do atributo anterior, obteve-se o atributo que considera a razão entre o número de pixels de contorno e a área do BB que o delimita. Um histograma do atributo para cada classe (C, NC e C+) é apresentado na Fig. 3.16. Nota-se que o atributo não promove uma separação nítida entre regiões textuais e não-textuais. No entanto, o modo do histograma da classe C está em uma região diferente do modo do histograma da classe NC. Sendo assim, tal atributo pode ser útil na classificação quando associado a outros atributos. A etapa de seleção de características, apresentada na Seção 3.2, é a responsável por decidir se tal atributo é relevante na

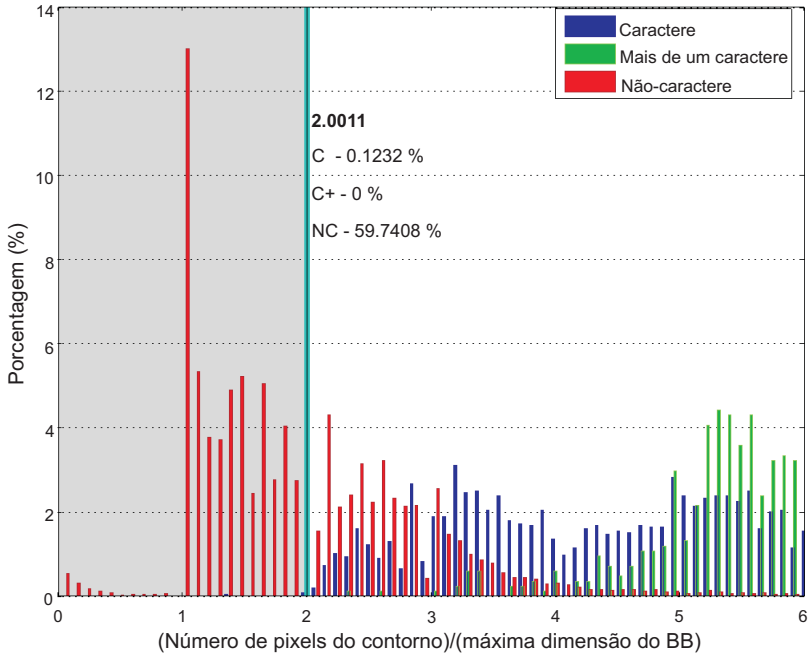


Fig. 3.15: Atributo x_{10} - Número de pixels do contorno/máxima dimensão do BB.

discriminação entre regiões textuais e não-textuais. Tal atributo é referenciado no restante do trabalho como x_{11}.

3.1.2 Atributos Texturais

Todos os atributos texturais utilizados neste trabalho são derivados de informações extraídas da transformada *wavelet*. A análise *wavelet* provê um método de identificação do conteúdo de freqüência espacial da imagem, além da localização do conteúdo frequencial existente. Devido a estas características, a transformada *wavelet* vem sendo bastante explorada como uma ferramenta de discriminação das propriedades texturais entre texto e não-texto. Cinco atributos texturais foram analisados: dois utilizam o desvio padrão dos coeficientes das sub-bandas obtidos da transformada *wavelet* (LH, HL, HH); três baseiam-se na energia dos coeficientes das sub-bandas (LH, HL, HH).

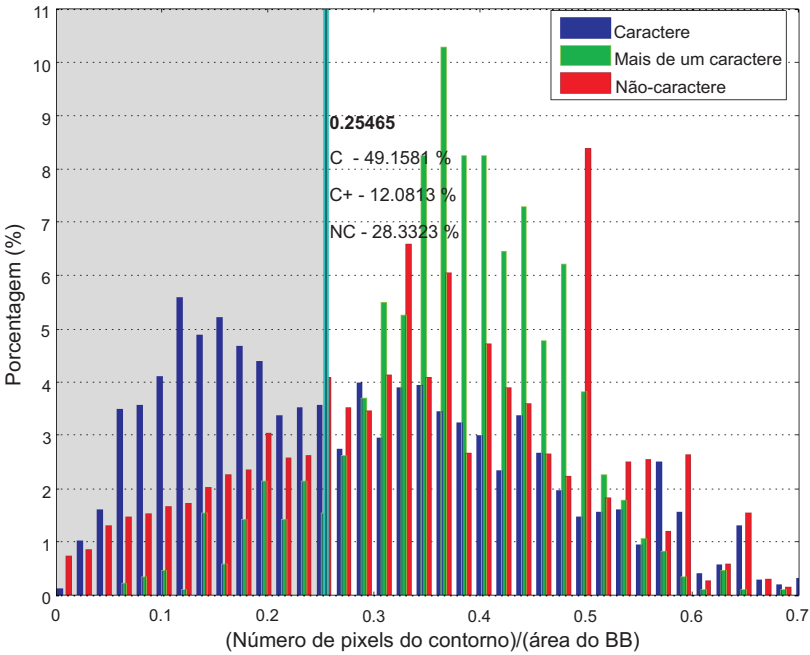


Fig. 3.16: Atributo x_{11} - Razão entre o número de pixels de contorno e a área do BB que o delimita.

3.1.2.1 Desvio Padrão do Histograma dos Coeficientes *Wavelet* (Imagem em níveis de cinza) - x_{12}

Li e Gray [40] observaram que os histogramas dos coeficientes da transformada *wavelet* nas sub-bandas (LH, HL, HH) de regiões não-textuais tendem a possuir uma distribuição Laplaciana, enquanto os coeficientes das regiões textuais são dispersos e concentrados em alguns poucos valores discretos. Baseando-se nessa informação, Gllavata et al. [22] assumiram que o desvio padrão do histograma dos coeficientes *wavelet* em regiões textuais são maiores do que as regiões não-textuais.

O atributo proposto neste trabalho é a norma do vetor de desvio padrão proposto por Gllavata et al. [22]. Tal norma foi criada para obter um atributo escalar contendo informações relacionadas ao desvio padrão das sub-bandas da transformada *wavelet*. Para a obtenção de tal vetor realizam-se os seguintes passos:

- Transformação das regiões localizadas na imagem para níveis de cinza mediante a conversão do espaço RGB em YUV.

- Normalização das regiões localizadas para a altura de 32 pixels utilizando interpolação bilinear.

- Decomposição da imagem em quatro sub-bandas, LL, LH, HL, HH. As três sub-bandas de alta freqüência (HL, LH, HH) possuem valores de coeficientes elevados para variações abruptas de intensidade na direção horizontal, vertical e diagonal, respectivamente. A decomposição *wavelet* da imagem em sub-bandas é obtida utilizando os filtros passa-baixas e passa-altas abaixo:

$$\boldsymbol{F}_{\mathbf{pb}} = [-0{,}176777\ 0{,}353535\ 1{,}06066\ 0{,}353535\ -0{,}176777] \quad (3.4)$$
$$\boldsymbol{F}_{\mathbf{pa}} = [-0{,}353535\ -0{,}707107\ 0{,}353535]. \quad (3.5)$$

- Construção de um histograma dos coeficientes de cada sub-banda da transformada *wavelet* (três histogramas no total).

- Realização do cálculo do desvio padrão referente a cada histograma obtido no item anterior da seguinte forma:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{K}(f(i) - E\{f(i)\})^2}{K-1}}, \quad \begin{cases} f(i) = j & \text{se } H_i > 0 \\ f(i) = 0 & \text{caso contrário} \end{cases} \quad (3.6)$$

  onde H_i é o número de ocorrências no histograma, K é o número de intervalos i do histograma, j é o valor central do intervalo e $E\{f(i)\}$, a média de todos os valores centrais j dos intervalos do histograma.

- Criação do vetor de desvio padrão das sub-bandas da transformada *wavelet*:

$$\boldsymbol{V} = [\sigma_{\mathrm{LH}}\ \sigma_{\mathrm{HL}}\ \sigma_{\mathrm{HH}}]. \quad (3.7)$$

O atributo extraído de cada região localizada na imagem é a norma do vetor $\boldsymbol{V}$ de desvio padrão das sub-bandas da transformada *wavelet*. Um histograma dos valores do atributo para cada uma das classes C, C+ e NC é mostrado na Fig. 3.17. Tal atributo é referenciado no restante do trabalho como x_{12}.

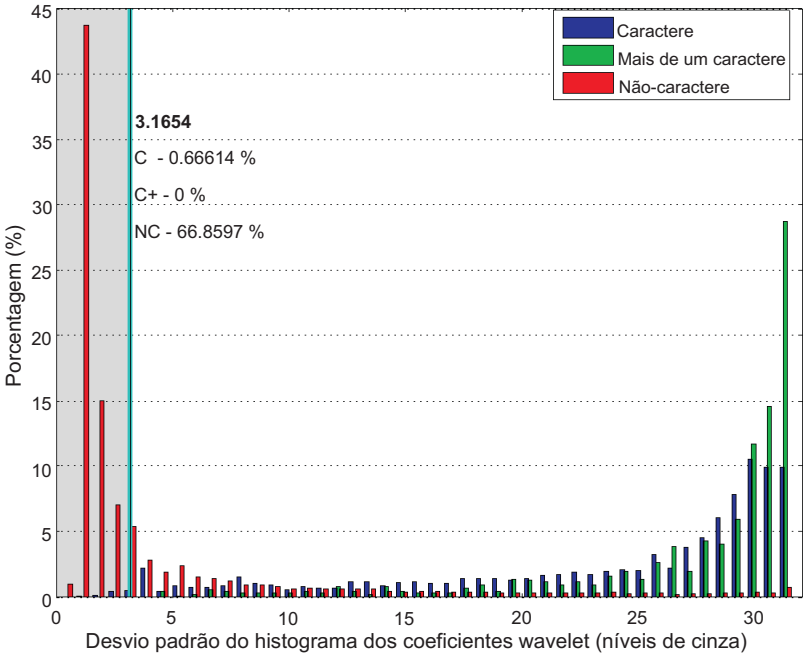


Fig. 3.17: Atributo x_{12} - Desvio padrão do histograma - *Wavelet* (imagem em níveis de cinza).

3.1.2.2 Desvio Padrão do Histograma dos Coeficientes *Wavelet* (Imagem colorida) - x_{13}

O segundo atributo textural é um melhoramento do método de Gllavata et al. [22] proposto por Saoi et al. [55]. Em vez de calcular o desvio padrão dos coeficientes *wavelets* de cada sub-banda da imagem em níveis de cinza (YUV), Saoi et al. propuseram a criação de um vetor de desvio padrão para cada canal da imagem colorida. Uma vez que tal método possui 3 vetores como saída (um vetor para cada canal), adaptou-se o método para este trabalho criando-se um vetor das médias do desvio padrão de cada plano da imagem. Assim,

$$\boldsymbol{V_{RGB}} = [\sigma_{\mathrm{RGB(LH)}}\ \sigma_{\mathrm{RGB(HL)}}\ \sigma_{\mathrm{RGB(HH)}}] \tag{3.8}$$

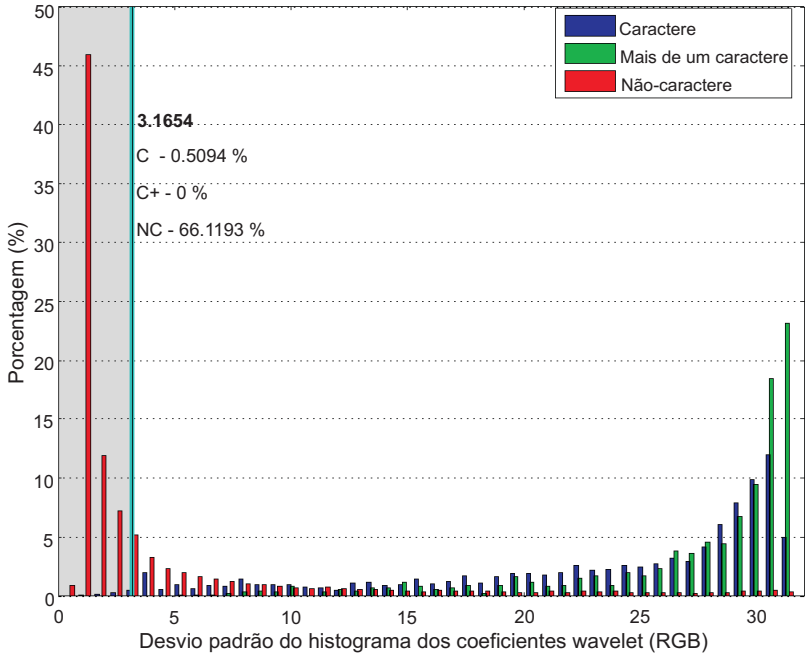


Fig. 3.18: Atributo x_{13} - Desvio padrão do histograma - *wavelet* RGB.

com

$$\sigma_{\mathrm{RGB(sub-banda)}} = \frac{\sigma_{\mathrm{planoR(sub-banda)}}}{3} + \\ + \frac{\sigma_{\mathrm{planoG(sub-banda)}}}{3} + \\ + \frac{\sigma_{\mathrm{planoB(sub-banda)}}}{3}. \tag{3.9}$$

O segundo atributo textural, então, é dado pela norma do vetor $\boldsymbol{V}_{\mathbf{RGB}}$. O histograma do atributo para cada classe pode ser visto na Fig. 3.18. Tal atributo é referenciado no restante do trabalho como x_{13}.

Devido à similaridade da informação extraída pelos atributos x_{12} e x_{13}, esses são redundantes, o que pode degradar a precisão de muitos classificadores se utilizados em conjunto durante o treinamento. Sendo assim, torna-se fundamental a utilização de um seletor de atributos que defina qual o melhor subconjunto de atributos para a melhoria do treinamento do classificador.

3.1.2.3 Momentos Centrais da Energia dos Coeficientes *Wavelet* - x_{14}, x_{15} e x_{16}

As regiões textuais da imagem são caracterizadas por uma seqüência de variações abruptas de intensidade (ou cor) ocasionadas pelas diferenças existentes entre os contornos dos caracteres e o plano de fundo (Fig. 1.5). Tais variações abruptas são responsáveis pelo povoamento das sub-bandas de alta freqüência da transformada *wavelet* com coeficientes de maior magnitude. Dessa forma, infere-se que os momentos centrais da energia contidos nos coeficientes *wavelet* nas sub-bandas de alta freqüência das regiões textuais diferem das regiões não-textuais. Para verificar tal hipótese, neste trabalho, cada região da imagem obtida da etapa de localização é normalizada em 32 pixels de altura utilizando a interpolação bilinear e decomposta utilizando a *wavelet* Haar de primeiro nível em 4 sub-bandas (LL, LH, HL e HH), como proposto por Li et al. [39]. Obtidas as sub-bandas, gera-se uma imagem de energia local $\mathcal{I}_{\text{energia}}(x,y)$ somando-se os valores absolutos dos coeficientes correspondentes de cada sub-banda de alta freqüência (LH, HL e HH) [43].

Três atributos propostos por Li et al. [40] foram extraídos da imagem de energia local ($\mathcal{I}_{\text{energia}}(x,y)$): os momentos de primeira, segunda e terceira ordens. Visto que a decomposição de primeiro nível da *wavelet* Haar sobre uma imagem de $N \times 32$ pixels gera uma imagem $N/2 \times 16$ para cada sub-banda, a soma dos coeficientes correspondentes de cada sub-banda também resulta em uma imagem $N/2 \times 16$. Dessa forma, pode-se obter os 3 momentos necessários como segue:

$$M(\mathcal{I}_{\text{energia}}) = \frac{1}{\frac{N}{2} \cdot 16} \sum_{x=1}^{\frac{N}{2}} \sum_{y=1}^{16} \mathcal{I}_{\text{energia}}(x,y) \tag{3.10}$$

$$M_2(\mathcal{I}_{\text{energia}}) = \frac{1}{\frac{N}{2} \cdot 16} \sum_{x=1}^{\frac{N}{2}} \sum_{y=1}^{16} \left[\mathcal{I}_{\text{energia}}(x,y) - M(\mathcal{I}_{\text{energia}})\right]^2 \tag{3.11}$$

$$M_3(\mathcal{I}_{\text{energia}}) = \frac{1}{\frac{N}{2} \cdot 16} \sum_{x=1}^{\frac{N}{2}} \sum_{y=1}^{16} \left[\mathcal{I}_{\text{energia}}(x,y) - M(\mathcal{I}_{\text{energia}})\right]^3 . \tag{3.12}$$

Para avaliar a capacidade discriminatória dos atributos M, M_2 e M_3, constrói-se um histograma para cada classe (C, C+ e NC), como mostrado nas Figs. 3.19, 3.20 e 3.21.

Nota-se que os atributos M, M_2 e M_3, referenciados no restante do trabalho por x_{14}, x_{15} e x_{16}, respectivamente, possuem poder classificatório

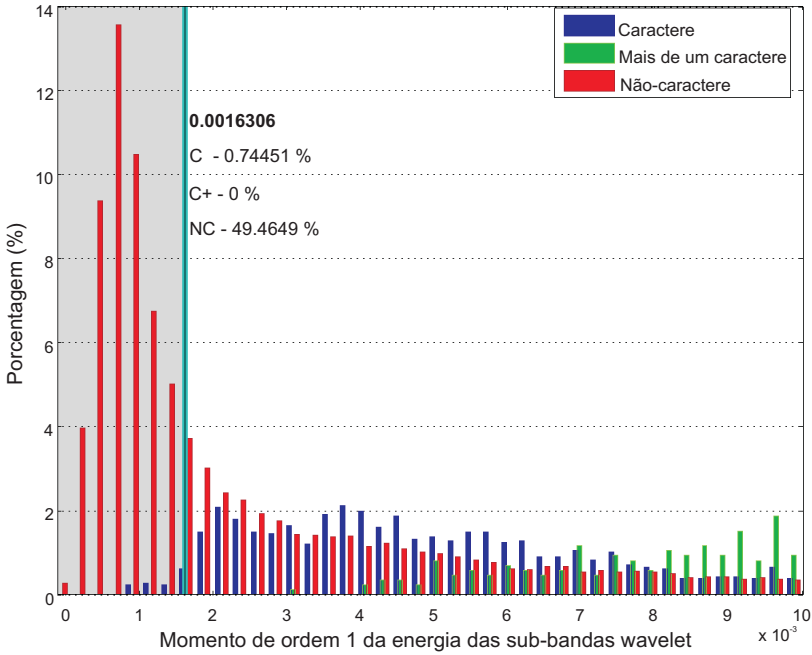


Fig. 3.19: Atributo x_{14} - Momento de primeira ordem das sub-bandas (LH, HL e HH) da transformada *wavelet* (Haar).

e devem compor o conjunto de atributos X para posterior seleção do melhor subconjunto de atributos.

3.1.3 Formação do Vetor de Atributos

Uma vez extraído, de cada região localizada, o conjunto de atributos $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}\}$, pode-se representar cada região localizada no espaço de atributos $\mathbb{X}$ por meio de um vetor $\boldsymbol{x} = [x_1, \ldots, x_{16}]^T$ de atributos.

O vetor de atributos $\boldsymbol{x}$, com o respectivo rótulo, é utilizado como entrada para o treinamento da máquina de aprendizado (Fig. 3.1). No entanto, há a possibilidade de um subconjunto Q de atributos proporcionar um melhor treinamento para o algoritmo de aprendizado, visto que atributos irrelevantes ou redundantes podem estar contidos no conjunto X. Visando encontrar o subconjunto Q capaz de prover uma melhor predição para o classificador, reali-

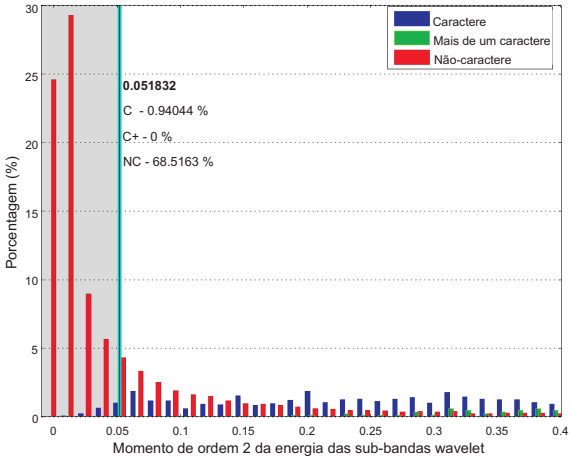


Fig. 3.20: Atributo x_{15} - Momento de segunda ordem das sub-bandas (LH, HL e HH) da transformada *wavelet* (Haar).

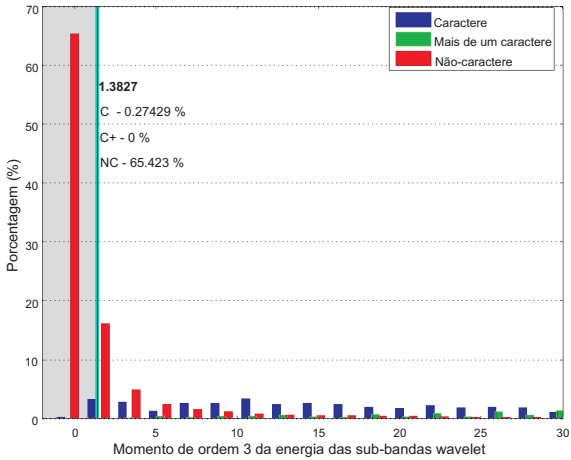


Fig. 3.21: Atributo x_{16} - Momento de terceira ordem das sub-bandas (LH, HL e HH) da transformada *wavelet* (Haar).

zou-se uma seleção dos atributos que está descrita detalhadamente na próxima seção.

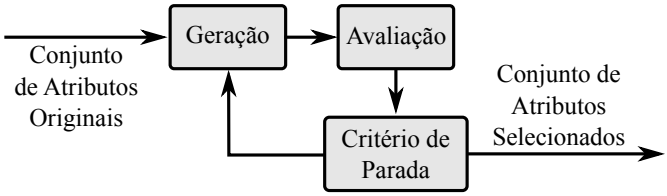


Fig. 3.22: Diagrama de blocos geral dos algoritmos de seleção de atributos.

3.2 Seleção de Atributos

Os atributos obtidos na seção anterior são extraídos de cada região identificada na etapa de localização. O conjunto de atributos extraídos das regiões localizadas, rotuladas como texto ou não-texto, formam um conjunto de dados. Tal conjunto de dados necessita de um pré-processamento para o aprendizado de máquina (*machine learning*), cujo objetivo é transformar dados em *conhecimento*.

A *feature selection* (FS) - é uma técnica de pré-processamento com o objetivo de selecionar, dentre todos os atributos, um subconjunto capaz de melhorar a predição ou diminuir a dimensão do vetor de atributos sem comprometer o seu desempenho.

Os principais propósitos da seleção de atributos são: redução de dimensionalidade, remoção de atributos irrelevantes[6] e redundantes[7], redução da quantidade de dados necessária ao aprendizado, melhoria do desempenho dos algoritmos preditivos (precisão) e aumentar a compreensão dos modelos obtidos.

Segundo Dash e Liu [18] pode-se modelar os métodos de seleção de atributos em três etapas (Fig. 3.22):

1. Geração.

2. Avaliação.

3. Critério de parada.

A etapa de *geração* é a responsável pelo procedimento de busca. Assim, tal etapa gera o subconjunto de atributos para a etapa de avaliação. O processo de geração pode iniciar: (i) com nenhum atributo; (ii) com todos os

[6]Existem diversos conceitos de relevância [36]. O conceito utilizado neste trabalho é: um atributo x_i é relevante se a probabilidade da classe é alterada quando eliminamos o conhecimento do valor de x_i.

[7]Um atributo é redundante se ele possui alto grau de correlação com outro(s) atributo(s).

atributos; ou (iii) com um subconjunto aleatório de atributos. Nos primeiros dois casos, atributos são iterativamente adicionados (*forward selection*) ou removidos (*backward elimination*), enquanto no último caso os atributos podem ser iterativamente adicionados, removidos ou aleatoriamente selecionados.

A etapa de *avaliação* destina-se, como o próprio nome indica, à avaliação da qualidade de um determinado subconjunto de atributos obtido de algum procedimento de geração.

O *critério de parada* define a regra que determina o término da seleção de atributos, evitando que o processo de busca seja exaustivo ou torne-se permanente dentro do espaço de subconjuntos. O critério de parada geralmente está relacionado à etapa de *geração* ou *avaliação*. Os critérios quanto à etapa de *geração* mais comuns atuam: (i) se um determinado número de atributos é alcançado; (ii) se um número pré-definido de iterações é alcançado. Quanto à etapa de avaliação, o critério de parada pode atuar: (i) se a adição ou eliminação de um atributo não produz um melhor subconjunto; (ii) se algum subconjunto ótimo é obtido de acordo com alguma função de avaliação.

Os algoritmos de seleção de atributos podem ser classificados em: *filtros*, *wrappers* e *embutidos* (*embedded*). Os *filtros* selecionam um subconjunto de atributos baseado na maximização de algum critério de relevância. A seleção é feita completamente independente do classificador a ser utilizado posteriormente. Apesar de possuir um custo computacional inferior às outras duas abordagens, os *filtros* ignoram completamente os efeitos dos atributos selecionados no desempenho do classificador. Os *wrappers* são algoritmos que selecionam um subconjunto de atributos "empacotando" o classificador na etapa de avaliação, ou seja, o classificador é utilizado como parte do processo de avaliação, selecionando, assim, os atributos que proporcionam o melhor desempenho para um classificador específico. Os métodos *embutidos* são aqueles cuja seleção do subconjunto de atributos está embutida no algoritmo de indução.

3.2.1 Seleção de Atributos Proposta

A escolha da abordagem a ser utilizada na seleção de características (*filtros*, *wrappers* ou *embedded*) é um passo fundamental para o desempenho do classificador. Os *filtros* possuem como vantagem a baixa complexidade computacional e a independência da seleção de atributos em relação ao classificador. Todavia, os filtros baseiam a seleção de acordo com algum critério, em que o subconjunto que maximiza tal critério não necessariamente maximiza o desempenho do classificador ao utilizar o subconjunto selecionado. Além disso, vários seletores baseados em filtros não avaliam a interação entre os atributos no poder de classificação; alguns apenas selecionam os atribu-

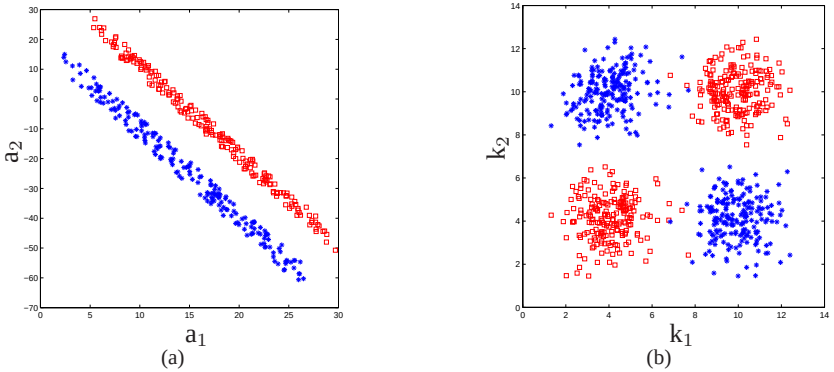


Fig. 3.23: Atributos irrelevantes com capacidade classificatória quando associados a outros atributos.

tos de acordo com a correlação entre esses e as classes. Como conseqüência, pode ocorrer a seleção dos k melhores atributos e esses não corresponderem ao melhor conjunto de k atributos para a classificação devido à:

1. Redundância de atributos.

2. Eliminação de atributos considerados irrelevantes, porém com capacidade classificatória quando associados a outros atributos [24], como ilustrado na Fig. 3.23.

As Figs. 3.23(a) e (b) apresentam dados contendo duas classes, representadas por quadrados vermelhos e asteriscos azuis. Da Fig. 3.23(a), observa-se que os dados de classes diferentes se sobrepõem quando projetados sobre um dos atributos (a_1 ou a_2). Caso a avaliação de relevância do atributo seja a discriminância entre as classes (ou a informação mútua entre o atributo e a classe) produzida por cada atributo individualmente, os dois atributos (a_1 e a_2) seriam considerados irrelevantes e eliminados. Contudo, percebe-se que os atributos associados possuem margem de separação entre as classes. O mesmo ocorre com os atributos k_1 e k_2 na Fig. 3.23(b).

A abordagem de seleção de atributos *wrapper* possui como objetivo direto a minimização do erro de classificação sobre um classificador específico. Devido a tal característica, geralmente, *wrappers* resultam em um conjunto de atributos de alto poder classificatório (classificador específico) ao custo de uma alta complexidade computacional e perda de generalidade dos atributos escolhidos para atuação em outros classificadores. Essa perda de generalidade indica que o subconjunto de atributos selecionado pela abordagem *wrapper*

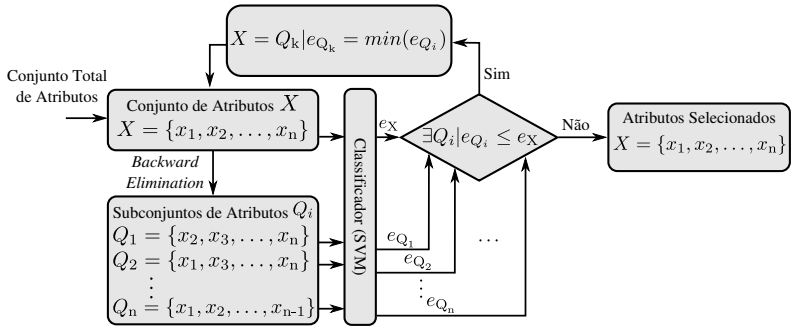


Fig. 3.24: Diagrama de blocos da abordagem *wrapper* utilizando a técnica *backward elimination* para a seleção de atributos.

pode resultar em um pior desempenho quando utilizados com outros classificadores.

Neste trabalho, a abordagem *wrapper* foi escolhida para a seleção de atributos devido:

- ao pequeno número de atributos, tornando viável a utilização da abordagem *wrapper* do ponto de vista de complexidade computacional;
- à minimização do erro de classificação sobre um classificador específico, visto que, neste trabalho, tal conjunto de atributos é utilizado apenas pelo classificador SVM.

Uma vez definida a abordagem *wrapper* de seleção de atributos, utiliza-se como procedimento de busca (geração) o método *backward elimination*, por possuir maior poder de captura da interação entre atributos do que a busca *forward* [36]. Apesar do algoritmo SVM possuir um seletor de atributos embutido, a seleção de atributos para o treinamento é necessária devido à perda de precisão na predição do classificador SVM quando existem atributos irrelevantes ou redundantes [65].

O método de seleção proposto é inicializado com o conjunto de todos os atributos, como apresentado na Fig. 3.24. Inicialmente, X é o conjunto total de atributos e, para tal conjunto, o erro de classificação utilizando o classificador SVM é e_X. A técnica *backward elimination* gera n subconjuntos Q retirando um atributo por vez do conjunto X (Fig. 3.24). Cada subconjunto de atributos Q_i é então utilizado pelo classificador SVM e os seus correspondentes erros de classificação e_{Q_i} são computados. Caso exista algum Q_i cujo $e_{Q_i} \leq e_X$, tal subconjunto Q_i é considerado melhor do que o conjunto X. O subconjunto Q_i então substitui o conjunto X e todo o processo é reinici-

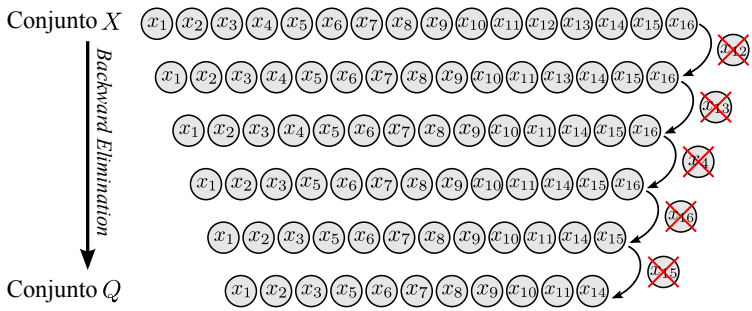


Fig. 3.25: Seqüência dos subconjuntos de atributos obtidos durante o processo de seleção de atributos utilizando a técnica *backward elimination*.

alizado. Tal substituição ocorre porque o subconjunto Q_i, além de possuir menos atributos do que o conjunto X, possui indícios de melhoria na precisão do preditor SVM. Caso não exista um subconjunto Q_i cujo $e_{Q_i} \leq e_{\mathrm{X}}$, o processo de seleção de atributos é finalizado e o conjunto X é considerado o melhor conjunto de atributos para a classificação (Fig. 3.24).

Utilizando a seleção de atributos proposta, 11 entre os 16 atributos são selecionados. A Fig. 3.25 apresenta a seqüência de subconjuntos escolhidos durante o processo de eliminação de atributos (*backward elimination*), resultando na escolha do subconjunto de atributos Q_{final}. Assim,

$$Q_{\mathrm{final}} = \{x_1,x_2,x_3,x_5,x_6,x_7,x_8,x_9,x_{10},x_{11},x_{14}\}.$$

Uma vez obtido um subconjunto de atributos de menor dimensionalidade e de maior relevância para o treinamento, cada imagem do banco de dados, com o rótulo correspondente (texto ou não-texto), pode ser representada por um vetor $\boldsymbol{x}$ de atributos selecionados. Esse conjunto de atributos alimenta o algoritmo de aprendizado SVM e o capacita a criar uma regra de decisão capaz de predizer rótulos a exemplos desconhecidos de imagens, como apresentado na Fig. 3.1.

3.3 Treinamento do Classificador SVM

O treinamento adequado dos dados é de fundamental importância para que a regra de decisão criada pelo algoritmo de aprendizado generalize adequadamente para exemplos desconhecidos. Para permitir alguma flexibilidade na separação das classes, os modelos SVM possuem um parâmetro de custo (ou termo de regularização) C, que controla o compromisso entre permitir erros de treinamento e forçar margens rígidas. Tal parâmetro cria uma margem

flexível que permite alguns erros de classificação. O aumento do valor de C aumenta o custo de exemplos classificados incorretamente, forçando a criação de um modelo mais complexo podendo não generalizar adequadamente os exemplos desconhecidos.

O primeiro passo para o treinamento do classificador SVM é a escolha do *kernel* que mapeia o conjunto de atributos em um espaço $\mathbb{F}$ de maior dimensionalidade. Nesse espaço, a separação de dados via funções lineares é possível. Os *kernels* mais utilizados são o linear, polinomial, sigmoidal e o *radial basis function* (RBF), definidos, respectivamente, como segue:

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = \boldsymbol{x_i}^T \cdot \boldsymbol{x_j} \quad (3.13)$$

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = (r + \gamma \cdot \boldsymbol{x_i}^T \cdot \boldsymbol{x_j})^d \quad (3.14)$$

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = \tanh(\boldsymbol{x_i}^T \cdot \boldsymbol{x_j} - \theta) \quad (3.15)$$

$$K(\boldsymbol{x_i},\boldsymbol{x_j}) = e^{\left(-\gamma \cdot ||\boldsymbol{x_i} - \boldsymbol{x_j}||^2\right)} \quad (3.16)$$

Os símbolos r, d e γ são parâmetros dos *kernels*, enquanto $\boldsymbol{x}^T$ representa o transposto do vetor $\boldsymbol{x}$.

Para este trabalho, escolheu-se o *kernel* RBF, visto que esse possui o melhor desempenho de classificação entre regiões textuais e não-textuais sobre um conjunto de dados reduzido [11]. Tal afirmação pode ser explicada pela capacidade do *kernel* em mapear não-linearmente os dados em um espaço de maior dimensão, obtendo sucesso mesmo quando a relação entre as classes e os atributos é não-linear.

Para o *kernel* RBF deve-se encontrar o melhor conjunto de parâmetros que gere uma regra de decisão generalista durante o treinamento. Um problema a ser evitado durante o treinamento e que prejudica a generalidade da regra de decisão é o *overfitting*, em que o algoritmo de aprendizado possui um bom desempenho de classificação para o conjunto de treinamento, porém um desempenho abaixo do esperado para exemplos desconhecidos, como apresentado na Fig. 3.26. Na tentativa de otimizar os parâmetros[8] visando obter melhor desempenho de predição para o conjunto de treinamento, pode-se obter um modelo preditivo muito complexo, capaz de classificar bem o conjunto de treinamento, no entanto, não obter um modelo generalista para um conjunto desconhecido de dados.

Uma solução para evitar o *overfitting* é utilizar uma técnica denominada *M-fold cross-validation* para avaliar o desempenho de um conjunto de parâmetros determinado pelo projetista. Tal técnica particiona o conjunto de treinamento em M subconjuntos iguais, em que $M - 1$ são utilizados para o treinamento e o restante para a validação do modelo (*validation set*). Tal

[8] C e γ do *kernel* RBF.

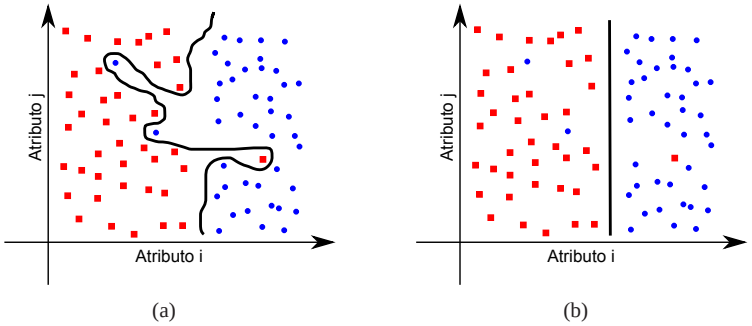


Fig. 3.26: Exemplo da perda de generalidade devido ao *overfitting*. (a) Modelo de predição de maior taxa de acerto sobre o conjunto de treinamento, porém com ocorrência de *overfitting*. (b) Modelo com melhor desempenho para exemplos desconhecidos, visto que é mais generalista.

processo é realizado M vezes seqüencialmente, até que todos os subconjuntos tenham sido utilizados uma vez para a validação. A estimação do desempenho do conjunto de parâmetros (C e γ) para predições futuras é dado pela média da taxa de acerto sobre os subconjuntos de validação.

Visando criar um modelo generalista, a busca pelos melhores parâmetros C e γ (relacionado ao *kernel* RBF) [64] são obtidos da seguinte forma:

1. Transformação linear da faixa de valores de cada atributo para o intervalo [0,1] (*scaling*).

2. Busca dos melhores parâmetros C e γ por meio do método *grid-search*. Tal método é caracterizado por uma busca exaustiva do melhor par de parâmetros variando-se seqüencialmente os valores do par e avaliando-se o desempenho, como apresentado na Fig. 3.27.

3. Para cada par de parâmetros (C,γ) obtido pelo método *grid-search*, é efetuado um treinamento do conjunto de dados utilizando a técnica *10-fold cross-validation*. O par (C,γ) que apresentar o melhor desempenho é utilizado para o treinamento final do conjunto de dados.

Para o treinamento do algoritmo SVM foi selecionado um conjunto de 92 imagens, cujas dimensões (largura $\times$ altura) variavam entre 588 $\times$ 138 e 4500 $\times$ 1500. Tal conjunto é composto por 49 *imagens-artificiais* (24 capas de livro e 25 capas de *compact disc* (CD)) e 43 *imagens-cena* (imagens do campeonato de localização textual realizado pelo ICDAR em 2003). O conjunto de imagens possui 2006 regiões textuais entre as classes C e C+. As dimensões das fontes dos caracteres nesse conjunto de imagens variam entre 4 $\times$ 5 e 275 $\times$ 280, provendo o conjunto de treinamento com uma ampla variedade de

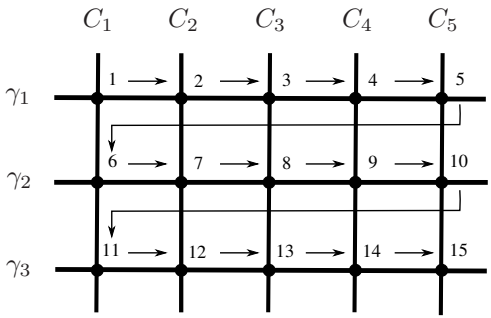


Fig. 3.27: Seqüência de busca do melhor par de parâmetros (C,γ) utilizando o método *grid-search*.

dimensões de caracteres. Uma vez que o número de exemplos não-textuais é muito superior ao número de exemplos textuais em um conjunto de imagens, é necessário limitar o número de regiões não-textuais para evitar o desbalanceamento[9] do conjunto de treinamento. Embora o algoritmo SVM utilize uma minimização de risco estrutural, o que indica que ele é insensível à quantidade de exemplos em cada classe [12], Chew et al. [14] demonstraram que em classes que possuem superposição, um desbalanceamento do conjunto de treinamento pode resultar em uma regra de decisão tendenciosa para a classe com o maior número de exemplos. Para se evitar o tratamento do conjunto de treinamento desbalanceado, são selecionadas aleatoriamente apenas 2608 exemplos de regiões não-textuais (30% a mais do que a classe textual)[10].

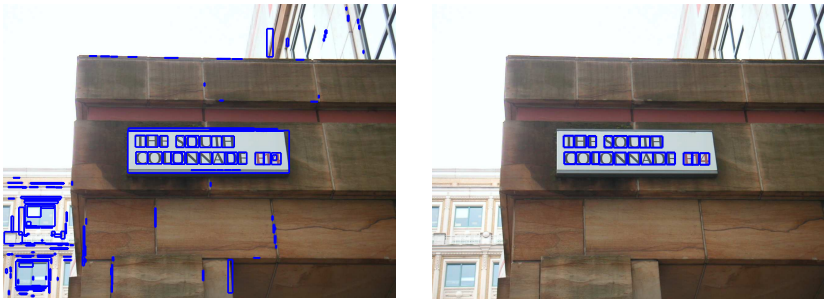
Um vetor $\boldsymbol{x}$ de atributos, obtidos na etapa de seleção de características, é extraído de cada um dos 4614 exemplos textuais e não-textuais. Tal vetor, em conjunto com o rótulo correspondente, é usado para o treinamento do algoritmo SVM, utilizando o método de busca de parâmetros *grid-search* associado à técnica *10-fold cross-validation*, como discutido anteriormente.

As Figs. 3.28 e 3.29 apresentam as áreas candidatas a texto obtidas durante a etapa de localização e as regiões consideradas textuais após a etapa de verificação. Nota-se que o método de verificação proposto é capaz de manter as regiões textuais eliminando uma considerável quantidade de falsos-positivos.

No Capítulo 5, é avaliado o desempenho do método proposto na identificação das regiões textuais após as etapas de localização-verificação, com-

[9]Um conjunto de treinamento balanceado é aquele que possui quantidades similares de exemplos em cada classe.

[10]O tratamento para o desbalanceamento do treinamento geralmente é aplicado quando a razão entre o número de exemplos entre as classes está acima de 2.



(a) (b)

Fig. 3.28: Exemplos das etapas de localização e verificação. (a) Resultado da etapa de localização (anterior à etapa de verificação). (b) Resultado após a etapa de verificação proposta.

(a) (b)

Fig. 3.29: Exemplos das etapas de localização e verificação. (a) Resultado da etapa de localização (anterior à etapa de verificação). (b) Resultado após a etapa de verificação proposta.

parando-o com o método de Jain e Yu [27].

3.4 Considerações Finais

O método de verificação proposto neste capítulo considera um algoritmo de aprendizado SVM, utilizando diversos atributos estruturais e texturais, que aumentam a robustez na determinação das áreas textuais. Tal verificação proporciona uma filtragem refinada sobre as imagens obtidas durante a etapa de localização, eliminando o grande número de falsos-positivos sem comprometer a identificação das regiões textuais. No entanto, regiões textuais não identificadas na etapa de localização não são recuperadas, visto que a

verificação é realizada apenas nas regiões previamente localizadas.

Vale ressaltar que uma quantidade considerável de regiões textuais identificadas na etapa de localização são rejeitadas na etapa de verificação devido à má extração dos contornos textuais dos caracteres. A variação abrupta de iluminação, sombras e efeitos artísticos aplicados sobre os caracteres podem acarretar na extração de contornos incompletos (não fechados) durante a etapa de localização. Por conseguinte, um conjunto de atributos extraídos de tais contornos possuem características não-textuais, levando o algoritmo SVM a classificar incorretamente e eliminar (posteriormente) tais regiões.

Capítulo 4

Etapa de Extração

Após as etapas de localização e verificação das regiões textuais, o sistema TIE conhece as regiões da imagem que provavelmente contêm texto. Em imagens complexas, tais regiões possuem milhares de cores e caracteres com fontes de estilo e tamanho desconhecidos. Além disso, podem apresentar baixa resolução, artefatos (incluídos durante o processo de compressão), baixo contraste entre caracteres e plano de fundo e iluminação não-uniforme. Os sistemas de reconhecimento óptico de caracteres OCR tradicionais, apesar de apresentarem uma alta taxa de reconhecimento para imagens documento[1], são incapazes de reconhecer texto de imagens que possuem as características supracitadas. Por conseguinte, para imagens complexas, torna-se inviável alimentar diretamente os sistemas OCR com as regiões da imagem identificadas como textuais para a conversão em texto plano.

Para que a taxa de reconhecimento de caracteres dos sistemas OCR torne-se aceitável, faz-se necessário um pré-processamento das regiões textuais conhecido como *binarização* ou segmentação da imagem em duas regiões: texto e plano de fundo. A segmentação pode ser definida como a decomposição de uma imagem em regiões que são homogêneas de acordo com algum critério. O algoritmo de segmentação deve adaptar-se ao conteúdo da imagem, separando cada objeto desejado em uma região. Além disso, é interessante que o algoritmo assegure a consistência espacial ou a intra-conectividade entre as regiões.

[1]Caracteres com um alto contraste em relação a um plano de fundo uniforme, por exemplo, plano de fundo branco com caracteres em preto.

A binarização para TIE consiste na transformação da imagem contendo textos em uma imagem binária, atribuindo-se aos pixels do plano de fundo o valor binário '1' (branco) e aos pixels dos caracteres o valor binário '0' (preto). Tal transformação gera uma imagem em que caracteres e plano de fundo exibem alto contraste, tornando-a adequada para os reconhecimento dos caracteres utilizando os sistemas OCR. Assim, denomina-se *extração* a etapa de adequação das imagens complexas às necessidades dos sistemas OCR para que o reconhecimento dos caracteres seja alcançado [32].

Este capítulo explora três abordagens para a binarização da imagem: limiarização global, limiarização local e clusterização de cor. Os algoritmos de limiarização global e local segmentam a imagem (níveis de cinza) em duas regiões por meio de uma superfície de limiarização. Uma região corresponde aos pixels que possuem valores de intensidade superiores aos valores da superfície e a outra, valores inferiores. A abordagem de clusterização de cor segmenta a imagem colorida em regiões através da aglomeração dos pixels com cores similares. Tal similaridade é avaliada por meio da distância entre as cores em um espaço em que pixels que satisfazem um certo grau de homogeneidade de cor são agrupados, formando um cluster. Ao final da clusterização, a imagem está segmentada em diversas regiões, em que cada região agrega pixels com cores similares. Partindo da premissa que o corpo dos caracteres é constituído de pixels com homogeneidade de cor, após a clusterização, existe uma região entre as geradas que representa os caracteres da imagem. Produz-se assim uma imagem binária atribuindo-se aos pixels de tal região o valor binário '0' (preto) e a todas as outras, o valor binário '1' (branco).

Os algoritmos de clusterização de cor podem ser classificados em duas categorias: *feature-space* e *image-domain* [46]. Os algoritmos *feature-space* utilizam como critério de similaridade unicamente as características de cor dos pixels. Uma vez que o critério de similaridade não avalia a relação espacial entre os pixels, a segmentação pode produzir regiões fragmentadas, ou seja, cada região gerada pode ser composta por diversos componentes conectados. A sua grande vantagem é a baixa complexidade computacional. Os algoritmos *image-domain* geram regiões coesas por incluir, no critério de similaridade, a relação espacial entre os pixels; porém, a melhoria na segmentação é obtida às custas de uma alta complexidade computacional.

Este trabalho propõe um novo método para a etapa de extração, utilizando a abordagem de clusterização de cor *feature-space*. A escolha da abordagem *feature-space* visa obter um extrator de baixa complexidade computacional, porém capaz de evitar a segmentação incorreta em regiões críticas da imagem: bordas dos caracteres. Tais bordas, por possuírem variações abruptas de intensidade, são caracterizadas por componentes de alta freqüência. Durante o processo de compressão, reduz-se a quantidade de informação presente

em tais componentes, conseqüentemente os pixels próximos às bordas dos caracteres podem adquirir cores perceptualmente muito diferentes dos pixels da sua vizinhança (artefatos). Uma vez que o critério de similaridade dos algoritmos *feature-space* avalia apenas a semelhança entre as cores, segmentações incorretas tendem a ocorrer principalmente nas áreas de variações abruptas de cor na imagem (bordas dos caracteres). Dessa forma, o algoritmo proposto, apesar de utilizar a abordagem de clusterização de cor *feature-space*, propõe um processo iterativo capaz de verificar e corrigir segmentações incorretas nas bordas dos caracteres. Tal processo iterativo descaracteriza o algoritmo proposto como puramente *feature-space*, pois avalia espacialmente a segmentação da imagem nas bordas dos caracteres.

O capítulo está organizado da seguinte forma. As Seções 4.2, 4.3 e 4.4 apresentam diversos algoritmos de binarização que são implementados para comparação de desempenho. Inclui-se na Seção 4.4 a descrição do algoritmo proposto em detalhe. A Seção 4.5 exibe os resultados experimentais de cada algoritmo e a Seção 4.6 apresenta as considerações finais do capítulo.

4.1 Algoritmos de Binarização

Esta seção descreve o método de binarização proposto, como também apresenta uma breve descrição de seis tradicionais algoritmos que foram implementados visando à comparação de desempenho. Os algoritmos de binarização (quanto aos princípios de segmentação da imagem) podem ser classificados em três abordagens diferentes: clusterização de cor, limiarização global e limiarização local.

As abordagens de clusterização de cor são capazes de tratar imagens coloridas, segmentando a imagem a partir da aglomeração dos pixels com cores similares. As abordagens de limiarização global e local lidam apenas com imagens em escala de cinza, logo, para a utilização de tais abordagens, as imagens coloridas em RGB são convertidas em uma imagem de luminância $\boldsymbol{Y}$ (níveis de cinzas) obtida pela seguinte equação:

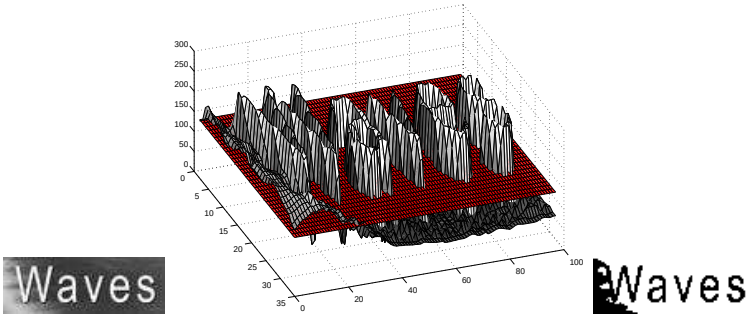
$$\boldsymbol{Y} = \boldsymbol{I}_{\mathbf{cinza}} = 0.2989\boldsymbol{R} + 0.5870\boldsymbol{G} + 0.1140\boldsymbol{B} \tag{4.1}$$

onde $\boldsymbol{R}$, $\boldsymbol{G}$ e $\boldsymbol{B}$ são os planos do espaço de cor RGB.

No restante desta seção, apresenta-se uma breve caracterização de cada abordagem, bem como uma descrição dos seus algoritmos correspondentes.

4.2 Abordagem de Limiarização Global

Os métodos de limiarização global calculam um único valor de limiar T para todos os pixels da imagem. Pixels que possuem um nível de cinza cujo valor é maior do que o limiar T são considerados pixels de impressão (preto), caso contrário são considerados como plano de fundo (branco), como apresentado na Fig. 4.1. O método proposto por Otsu [49] está entre os melhores métodos de limiarização global [62] e é utilizado para comparação com o método proposto neste trabalho.



(a) (b) (c)

Fig. 4.1: Exemplo de limiarização global. (a) Imagem original (nível de cinza). (b) Imagem original apresentada como uma superfície de valores de intensidade (cinza) e segmentada pelo plano de limiarização global (vermelho). (c) Imagem binarizada.

4.2.1 Método de Otsu

O algoritmo de Otsu [49] propõe a segmentação da imagem intensidade em duas classes, em que o limiar ótimo T é aquele que minimiza a variância interna da intensidade dos pixels de cada classe. Um outro ponto de vista seria encontrar o limiar que maximiza a variância entre tais classes. Devido à vantagem computacional, geralmente, utiliza-se a abordagem da maximização da variância entre as classes para obtenção do limiar ótimo T. Assim, busca-se o limiar T que maximiza a variância entre as classes, como apresentado na seguinte equação:

$$\sigma^2_{\mathrm{Entre-Classes}}(T) = \omega_{\mathrm{PF}}(T)(\mu_{\mathrm{PF}}(T) - \mu_{\mathrm{I}})^2 + \omega_{\mathrm{O}}(T)(\mu_{\mathrm{O}}(T) - \mu_{\mathrm{I}})^2 \quad (4.2)$$

onde $\omega_{\mathrm{PF}}(T)$ e $\omega_{\mathrm{O}}(T)$ representam a soma das probabilidades das intensidades dos pixels do plano de fundo e dos pixels do objeto, respectivamente. As

variáveis $\mu_{\mathrm{PF}}(T)$, $\mu_{\mathrm{O}}(T)$ e μ_{I} representam a média da intensidade dos pixels do plano de fundo, do objeto e da imagem, respectivamente.

Este método possui um desempenho adequado na binarização de imagens cujo histograma é bimodal com um vale bem definido entre picos. Quando a área do objeto é pequena comparada à área do plano de fundo, o histograma deixa de apresentar bimodalidade [35], prejudicando o desempenho do método de Otsu. Além disso, o método de Otsu mostra-se limitado quando a diferença entre μ_{PF} e μ_{O} é reduzida ou o objeto e o plano de fundo exibem variâncias elevadas de intensidade [37].

4.3 Abordagem de Limiarização Local

Os algoritmos de limiarização local atribuem um limiar $T(x,y)$ para cada pixel da imagem utilizando características da região onde este está alocado. Cria-se assim, uma superfície de limiarização que adapta-se às características locais da imagem, como mostrado na Fig. 4.2 (vermelho). Para coletar tais características, uma janela (ou máscara) deslizante W move-se extraindo informações dos pixels sob a janela, tais como, média, variância, faixa dinâmica, etc. Tais dados servem de suporte para a determinação de um limiar $T(x,y)$ para o pixel sob avaliação, que corresponde ao pixel sob o centro da janela W. Esse processo continua até que a janela tenha percorrido todos os pixels da imagem.

Neste trabalho, são implementados os métodos de limiarização local de Niblack [48], Savoula [57], Bernsen [6] e Chen e Yuille [13] para a avaliação de desempenho.

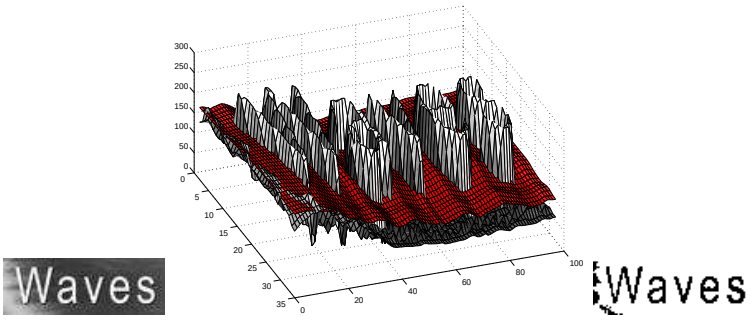
4.3.1 Método de Niblack

O método de Niblack [48] determina os valores dos limiares por meio do cálculo da média e do desvio padrão dos pixels sob uma janela deslizante W que percorre toda a imagem. Para cada pixel da imagem obtém-se um limiar $T(x,y)$ mediante a seguinte equação:

$$T(x,y) = m(x,y) + K \cdot s(x,y) \tag{4.3}$$

onde $m(x,y)$ denota a média, e $s(x,y)$, o desvio padrão da intensidade dos pixels presentes em uma região da imagem sob a janela W.

O valor K é uma constante que determina o quanto da região das bordas do objeto é considerado como parte do objeto. Valores de K próximos ao limiar inferior geram caracteres de traços espessos, enquanto valores próximos ao limiar superior produzem traços delgados, possibilitando a ruptura



(a) (b) (c)

Fig. 4.2: Exemplo de limiarização local. (a) Imagem original (em níveis de cinza); (b) Imagem original apresentada como uma superfície de valores de intensidade (cinza) e segmentada pela superfície de limiarização local (vermelho). (c) Imagem binarizada.

dos caracteres durante a binarização. O tamanho da janela W também é determinante no processo de binarização. Janelas subdimensionadas fazem com que os pixels de ruído influenciem na determinação dos valores de limiar, enquanto janelas sobre-dimensionadas resultem em um limiar inadequado por não preservar os detalhes locais.

A equação (4.3) possui valores fixos de K e W independentes da imagem de entrada. Como conseqüência, torna-se difícil encontrar valores para tais constantes que produzam resultados satisfatórios para diferentes imagens. Para a comparação desse método com outros algoritmos utilizou-se o valor $0{,}6$ para K e janela W de 25×25. Tais valores foram obtidos heuristicamente por He et al. [25].

4.3.2 Método de Chen e Yuille

O método de Chen e Yuille [13] trata-se de uma variante do algoritmo de Niblack. Uma vez que uma janela $W = r \times r$ de dimensões fixas é inadequada para diferentes imagens e até mesmo para regiões dentro da mesma imagem, Chen e Yuille propuseram uma janela cujas dimensões se adaptam à região da imagem. Assim,

$$W(x,y) = r(x,y) \times r(x,y) \quad \text{onde} \quad r(x,y) = \min_{r}(s_r(x,y) > T_\sigma). \quad (4.4)$$

Para cada pixel da imagem as dimensões da janela são incrementadas gradativamente, até que o desvio padrão da intensidade dos pixels $s_r(x,y)$

sob a janela seja maior do que uma constante T_σ previamente determinada. Tal constante indica quando a janela atingiu a dimensão apropriada para a geração do limiar do pixel sob avaliação. O valor de T_σ deve possuir uma magnitude superior ao desvio padrão $s_r(x,y)$ em áreas (sob a janela W) com variações suaves de intensidade. Uma vez determinadas as dimensões das janelas para cada pixel, aplica-se o método de Niblack (4.3) à imagem com as janelas correspondentes a cada pixel.

Para propósito de comparação de desempenho, os testes experimentais são realizados utilizando os valores $0{,}4$ e 40 para as constantes K e T_σ, respectivamente.

4.3.3 Método de Sauvola

Sauvola [57] propõe uma versão modificada do método de limiarização local de Niblack, incluindo para o cálculo do limiar uma faixa dinâmica para o desvio padrão. Além disso, a média local $m(x,y)$, proporciona uma amplificação da contribuição do desvio padrão $s(x,y)$ de forma adaptativa, como apresentado na equação seguinte:

$$T(x,y) = m(x,y) \cdot \left[1 + K \cdot \left(\frac{s(x,y) - R}{R}\right)\right]. \tag{4.5}$$

Segundo estudo realizado por He et al. [25], o melhor desempenho do método de Sauvola foi alcançado para uma janela W de dimensões 9×9 e constantes K e R com os valores $0{,}09$ e 128 respectivamente. Tais valores são utilizados neste trabalho para propósito de comparação com os demais algoritmos.

4.3.4 Método de Bernsen

Bernsen [6] desenvolveu um método de limiarização local baseado na máxima diferença entre pixels englobados por uma janela deslizante $W = r \times r$, onde r é um número ímpar. À medida que a janela desliza sobre a imagem, um limiar é atribuído ao pixel central da janela baseado nas características dos pixels da vizinhança. Bernsen propôs a seleção dos pixels de maior e menor intensidades, $Z_{\max}$ e $Z_{\min}$, respectivamente, sob a janela W. Caso a diferença entre tais pixels seja maior ou igual a uma constante L, atribui-se um limiar dado pela média aritmética de tais pixels, caso contrário o pixel central da

janela é considerado como plano de fundo.

$$T(x,y) = \begin{cases} \dfrac{Z_{\max} + Z_{\min}}{2}, & (Z_{\max} - Z_{\min}) \geq L \\ \text{Plano de fundo}, & \text{caso contrário.} \end{cases} \tag{4.6}$$

Visando comparar o desempenho dos algoritmos de binarização, adotou-se para o método Bernsen uma janela W de dimensões 15×15 e valor 15 para a constante L, valores esses também propostos por Trier e Taxt [63].

4.4 Abordagem de Clusterização de Cor

Os algoritmos de binarização baseados na clusterização de cor segmentam a imagem em regiões por meio da aglomeração dos pixels com cores similares. Tal similaridade é avaliada utilizando-se uma medida de distância entre as cores em um espaço de cor. As métricas mais comuns para a avaliação de similaridade entre as cores são as distâncias de Minkowski [23], tais como euclidiana, *city-block (ou Manhattan)* e Chebyshev. Após a segmentação da imagem, cada região é transformada em uma imagem binária. Tais imagens binárias são comumente identificadas como planos e estes permitem tratar cada objeto segmentado individualmente, separando do objeto de interesse (caracteres) toda informação adicional contida na imagem.

4.4.1 Método de Jain e Yu

O algoritmo de extração de texto de Jain e Yu [27] considera que os caracteres possuem uma satisfatória uniformidade de cor. Contudo, uma imagem colorida em RGB possui 24-bits (8-bits para cada canal), permitindo que uma imagem apresente até 2^{24} cores. Visando reduzir a complexidade computacional de manipular um grande número de cores, Jain e Yu aplicam a técnica *bit dropping*. Tal técnica utiliza apenas alguns bits mais significativos de cada canal da imagem (RGB), reduzindo assim, o número de cores possíveis. Jain e Yu utilizam os dois bits mais significativos, transformando a imagem para o espaço de cor de 6-bits e restringindo para 64 o número de cores possíveis. O *bit dropping* reduz consideravelmente o custo computacional para a clusterização das cores; porém, o efeito da quantização das cores é a transformação de cores perceptualmente diferentes em uma única cor.

O algoritmo de Jain e Yu emprega o método de clusterização hierárquica *single-link* sobre as cores da imagem de 6-bits. A clusterização hierárquica exige uma medida de dissimilaridade entre duas cores

$$\boldsymbol{x} = (x_{\mathrm{R}'}, x_{\mathrm{G}'}, x_{\mathrm{B}'}) \text{ e } \boldsymbol{y} = (y_{\mathrm{R}'}, y_{\mathrm{G}'}, y_{\mathrm{B}'})$$

que é definida como

$$d_{\mathrm{JY}}(\boldsymbol{x},\boldsymbol{y}) = (x_{\mathrm{R}'} - y_{\mathrm{R}'})^2 + (x_{\mathrm{G}'} - y_{\mathrm{G}'})^2 + (x_{\mathrm{B}'} - y_{\mathrm{B}'})^2 \qquad (4.7)$$

onde R', G' e B' representam os valores de cada canal da imagem de 6-bits.

A partir de (4.7) torna-se possível construir uma matriz de proximidade para cada estágio do processo de clusterização, em que as duas cores com o mínimo valor na matriz de proximidade são fundidas. Quando duas cores são fundidas, a cor assumida pelos pixels que possuem as cores da fusão é dada pela cor de maior ocorrência no histograma. Após a fusão, uma nova matriz de proximidade é calculada com as cores restantes. Esse processo de quantização de cor continua até que o número de cores atinja o valor predeterminado de 2 ou o mínimo valor na matriz de proximidade seja maior do que 1.

4.4.2 Algoritmo Proposto de Extração de Texto

O método aqui proposto para a etapa de extração de texto em imagens visa alcançar três objetivos:

- baixa complexidade computacional;
- geração do menor número possível de regiões durante a segmentação[2];
- geração de regiões com coesão espacial em imagens contendo artefatos e objetos de baixa densidade, evitando que caracteres sejam fragmentados.

4.4.2.1 Espaço de Cor e Métrica de Similaridade

Para reduzir a complexidade computacional, optou-se por trabalhar com o espaço de cor RGB. Embora ele não seja adequado à percepção visual humana [33], o conjunto de dados nesse espaço não exige qualquer transformação para inicializar o processamento.

Para reduzir a influência da não conformidade do espaço de cor utilizado com respeito à percepção visual humana, buscou-se uma métrica de similaridade em que cores distantes no espaço RGB sejam significativamente diferentes para o observador humano. Fundamentado em um experimento de percepção visual realizado por Loo e Tan [44], escolheu-se a métrica *city-block*. A distância entre cores em um espaço é utilizada como uma medida de similaridade de cor. Cada cor no espaço RGB é representada por um

[2]Considera-se que duas regiões seja o número ótimo, na qual uma região corresponde ao texto e a outra, ao plano de fundo.

vetor de 3 componentes (R, G, B) e a distância *city-block* entre duas cores $\boldsymbol{x} = (x_{\mathrm{R}}, x_{\mathrm{G}}, x_{\mathrm{B}})$ e $\boldsymbol{y} = (y_{\mathrm{R}}, y_{\mathrm{G}}, y_{\mathrm{B}})$ é dada por:

$$d_{\text{city-block}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{i=1}^{3} |x_i - y_i| = |x_{\mathrm{R}} - y_{\mathrm{R}}| + |x_{\mathrm{G}} - y_{\mathrm{G}}| + |x_{\mathrm{B}} - y_{\mathrm{B}}|. \quad (4.8)$$

Para definir qual das métricas, euclidiana ou *city-block*, detém maior conformidade com o sistema visual humano, Loo e Tan [44] realizaram o seguinte experimento:

1. uma cor pivô é escolhida aleatoriamente;
2. escolhe-se uma distância $\mathcal{D}$ para inicializar o experimento;
3. utilizando a métrica *city-block*, coletam-se aleatoriamente 620 cores distantes de $\mathcal{D}$ unidades da cor pivô;
4. tais cores são visualmente inspecionadas por 10 pessoas para determinar a similaridade entre tais cores e a cor pivô;
5. repetem-se os passos 3 e 4 utilizando a métrica Euclidiana;
6. incrementa-se o valor da distância $\mathcal{D}$ e repete-se o experimento do passo 3 até que as cores sejam perceptualmente distintas da cor pivô;

Uma reprodução do experimento pode ser observado na Fig. 4.3, demonstrando que, na métrica *city-block*, cores a uma distância fixa da cor pivô são perceptualmente mais estáveis. Em [44], pode-se obter uma tabela de correspondência entre a percepção visual humana e a distância entre cores (*city-block*) no espaço RGB. Percebe-se de tal tabela que cores distantes de 120 unidades são perceptualmente distintas quando utilizada a métrica *city-block*. Além disso, tal métrica possui complexidade computacional inferior a métrica euclidiana, visto que a distância *city-block* entre dois vetores é apenas a soma do módulo das diferenças dos componentes correspondentes (4.8).

4.4.2.2 Região de Borda

Aqui é discutido o conceito de região de borda como também é fundamentada a necessidade da eliminação desse tipo de região.

Define-se região de borda como toda região gerada durante a clusterização de cor que possui mais de 25% dos seus pixels sobre as bordas da imagem. Tais bordas são representadas por qualquer imagem binária obtida a partir de alguma técnica de detecção de borda, por exemplo, *Sobel* ou *Canny* [23].

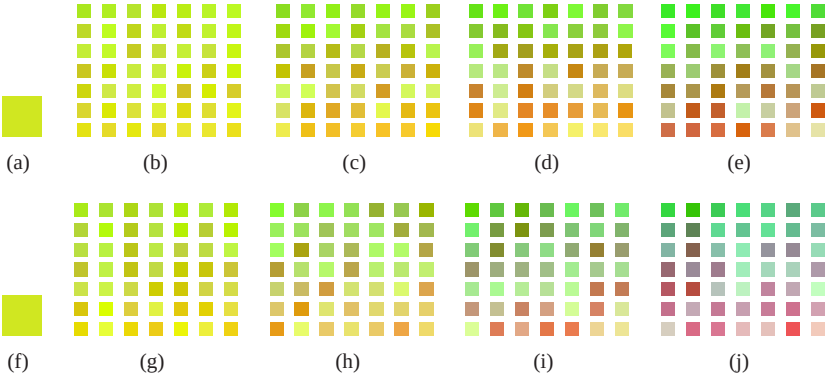


Fig. 4.3: Experimento de percepção visual realizado por Loo e Tan. (a) Cor pivô. Cores a uma distância *city-block* de: (b) 40; (c) 80; (d) 120; (e) 160; (f) Cor pivô. Cores a uma distância euclidiana de: (g) 40; (h) 80; (i) 120; (j) 160.

Regiões de borda, como mostradas na Fig. 4.7(b) (região em azul), são *indesejadas* por três razões principais:

1. geralmente são regiões com muitas fragmentações;
2. são constituídas de pixels que sofreram variação de cor durante o processo de compressão, adquirindo cores muito distintas dos pixels de sua vizinhança, não representando qualquer objeto na imagem;
3. possuem pixels que deveriam estar agregados à região do corpo do caractere, porém foram alocados em uma região diferente;

Devido às razões supracitadas, o método de extração proposto inclui uma malha de realimentação capaz de evitar a geração de regiões de borda durante o processo de clusterização das cores, melhorando a qualidade da segmentação da imagem.

4.4.2.3 Método K-means Recorrente

As técnicas de clusterização de cor baseadas na abordagem *feature-space* possuem baixa complexidade computacional, porém geralmente fragmentam objetos de baixa densidade (característica comum em caracteres). O método proposto utiliza a abordagem *feature-space*, produzindo regiões compactas por associar uma malha de realimentação capaz de identificar e eliminar regiões segmentadas incorretamente nas bordas dos caracteres, como ilustrado no diagrama de blocos da Fig. 4.4. Tal técnica é denominada *K-means*

recorrente [60], devido à recorrência feita ao algoritmo de clusterização *K-means* [31] sempre que ocorre a identificação de regiões de borda. No presente trabalho, extraiu-se o texto da *imagem-exemplo* (Fig. 4.5) para descrever detalhadamente o algoritmo proposto. O método *K-means* tradicional não

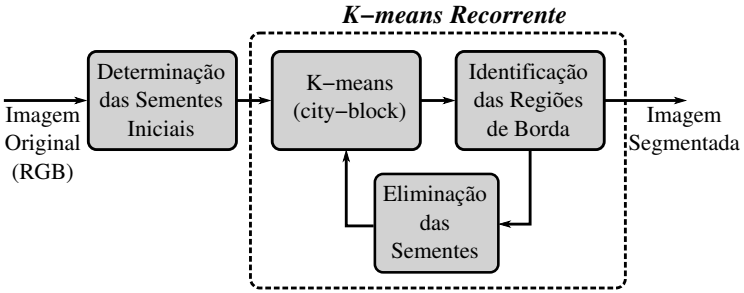


Fig. 4.4: Diagrama de blocos do método de extração de caracteres *K-means recorrente*.

Fig. 4.5: Imagem exemplo contendo artefatos.

determina automaticamente nem o número nem quais as sementes iniciais que devem ser usadas para o processo de clusterização de cor. Para automatizar tal processo, o método proposto determina quais são as sementes iniciais através da avaliação do histograma de cores da imagem a partir dos seguintes passos (1º bloco, Fig. 4.4):

1. Cálculo do histograma de cores da imagem.

2. Seleção das 5 cores de maior ocorrência no histograma como candidatas a sementes. A escolha de apenas 5 cores produz resultados similares à avaliação de todas as cores como candidatas.

3. Dentre as cores candidatas, escolhe-se como semente aquela que possui o maior número de pixels com cores a uma distância *city-block* menor do que um limiar $\mathcal{T}$, em que $0 \leq \mathcal{T} \leq 765$ (ou seja, 3×255). Utilizou-se neste trabalho $\mathcal{T} = 120$, por tal valor estar em conformidade com o experimento de percepção realizado por Loo e Tan [44].

4. Atribui-se à semente todas as cores que estão a uma distância (*city-block*) menor do que o limiar $\mathcal{T}$.

5. O processo é reinicializado a partir do passo 2 com todas as cores não atribuídas a uma semente. A determinação do conjunto de sementes iniciais é concluída quando todas as cores da imagem estão atribuídas a uma semente.

Após a determinação automática das sementes, o método *K-means* é inicializado usando a métrica *city-block* para medir as distâncias entre as amostras e determinar o centróide de cada *cluster* (bloco *K-means*, Fig. 4.4). Ao final da clusterização, as cores da imagem são segmentadas em um número de *clusters* igual ao número de sementes iniciais. Para a *imagem-exemplo* (Fig. 4.5), são geradas 3 sementes, segmentando as cores da imagem em 3 *clusters*, como ilustrado na Fig. 4.6. Cada *cluster* corresponde a uma região da imagem original. Tais regiões estão representadas na Fig. 4.7(b) com cores equivalentes aos *clusters* correspondentes. É comum no processo de extração de caracteres trabalhar com cada região separadamente, em que cada região torna-se um *plano* (imagem binária), como mostrado nas Figs. 4.7(c), (d) e (e). Tal separação da imagem original em planos faz-se necessária para a seleção do plano em que os únicos objetos da imagem são os caracteres [Fig. 4.7(e)], tornando possível a transformação de tais caracteres em texto plano através do sistema de OCR.

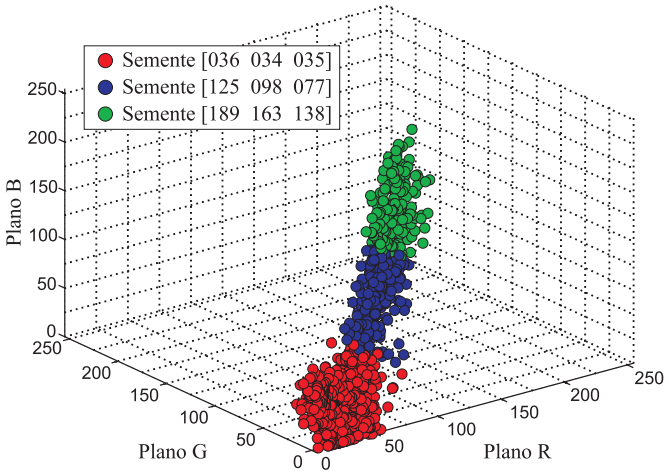


Fig. 4.6: Clusterização de cores da *imagem-exemplo* através das 3 sementes iniciais.

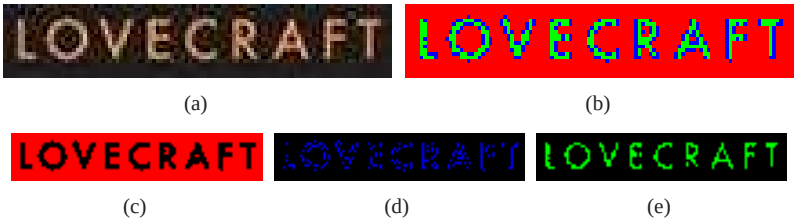


Fig. 4.7: Regiões clusterizadas antes da recorrência ao método *K-means*. (a) Imagem original. (b) Imagem segmentada em três regiões definidas pela clusterização apresentada na Fig. 4.6. Plano gerado pelo *cluster* referente à: (c) semente [036 034 035]; (d) semente [125 098 077]; (e) semente [189 163 138].

Observando as regiões geradas pelo método *K-means*, percebe-se que o *cluster* referente à semente [125 098 077] (representando os valores dos planos [R G B]), como ilustrado na Fig. 4.7(d), é indesejável por originar uma imagem binária bastante fragmentada, não representando qualquer objeto da imagem e constituída preponderantemente das bordas dos caracteres. Através das Figs. 4.7(e) e (d) observa-se que os pixels referentes a esses dois planos deveriam ter sido agrupados em um único cluster, ou seja, os clusters verde e azul da Fig. 4.7(b) deveriam ser fundidos em um único cluster. Isto evitaria que os caracteres apresentassem deformações e rupturas [Fig. 4.7(e)] devido a pixels que são clusterizados em uma região de borda [Fig. 4.7(d)], quando deveriam estar agregados ao corpo dos caracteres.

Devido ao processo de compressão, os pixels das bordas da imagem possuem cores perceptualmente muito diferentes dos pixels da sua vizinhança, tornando comum agregar tais pixels, durante a clusterização de cor, em uma região composta predominantemente pelas bordas da imagem [Fig. 4.7(d)].

A geração automática de sementes pode eleger, erroneamente, uma semente indesejada, resultando em uma região de borda. Para corrigir esse tipo de erro na determinação do número de sementes, usa-se então o método *K-means recorrente* proposto. Tal método verifica, ao final de cada clusterização, a existência de regiões de borda entre as regiões geradas (bloco de identificação, Fig. 4.4). Caso alguma região seja caracterizada como região de borda, a semente geradora dessa região é descartada (bloco de eliminação, Fig. 4.4) e o método *K-means* é reinicializado com as sementes sobreviventes até que não existam mais regiões de borda.

A etapa de identificação das regiões de borda (bloco de identificação, Fig. 4.4) é feita por meio da operação binária *AND* entre cada plano obtido após a clusterização [Figs. 4.8(c), 4.8(f) e 4.8(i)] e uma *imagem-borda*. Esta

última é obtida por qualquer método de detecção de bordas sobre a imagem original [Fig. 4.8(b)]. No presente trabalho, utilizou-se o método *Sobel* [23], como mostrado na Fig. 4.8(b). Como resultado, obtém-se imagens de intersecção entre as regiões clusterizadas e a *imagem-borda*, como ilustrado nas Figs. 4.8(e), 4.8(h) e 4.8(k). Efetua-se então a contagem dos pixels de cada imagem de intersecção. Caso o número ultrapasse em 25% o número de pixels da respectiva região clusterizada, a região é considerada de borda.

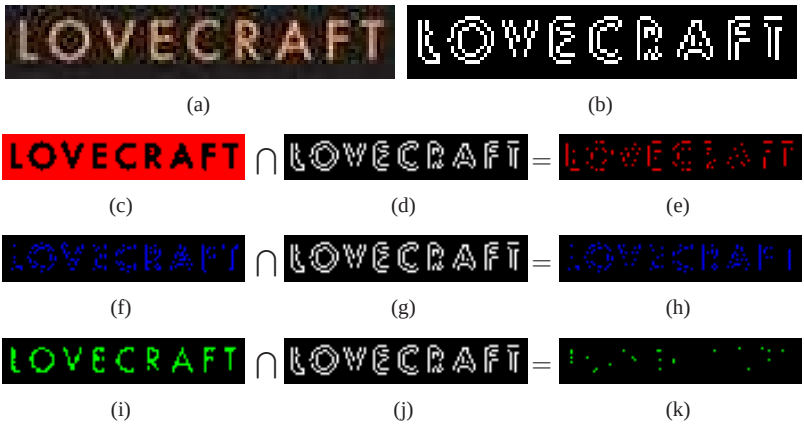


Fig. 4.8: Processo de identificação das regiões de borda. (a) Imagem original. (b) *Imagem-borda*. Imagens binárias representando o plano referente à: (c) semente [036 034 035]; (f) semente [125 098 077]; (i) semente [189 163 138]. (d), (g), (j) *Imagens-borda*. Resultado da intersecção da *imagem-borda* e os planos referentes à: (e) semente [036 034 035] (7,91% dos pixels possuem intersecção); (h) semente [125 098 077] (76,56% dos pixels possuem intersecção); (k) semente [189 163 138] (10,48% dos pixels possuem intersecção).

No exemplo em questão, pode-se observar que apenas a região representada pela Fig. 4.8(f) resulta em uma imagem de intersecção [Fig. 4.8(h)] que possui mais do que 25% dos pixels da região que a originou [Fig. 4.8(f)], caracterizando-a como região de borda. O bloco de eliminação da Fig. 4.4 é responsável por descartar a semente geradora da região de borda (semente [125 098 077] para a *imagem-exemplo*) e reinicializar o método *K-means* com as sementes sobreviventes ([189 163 138] e [036 034 035]).

Após cada iteração do método *K-means recorrente* usando as sementes sobreviventes, uma nova avaliação é feita sobre as imagens binárias que representam cada região (*planos*) até que não existam mais regiões de borda ou o número de *clusters* (planos) seja igual a dois. É apresentado na Fig. 4.9 o resultado da clusterização das cores referente à primeira iteração do método

K-means recorrente utilizando as duas sementes sobreviventes. As regiões que representam cada cluster na imagem são mostradas na Fig. 4.10(b). Observa-se, ao final da clusterização utilizando o método *K-means recorrente* [Figs. 4.10(c) e 4.10(d)], que o número de regiões clusterizadas (planos) é reduzido a 2 e os pixels que antes representavam uma região de borda agora estão agregados ao corpo dos caracteres, melhorando significativamente a qualidade da segmentação.

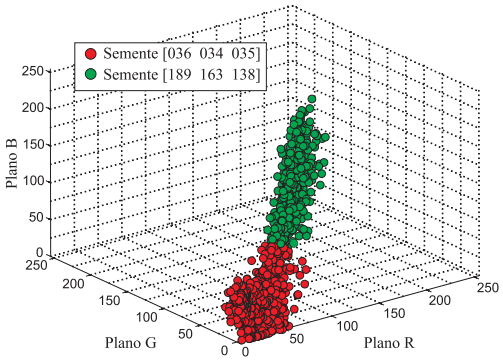


Fig. 4.9: Clusterização das cores da *imagem-exemplo* após a primeira iteração do método *K-means recorrente* com as 2 sementes sobreviventes.

(a) (b)

(c) (d)

Fig. 4.10: *Imagem-exemplo* segmentada pelo método *K-means recorrente*. (a) Imagem original. (b) Imagem segmentada em duas regiões definidas pela clusterização da Fig. 4.9. Plano gerado pelo cluster referente à: (c) semente [036 034 035]; (d) semente [189 163 138].

4.5 Resultados Experimentais

Para avaliar o desempenho do algoritmo proposto, 25 imagens contendo apenas regiões textuais foram coletadas aleatoriamente na Web (Fi-

gura C.1 - Apêndice C). Tais imagens possuem diferentes tipos de fonte, dimensões, planos de fundo e contrastes. A faixa de resolução das imagens varia entre 50×10 e 600×200 e o tamanho dos caracteres, entre 8 e 530 pixels de altura.

Muitos autores propõem, como método de avaliação de desempenho, a contagem da quantidade de caracteres reconhecidos corretamente do conjunto total de caracteres após a passagem por um sistema OCR. Tal técnica leva em consideração a qualidade do extrator em conjunto com o OCR, mascarando dessa forma o real desempenho do processo de extração. Por esse motivo, no presente trabalho, utilizou-se uma medida de discrepância baseada no número de pixels segmentados incorretamente [69]. A binarização é um processo de classificação dos pixels como texto ou plano de fundo; assim, pode-se avaliar a qualidade da segmentação de um algoritmo de binarização a partir do número de pixels que são classificados como texto e pertenciam incorretamente ao plano de fundo e vice-versa. Para que tal avaliação seja viável, faz-se necessária a geração de uma imagem referência de segmentação *ground-truth* (GT), possibilitando julgar o número de pixels classificados incorretamente por qualquer algoritmo de binarização.

Para a comparação de desempenho dos algoritmos de binarização utilizou-se a medida de discrepância denominada *probabilidade de erro* ($\mathrm{P_E}$) [42]. Para um problema de duas classes (texto e plano de fundo), a probabilidade de erro pode ser obtida da seguinte forma:

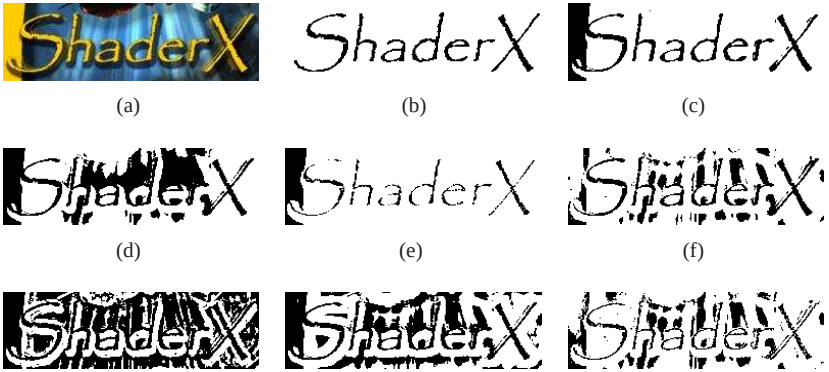
$$P_{\mathrm{E}} = P(T) \cdot P(P_{\mathrm{F}}|T) + P(P_{\mathrm{F}}) \cdot P(T|P_{\mathrm{F}}) \tag{4.9}$$

onde $P(P_{\mathrm{F}}|T)$ é a probabilidade em classificar o pixel como plano de fundo tal que ele seja texto, $P(T|P_{\mathrm{F}})$ é a probabilidade em classificar o pixel como texto tal que ele seja plano de fundo, $P(T)$ e $P(P_{\mathrm{F}})$ são as probabilidades *a priori* do texto e plano de fundo, respectivamente. As probabilidades *a priori* são obtidas da imagem referência (*ground-truth*).

Para a geração de imagens referência criou-se uma interface que permite ao usuário, a partir de uma imagem escolhida, marcar os pixels que pertencem ao texto, segmentando a imagem em duas classes: texto e plano de fundo (Fig. C.4 - Apêndice C) . Tal procedimento gera a imagem de referência *ground-truth* que é utilizada pela interface para comparar o desempenho dos algoritmos de binarização nele implementados [Fig. C.5(a) - Apêndice C]. A interface possui 7 algoritmos de binarização: dois algoritmos baseado na clusterização de cor (método proposto e o método de Jain e Yu), um de limiarização global (método de Otsu) e quatro de limiarização local (métodos de Niblack, Sauvola, Bernsen, Chen e Yuille).

As Figs. 4.11 a 4.14 representam uma amostra dos resultados para di-

ferentes tipos de fonte, planos de fundo e iluminação.

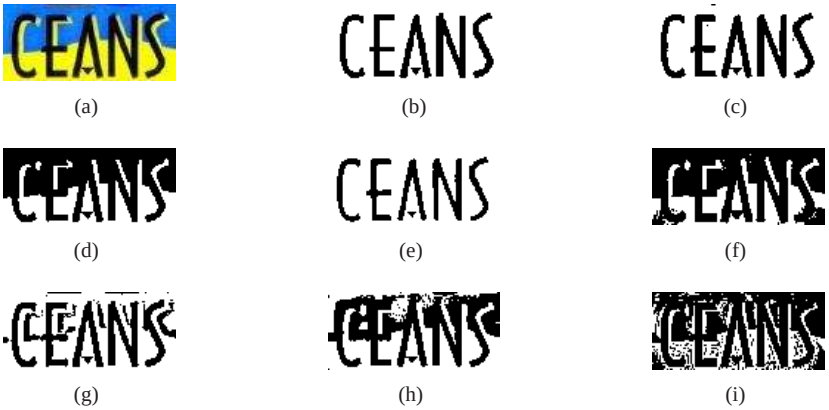


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4.11: Resultado da extração de uma imagem com plano de fundo complexo. (a) Imagem original. (b) Imagem *ground-truth*. (c) Método proposto (5 planos). (d) Método de Otsu. (e) Método de Jain e Yu (11 planos). (f) Método de Niblack. (g) Método de Sauvola. (h) Método de Bernsen. (i) Método de Chen e Yuille.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 4.12: Resultado da extração de uma imagem com 3 regiões bem definidas. (a) Imagem original. (b) Imagem *ground-truth*. (c) Método proposto (3 planos). (d) Método de Otsu. (e) Método de Jain e Yu (8 planos). (f) Método de Niblack. (g) Método de Sauvola. (h) Método de Bernsen. (i) Método de Chen e Yuille.

É importante ressaltar que apenas são ilustrados os planos referentes à região textual. Contudo, o número de planos gerados é uma informação im-

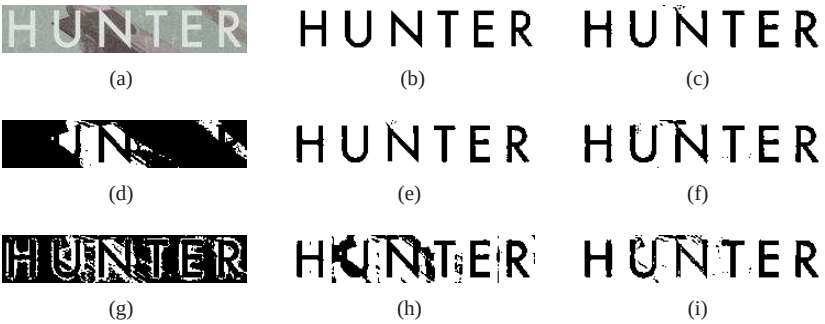


Fig. 4.13: Resultado da extração de uma imagem com iluminação não-uniforme. (a) Imagem original. (b) Imagem *ground-truth*. (c) Método proposto (4 planos). (d) Método de Otsu. (e) Método de Jain e Yu (5 planos). (f) Método de Niblack. (g) Método de Sauvola. (h) Método de Bernsen. (i) Método de Chen e Yuille.

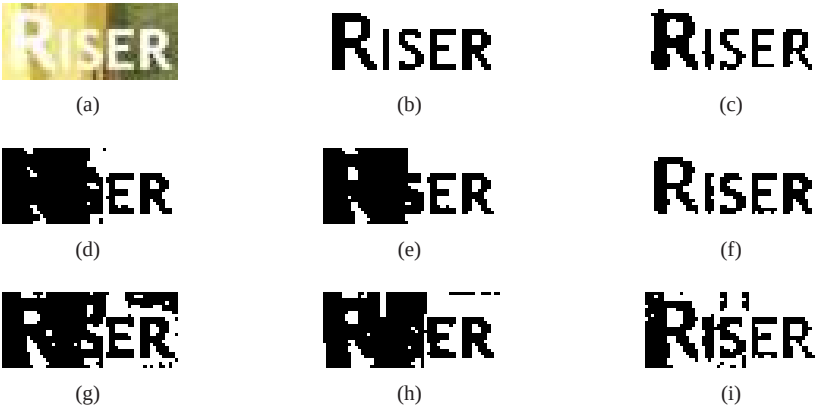


Fig. 4.14: Resultado da extração de uma imagem com baixo contraste entre caracteres e o plano de fundo. (a) Imagem original. (b) Imagem *ground-truth*. (c) Método proposto (5 planos). (d) Método de Otsu. (e) Método de Jain e Yu (8 planos). (f) Método de Niblack. (g) Método de Sauvola. (h) Método de Bernsen. (i) Método de Chen e Yuille.

portante, visto que, após qualquer método de extração, deve-se utilizar uma ferramenta para identificar qual imagem binária representa a região textual (por exemplo, projeções de perfil). Um menor número de planos indica menor complexidade computacional na identificação da imagem binária que representa os caracteres.

Tabela 4.1: Probabilidade de erro dos algoritmos para as 25 *imagens-teste*. Os valores destacados em negrito representam o algoritmo de extração que alcançou a menor P_{E} para a imagem correspondente

	Probabilidade de erro							**Nº de planos**	
	Otsu	**Proposto**	**Jain e Yu**	**Niblack**	**Sauvola**	**Bernsen**	**Chen**	**Proposto**	**Jain e Yu**
Imagem 1	**0,0327**	0,0359	0,0588	0,3137	0,0556	0,0359	0,2222	2	3
Imagem 2	**0,0250**	**0,0250**	0,0361	0,0361	0,1722	0,0361	0,0444	2	5
Imagem 3	0,3564	0,0536	0,3192	**0,0327**	0,4100	0,3586	0,1518	5	8
Imagem 4	0,0329	**0,0267**	0,0416	0,0366	0,4835	0,0801	0,0416	2	4
Imagem 5	0,2762	**0,0809**	0,1023	0,1483	0,4983	0,3034	0,1980	5	11
Imagem 6	0,0595	**0,0565**	0,0774	0,1429	0,0417	0,0714	0,1429	2	5
Imagem 7	0,1548	0,1572	0,1483	0,7166	**0,0741**	0,0974	0,2346	2	6
Imagem 8	0,0357	**0,0329**	0,1071	0,0414	0,1457	0,0400	0,1143	2	8
Imagem 9	0,0699	0,0521	**0,0335**	0,0451	0,4105	0,1108	0,1050	3	6
Imagem 10	**0,0163**	0,0327	0,0378	0,0364	0,4854	0,0255	0,1071	4	5
Imagem 11	0,0530	0,0499	0,0894	0,0520	0,1559	**0,0218**	0,0322	2	5
Imagem 12	0,3752	**0,0279**	0,0437	0,4295	0,1054	0,2554	0,4297	3	8
Imagem 13	0,0642	0,0392	**0,0187**	0,5431	0,1433	0,1140	0,4345	4	5
Imagem 14	**0,0574**	0,0676	0,2825	0,4978	0,0936	0,1320	0,4578	4	3
Imagem 15	0,5899	0,0172	**0,0142**	0,0289	0,5517	0,1337	0,0526	3	5
Imagem 16	**0,0000**	**0,0000**	**0,0000**	0,0235	0,0588	**0,0000**	0,0118	2	4
Imagem 17	**0,0108**	0,0115	0,0776	0,0201	0,2356	0,0244	0,0467	2	7
Imagem 18	0,0379	**0,0307**	0,1340	0,0327	0,1862	0,0317	0,0834	2	6
Imagem 19	**0,0148**	0,0289	0,0352	0,0748	0,3020	0,1273	0,1282	5	6
Imagem 20	0,1049	**0,0383**	0,0444	0,0644	0,3357	0,1903	0,0744	4	5
Imagem 21	0,0855	0,0684	**0,0235**	0,2308	0,0748	0,0833	0,3056	2	7
Imagem 22	0,2094	0,0863	**0,0854**	0,4978	0,2021	0,2916	0,4705	3	8
Imagem 23	0,0189	**0,0156**	0,0769	0,2879	0,0452	0,0653	0,2306	3	5
Imagem 24	0,0084	**0,0075**	0,0139	0, 0153	0,3318	0,0310	0,0443	2	5
Imagem 25	0,1777	0,1761	0,2542	**0,0616**	0,4165	0,1462	0,1755	3	7
Média	**0,1147**	**0,0487**	**0,0862**	**0,1764**	**0,2406**	**0,1123**	**0,1736**	**2,92**	**5,88**
Variância	**0,0211**	**0,0017**	**0,0071**	**0,0423**	**0,0278**	**0,0096**	**0,0205**	**1,16**	**3,28**

Os resultados da probabilidade de erro de todos os algoritmos implementados para cada imagem teste podem ser observados na Tabela 4.1. As médias das probabilidades de erro aparentemente indicam que o método proposto possui o melhor desempenho. Para uma avaliação criteriosa, faz-se necessário a verificação da significância estatística de tais resultados.

Para a avaliação dos resultados (Tabela 4.1) entre os métodos de binarização apresentados e o método proposto, é utilizado o teste de hipóteses da diferença entre duas médias populacionais. Considerando que a probabilidade de erro seja uma variável aleatória obtida de 7 populações diferentes (os 7 algoritmos de binarização implementados), em que a variância da probabilidade de erro em cada população é desconhecida, são definidas a hipótese nula H_0 e a hipótese alternativa H_1. Assim,

$$H_0 \Rightarrow \mu_{\text{algoritmo}} = \mu_{\text{proposto}} \tag{4.10}$$

$$H_1 \Rightarrow \mu_{\text{algoritmo}} > \mu_{\text{proposto}} \tag{4.11}$$

onde $\mu_{\text{algoritmo}}$ representa a média da P_{E} de um dos algoritmos implementados e μ_{proposto} a média da P_{E} do algoritmo proposto. Caso a hipótese nula H_0 seja verdadeira, indica que o algoritmo proposto provavelmente não possui um desempenho superior aos outros algoritmos, enquanto a rejeição da hipótese nula H_0 indica uma provável superioridade do algoritmo proposto e aceita-se a hipótese alternativa H_1.

Para a avaliação da significância estatística utilizou-se o método *t-student*, visto que dispomos apenas de 25 amostras (imagens) de P_{E} para cada algoritmo com variâncias desconhecidas[3]. O nível de significância estatística α escolhido é de 5% e representa a probabilidade de rejeitar *incorretamente* a hipótese nula H_0 (4.11). Assim, caso os valores obtidos t_{stat} sejam maiores do que os valores tabelados para o nível de significância de 5%, a hipótese H_1 é aceita, corroborando com a idéia de que o algoritmo proposto possui melhor desempenho.

Os resultados para o teste de hipóteses para a diferença entre duas médias utilizando *t-student* para a probabilidade de erro e para o número de planos gerados são apresentados na Tabela 4.2. No caso do número de planos gerados, consideram-se as mesmas hipóteses, porém as médias estão relacionadas com o número de planos gerados pelo algoritmo. Além disso, em relação ao número de planos, realizou-se apenas a comparação do método proposto com o método de clusterização de cor de Jain e Yu, uma vez que todos os outros algoritmos geram apenas 2 planos como saída.

Observa-se por meio da Tabela 4.2 que todos os algoritmos apresen-

[3] As variâncias são consideradas diferentes para o cálculo de significância estatística.

Tabela 4.2: Nível de significância estatística dos algoritmos

Algoritmos	Probabilidade de erro		Número de planos	
	t_{stat}	Nível de significância (%)	t_{stat}	Nível de significância (%)
Otsu	2,1817	1,89	-	-
Jain e Yu	1,9893	2,73	7,0260	$9{,}723 \cdot 10^{-7}$
Niblack	3,0405	0,27	-	-
Sauvola	5,5837	$3{,}18 \cdot 10^{-4}$	-	-
Bernsen	2,9844	0,27	-	-
Chen	4,1860	0,01	-	-

taram um nível de significância abaixo de 5% para a P_{E}, tal resultado indica que pode-se rejeitar a hipótese nula H_0 com 95% de chance de ter decidido corretamente pela hipótese alternativa H_1. Baseado na significância estatística dos resultados para as 25 *imagens-teste*, a média da P_{E} dos algoritmos implementados provavelmente é superior ao método proposto.

A mesma avaliação pode ser aplicada para a comparação do número de planos gerados pelo método de Jain e Yu e o método proposto, em que o nível de significância está abaixo de 1% (Tabela 4.2). Isso indica que, muito provavelmente, a média de planos gerados pelo método de Jain e Yu é superior ao método proposto.

Pode-se inferir que a complexidade computacional do método proposto é inferior aos algoritmos *image domain*, visto que a verificação espacial associada ocorre apenas nas regiões de borda por meio da operação lógica *AND*. No entanto, a complexidade computacional do método proposto não pode ser predita, uma vez que o número de recorrências à clusterização *K-means* varia para cada imagem. Para a avaliação do número de recorrências, 130 imagens contendo apenas texto são submetidas ao método proposto resultando em uma média de recorrências de $1{,}15$, significando que, em média, as imagens necessitam de apenas uma recorrência para correção da segmentação na região de borda da imagem.

4.6 Considerações Finais

Neste trabalho, desenvolveu-se uma nova técnica de extração de texto que associa o bom desempenho das técnicas de clusterização de cor *feature-space* (utilizando todas as cores da imagem) com uma avaliação da segmentação em regiões críticas (bordas da imagem). O algoritmo proposto é capaz de identificar, iterativamente, regiões segmentadas desnecessariamente, corrigindo-as, de forma recorrente, pelo método *K-means* após a eliminação

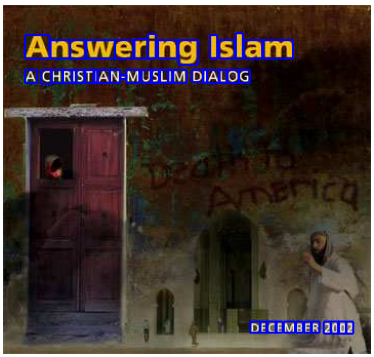
das sementes correspondentes às regiões de borda.

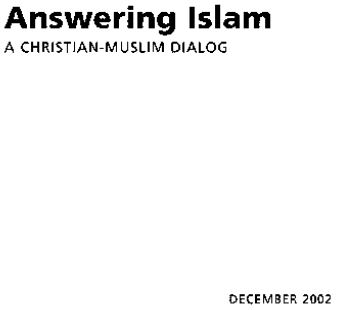
De acordo com o grupo de imagens testadas, percebe-se que o método de Otsu (limiarização global) é incapaz de segmentar imagens com presença de iluminação não-uniforme por atribuir um limiar único para toda a imagem [Fig. 4.13(d)]. Os métodos de limiarização local dependem do tamanho da janela W e de constantes fixas que definem o valor do limiar para cada pixel, tornando-se inviável a aplicação de tais algoritmos como método de extração para propósitos gerais. Um outro problema relacionado aos métodos de limiarização é a segmentação, independente da imagem, em apenas duas regiões. Tal limitação prejudica a segmentação de imagens com mais de duas regiões com intensidades distintas (Fig. 4.12), ou seja, cujo histograma de intensidade dos pixels apresentam 3 ou mais modos. Além disso, os métodos de limiarização local e global atuam sobre imagens intensidade, não utilizando as informações associadas às cores. O método baseado na clusterização de cor de Jain e Yu possui um desempenho adequado quando os caracteres possuem densidade, porém fragmenta caracteres delgados devido à quantização de cor [Fig. 4.11(e)]. Além disso, o método de Jain e Yu gera um grande número de regiões (planos), dificultando a seleção do plano de caracteres utilizado pelo OCR para o reconhecimento.

O método proposto mostrou-se robusto na extração de textos em imagens contendo artefatos, caracteres de baixa densidade, plano de fundo complexo, fontes de tamanho reduzido e iluminação não-uniforme, gerando um pequeno número de *clusters* e um baixo número de recorrências ao método *K-means*.

É importante ressaltar que o método de localização proposto geralmente identifica individualmente cada caractere. Dessa forma, o algoritmo de extração é alimentado com áreas da imagem contendo apenas um caractere, em vez de áreas contendo palavras. O benefício obtido está na redução da área a ser tratada pelo algoritmo extrator, reduzindo a complexidade do plano de fundo, a iluminação não-uniforme e a quantidade de artefatos. São apresentadas nas Figs. 4.15 e 4.16 os resultados obtidos da seqüência de localização, verificação e extração proposta.

O principal motivo pelo qual todo o capítulo apresenta a extração sobre regiões contendo palavras está na avaliação dos algoritmos extratores para o pior caso, em que existe uma extensa área de plano de fundo e maior probabilidade desse possuir uma iluminação não-uniforme. Dessa forma, o algoritmo extrator mostra-se eficiente independente do método de localização utilizado, tornando-o útil aos sistemas TIE que possuem localizadores que delimitam as regiões por palavras.





(a) (b)

Fig. 4.15: Extração de uma *imagem-artificial*. (a) Resultado das etapas de localização e verificação; (b) Resultado da etapa de extração.

(a) (b)

Fig. 4.16: Extração de uma *imagem-cena*. (a) Resultado das etapas de localização e verificação; (b) Resultado da etapa de extração.

Capítulo 5

OCR e Resultados

As etapas de localização, verificação e extração visam preencher a lacuna existente entre as imagens complexas e as *imagens-documento*. A transformação das imagens complexas em binárias é a abordagem mais utilizada devido ao sucesso dos sistemas de OCR atuais no reconhecimento de caracteres em *imagens-documento*. Cada etapa de tal transformação pode ser vista na Fig. 5.1.

O método de localização proposto busca identificar e delimitar individualmente cada caractere da imagem. Tal proposta permite a localização de caracteres isolados e facilita a extração devido à redução de complexidade do plano de fundo. No entanto, os sistemas de OCR possuem uma maior taxa de reconhecimento quando alimentados com palavras completas, visto que dicionários são utilizados na determinação de caracteres duvidosos. Dessa forma, torna-se necessário agrupar os caracteres pertencentes a uma mesma palavra para alimentar o sistema de OCR.

Assim, este capítulo tem como objetivo descrever o método proposto de clusterização e normalização de caracteres anterior à entrada ao sistema de OCR. O restante do capítulo está dividido da seguinte forma. A Seção 5.1 descreve o método proposto para agrupar caracteres de uma mesma palavra. A normalização das dimensões das regiões textuais clusterizadas visando aumentar a taxa de reconhecimento dos caracteres pelo sistema de OCR está descrita na Seção 5.2. A Seção 5.3 apresenta os métodos de avaliação de desempenho geralmente utilizados em cada etapa de um sistema TIE, bem como os métodos propostos de avaliação. A Seção 5.4 compara o desempenho do

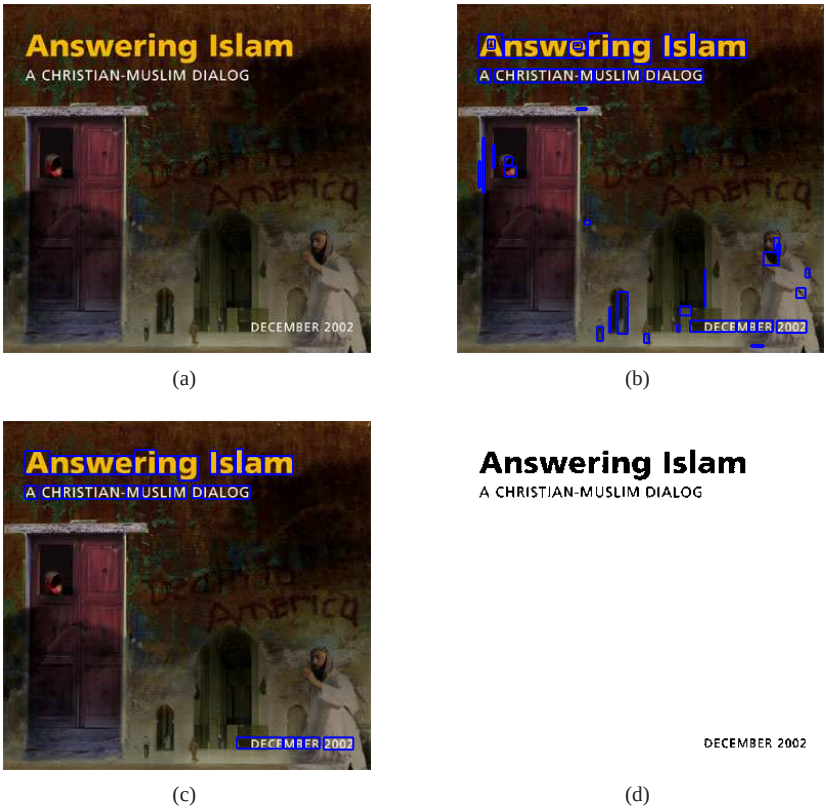


Fig. 5.1: Seqüência de etapas realizada pelo sistema TIE proposto. (a) Imagem original. (b) Resultado da etapa de localização. (c) Resultado da etapa de verificação. (d) Resultado da etapa de extração.

algoritmo de Jain e Yu ao sistema TIE proposto sobre um mesmo conjunto de imagens. A Seção 5.6 apresenta as considerações finais sobre este capítulo.

5.1 Clusterização dos Caracteres

Imagens complexas possuem palavras esparsas em diferentes orientações com caracteres de diferentes dimensões, possuindo ascendência e descendência. Tais características continuam a ser desafiadoras no processo de clusterização dos caracteres em palavras. Para reduzir a complexidade computacional e facilitar o processo de clusterização, diversos autores [12], [27],

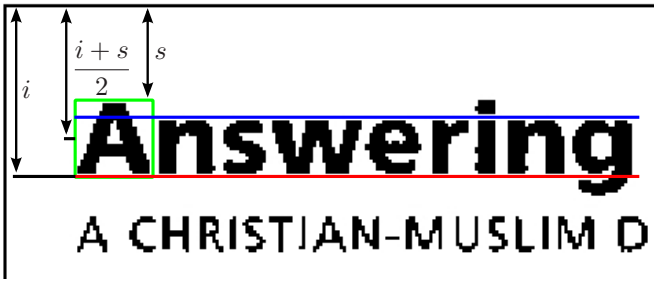


Fig. 5.2: Medidas extraídas de cada BB para a formação do vetor de clusterização.

[70] apenas clusterizam os caracteres alinhados horizontalmente. A restrição do alinhamento horizontal pode ser justificada devido à maioria das palavras estarem alinhadas em tal direção visando facilitar a leitura [27]. Neste trabalho, apenas as palavras alinhadas horizontalmente, ou com desvios inferiores a 20°do eixo horizontal, são clusterizadas corretamente e conseqüentemente reconhecidas pelo sistema de OCR.

Para agrupar os caracteres localizados individualmente propõe-se a utilização dos BBs sobreviventes após a etapa de *verificação*. Tais BBs não correspondem exatamente aos limites dos caracteres extraídos na imagem binária [Fig. 5.3(b), canto inferior direito], no entanto, não comprometem a qualidade da clusterização e evitam a redelimitação de tais caracteres por novos BBs.

Caracteres pertencentes a uma mesma palavra (horizontal) geralmente possuem BBs cujas alturas são similares e com os centros aproximadamente alinhados. Além disso, os caracteres possuem alinhamento inferior e superior com alguns desvios referentes aos caracteres com ascendência e descendência. Destaca-se na Fig. 5.2 o alinhamento inferior (segmento em vermelho) e superior (segmento em azul) da palavra '*answering*', em que apenas os caracteres com descendência ('g') e ascendência ('A', 'i') não respeitam tal alinhamento.

Devido às características mencionadas anteriormente, propõe-se a representação de cada BB por um vetor contendo 4 medidas relacionadas à dimensão e posição de cada BB na imagem.

1. Distância da borda superior do BB que delimita o caractere (ou conjunto de caracteres) à borda superior da imagem s (Fig. 5.2). Durante a clusterização dos BBs, tal medida indica os BBs que possuem os topos alinhados (parte superior).

2. Distância da borda inferior do BB que delimita o caractere (ou conjunto de caracteres) à borda superior da imagem i (Fig. 5.2). Tal medida permite quantificar o quão alinhados estão as bases (parte inferior) dos

BBs.

3. Distância do centro do BB que delimita o caractere à borda superior da imagem $\left(\frac{i+s}{2}\right)$ (Fig. 5.2). Tal medida quantifica o alinhamento dos centros dos BBs a serem clusterizados, mesmo que não possuam alinhamento superior ou inferior.

4. Distância da borda inferior à borda superior do BB que delimita o caractere $(i - s)$ (Fig. 5.2). Tal medida quantifica a altura de cada BB da imagem.

Cada BB obtido na etapa de verificação é representado por um vetor $\boldsymbol{v}_{\mathbf{BB}}$. Tal vetor é composto pelas quatro medidas supracitadas. Uma vez que o sistema TIE proposto permite a entrada de imagens de quaisquer dimensões, as medidas são normalizadas pela altura da imagem (a):

$$\boldsymbol{v}_{\mathbf{BB}} = \frac{\left[s, i, \frac{i+s}{2}, i - s\right]}{a}. \tag{5.1}$$

Os vetores $\boldsymbol{v}_{\mathbf{BB}}$ são então agrupados pela técnica de clusterização hierárquica *single-link* [31] utilizando a métrica *city-block* [23]. Optou-se pelo método *single-link* sobre os métodos *complete* e *average-link* para permitir que caracteres com um eixo de alinhamento com pequenos desvios da direção horizontal fossem clusterizados corretamente. Apresenta-se na Fig. 5.3(c) o dendrograma referente à clusterização dos BBs da Fig. 5.3(b). Cada cluster de BBs está destacado em uma cor no dendrograma da Fig. 5.3(c) e um novo BB, correspondente a cada cluster, é gerado na imagem binária como mostrado na Fig. 5.3(d).

A determinação do valor de corte no dendrograma para a determinação dos clusters não é simples, visto que o sistema TIE proposto não impõe qualquer restrição às dimensões das imagens e dos caracteres. Os vetores $\boldsymbol{v}_{\mathbf{BB}}$ estão normalizados pela altura da imagem; no entanto, para diferentes imagens, existem grandes variações da relação entre a altura dos caracteres e a altura da imagem. Dessa forma, um valor fixo de corte para o dendrograma pode resultar no não agrupamento de caracteres de uma mesma palavra com pequenos desvios de alinhamento horizontal ou agrupar mais de uma linha de texto. Para exemplificar, um valor de corte que clusterize corretamente caracteres cuja altura são similares à altura da imagem (por exemplo, *banners* de internet) pode acarretar no agrupamento de diferentes linhas de texto para caracteres relativamente pequenos à altura da imagem (por exemplo, lista das músicas em uma capa de CD). Para solucionar tal problema, optou-se por um

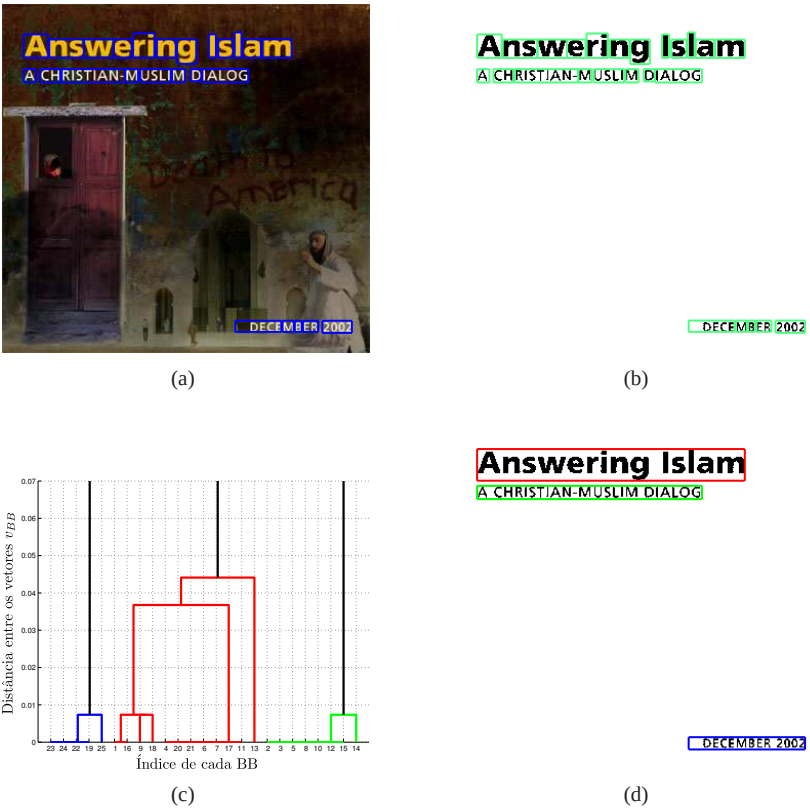


Fig. 5.3: Clusterização dos BBs após a etapa de verificação e extração. (a) Imagem original e os BBs restantes após as etapas de localização e verificação. (b) BBs restantes após as etapas de localização e verificação sobrepostos à imagem binária obtida da etapa de extração. (c) Dendrograma demonstrando o processo de clusterização dos BBs na direção horizontal. (d) Resultado da clusterização.

valor de corte capaz de clusterizar caracteres cuja altura estão próximas da altura da imagem (0,3), associado a um algoritmo de detecção e correção do agrupamento de linhas diferentes em um mesmo cluster. Tal algoritmo efetua os seguintes passos:

1. Entre os BBs de um cluster, busca-se o que possui o maior valor s ($s_{\max}$).

2. Caso existam BBs cujos valores de i sejam menores do que $s_{\max}$, um processo diviso é realizado, gerando-se dois clusters (c_{i} e c_{r}): um con-

tendo os BBs cujos valores de i são menores do que $s_{\max}$ (c_{i}) e outro contendo os BBs restantes (c_{r}).

3. O processo é recursivamente reiniciado do passo 1 sobre o cluster c_{i}, até que a condição avaliada no passo 2 não seja satisfeita.

A etapa anterior produz clusters de caracteres alinhados horizontalmente; no entanto, podem existir palavras alinhadas pertencentes a frases diferentes. Para evitar tal ocorrência, um processo divisivo horizontal é realizado utilizando uma heurística de que caracteres de uma mesma palavra geralmente estão separados por distâncias similares e inferiores à altura do caractere. Dessa forma, a separação de frases ocorre da seguinte forma:

1. Calcula-se a média da altura dos BBs de cada cluster.

2. Selecionam-se os BBs cuja altura são menores do que duas vezes o valor da média do cluster. Tal procedimento é realizado para reduzir a variância de altura dentro do cluster, obtendo uma média de altura próxima dos caracteres que não possuem ascendência ou descendência.

3. Calcula-se a média dos BBs selecionados de cada cluster.

4. Varre-se horizontalmente o conjunto de BBs de cada cluster, dividindo-o nos pontos em que os BBs possuem uma distância horizontal maior do que a média do passo 3.

O processo divisivo pode acarretar a divisão de palavras de uma mesma frase; no entanto é improvável a separação de caracteres de uma mesma palavra. Sendo assim, cada palavra é capturada individualmente sem qualquer perda de informação.

Após a clusterização, recorta-se cada cluster de caracteres da imagem [regiões delimitadas na Fig. 5.3(d)] para alimentar o sistema de OCR, como ilustrado na Fig. 5.4.

Devido à variedade de dimensões dos caracteres, faz-se necessário uma normalização das dimensões de cada região textual extraída para alimentar o sistema de OCR. Tal normalização é descrita em detalhes na próxima seção.

5.2 Normalização das Regiões Localizadas

Os sistemas de OCR convencionais são projetados para reconhecer texto em documentos escaneados a uma resolução de 200 a 300 *dots per inch* (dpi), resultando em caracteres com pelo menos 30 pixels de altura. Devido ao número de pixels que compõe cada caractere, a taxa de reconhecimento

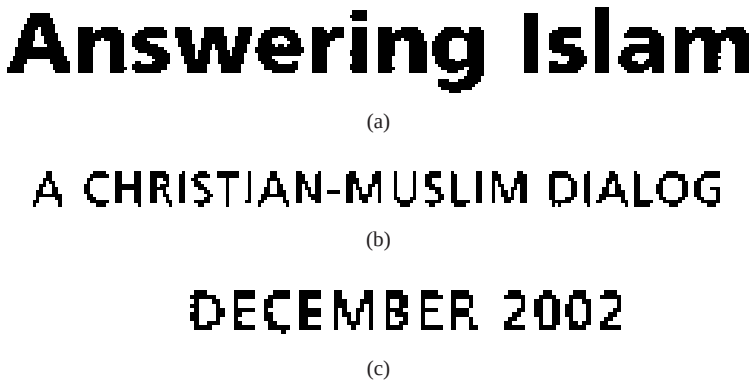


Fig. 5.4: Linhas de texto clusterizadas e extraídas para alimentação do sistema de OCR.

pelos sistemas de OCR é prejudicada quando a resolução é reduzida a valores inferiores a 300 dpi [61].

Os caracteres extraídos de imagens complexas possuem alturas que variam de 5 a 300 pixels; assim, faz-se necessário um pré-processamento de normalização da altura das palavras extraídas. Tal normalização é realizada para aumentar a taxa de reconhecimento de caracteres cuja altura é menor do que 30 pixels e reduzir complexidade computacional aos que possuem altura maior do que 100 pixels [41]. Testes realizados sobre documentos escaneados mostraram que o aumento das dimensões das linhas de texto acima de 50 pixels de altura não resultam em melhoria da taxa de reconhecimento dos caracteres[1].

Como último estágio do sistema TIE, devido às restrições impostas pelos sistemas de OCR, realizou-se o redimensionamento de todas as linhas de texto obtidas no processo de clusterização para a altura de 50 pixels (mantendo-se a razão de aspecto). Visando determinar qual tipo de interpolação produz a melhor imagem para o reconhecimento dos caracteres, realizou-se o redimensionamento utilizando as interpolações *nearest neighbor*, bilinear e bicúbica, como mostrado na Fig. 5.5.

As interpolações bilinear e bicúbica produziram imagens com taxas de reconhecimento similares, porém superiores à interpolação *nearest neighbor* para o sistema de OCR Tesseract. Optou-se assim pela normalização utilizando a interpolação bilinear (computacionalmente menos complexa do que a bicúbica) associada ao sistema de OCR Tesseract [61].

A Tabela 5.2 apresenta os resultados obtidos do sistema de OCR Tes-

[1]Tais resultados são obtidos utilizando o sistema de OCR Tesseract mantido pela Google Inc.

(a) (b)

(c)

Fig. 5.5: Interpolação e normalização das regiões clusterizadas. (a) Interpolação *nearest neighbor*. (b) Interpolação bilinear. (c) Interpolação bicúbica.

seract para as três linhas de texto extraídas da Fig. 5.4.

Tabela 5.1: Resultados de reconhecimento das linhas de texto da Fig. 5.4 utilizando as interpolações bilinear, *nearest neighbor* e bicúbica

Nearest Neighbor	**Bilinear**	**Bicúbica**
Answering Islam	Answering Islam	Answering Islam
A CHRISTIAN-I'·».·?I LJSLIM DIALOG	A CHRISTIAN-MUSLIM DIALOG	A CHRISTIAN-MUSLIM DIALOG
DECEM B EF'. 2·I]·IZl2	DECEMBER 2002	DECEMBER 2002

5.3 Avaliação de Desempenho

Existem diversas dificuldades relacionadas à avaliação de desempenho em praticamente todas as áreas de visão computacional e reconhecimento de padrões - *computer vision and pattern recognition* (CVPR) [32]. Embora existam esforços para criar medidas objetivas para a comparação de desempenho de algoritmos em CVPR, muito pouco tem sido apresentado para o problema da extração da informação textual em imagens e vídeos. Esta seção comenta os principais problemas dos métodos de avaliação de desempenho dos algoritmos relacionados a um sistema TIE.

5.3.1 Métodos de Avaliação - Etapa de Localização

A avaliação de desempenho de um determinado algoritmo de identificação das regiões textuais é realizado após as etapas de localização e verificação. Tal procedimento é justificado pelo objetivo da etapa de verificação, que visa apenas realizar uma filtragem refinada das regiões textuais identifi-

cadas pelo algoritmo de localização. Dessa forma, a etapa de verificação é geralmente desacoplada da localização para fins didáticos; no entanto, é tecnicamente parte integrante do processo de localização das regiões textuais.

Ainda não existe um procedimento formalizado de avaliação de desempenho padrão para a localização de caracteres devido aos seguintes problemas:

- **Dados *Ground-Truth* (referência).** Para avaliar um algoritmo de localização é necessário uma marcação de referência, geralmente fornecida por meio de BBs que delimitam a região textual. Tal marcação geralmente é realizada sobre cada palavra da imagem, dificultando a avaliação caso a saída do algoritmo de localização seja a delimitação de linhas textuais completas ou caracteres individuais. Uma vez que cada algoritmo de localização possui um formato de saída, não é simples criar uma base de dados *ground-truth* que contemple todas as técnicas.

- **Medida de Desempenho.** Uma vez determinada o formato da referência, o novo desafio torna-se determinar o método de comparação entre os BBs obtidos pelo algoritmo de localização e os BBs de referência.

- **Aplicação.** Os objetivos diferem de aplicação para aplicação, resultando em uma variedade de avaliações de desempenho. Em algumas aplicações, o número de falsos-positivos não é importante, contanto que as regiões textuais sejam identificadas. Outras visam apenas a extração dos textos mais relevantes ou de maiores dimensões. Assim, os objetivos de cada aplicação é que determinam os pesos aplicados ao número de caracteres reconhecidos e de falsos-positivos durante a avaliação de desempenho.

- **Base de Dados.** A avaliação de desempenho visa definir os algoritmos e técnicas mais eficientes; no entanto, não existe uma base de dados pública que permita a comparação justa entre os algoritmos. As comparações são realizadas sobre base de dados particulares que podem estar otimizadas para uma determinada aplicação, tornando-se complicado definir a superioridade de desempenho de uma técnica sobre outra.

Devido à diversidade de métodos de avaliação de desempenho, o algoritmo de localização proposto é avaliado por 3 métodos: *área de intersecção*, *ICDAR* e *mínimo BB* (proposto). Cada método é descrito em detalhes nas seções seguintes.

5.3.1.1 Método Área de Intersecção

O método área de intersecção baseia-se no número de pixels de intersecção entre os BBs de referência e os BBs localizados por cada algoritmo.

O desempenho de um algoritmo de localização submetido a esse método de avaliação é obtido utilizando os seguintes passos:

- Calcula-se a soma das áreas de intersecção A_{int} entre os BBs localizados BB_{loc} e os BBs de referência BB_{gt}. Destaca-se em cinza na Fig. 5.6 a área de interseção A_{int} entre um BB_{loc} e um BB_{gt}.

- Calcula-se a área localizada total A_{loc}, determinada pelo número de pixels interno aos BB_{loc}.

- Calcula-se a área de referência (*ground-truth*) total A_{gt}, definida pelo número de pixels interno aos BB_{gt} (Fig. 5.6).

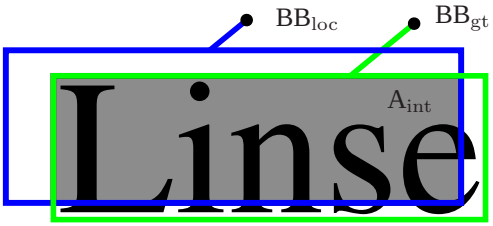


Fig. 5.6: BBs utilizados pelo método área de intersecção durante a avaliação de desempenho.

Uma vez determinado os valores de A_{int}, A_{loc} e A_{gt}, o método define as medidas de precisão e o reconhecimento por meio das seguintes equações:

$$P_{\text{area}} = \frac{A_{\text{int}}}{A_{\text{loc}}} \tag{5.2}$$

e

$$R_{\text{area}} = \frac{A_{\text{int}}}{A_{\text{gt}}} \tag{5.3}$$

em que a precisão P_{area} avalia quanto das áreas localizadas são textuais, enquanto o reconhecimento R_{area} quantifica a área textual identificada corretamente na imagem.

Em (5.2) e (5.3), notam-se duas desvantagens para este método de avaliação. A precisão de 100% é alcançada apenas se os BB_{gt} e os BB_{loc} se sobrepuserem perfeitamente. Além disso, caso um BB localizado possua as dimensões da imagem, o método de avaliação indica uma taxa de reconhecimento de 100%, visto que todas as áreas textuais estão contempladas por um BB_{loc}.

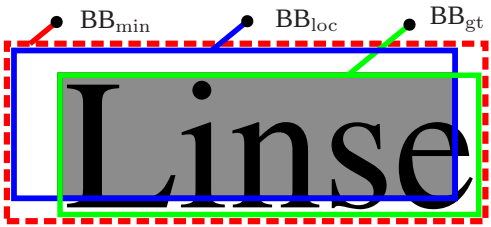


Fig. 5.7: BBs utilizados pelo método ICDAR durante a avaliação de desempenho.

5.3.1.2 Método ICDAR

A *International Conference on Document Analysis and Recognition* (ICDAR) realizou um campeonato para determinar o melhor algoritmo de localização submetido por diversos candidatos. Para tal campeonato foi proposto um método de avaliação em que os BBs de referência demarcam as *palavras* de cada imagem. A avaliação de desempenho é realizada por meio de 3 BBs (Fig. 5.7):

- BB de referência (BB$_{\text{gt}}$).
- BB localizado pelo algoritmo proposto (BB$_{\text{loc}}$).
- BB que engloba o BB localizado (BB$_{\text{loc}}$) e o BB referência (BB$_{\text{gt}}$), denominado BB$_{\text{min}}$.

Visto que é improvável coincidir perfeitamente o resultado da localização (BB$_{\text{loc}}$) com a referência (BB$_{\text{gt}}$) criada manualmente, os organizadores do evento criaram um valor de casamento m_{p}. Tal valor é obtido pela divisão da área de intersecção entre o BB$_{\text{loc}}$ e BB$_{\text{gt}}$ (área em cinza da Fig. 5.7) e o mínimo retângulo que os contém (BB$_{\text{min}}$ - retângulo tracejado em vermelho).

$$m_{\text{p}} = \frac{\text{BB}_{\text{loc}} \cap \text{BB}_{\text{gt}}}{\text{BB}_{\text{min}}}. \tag{5.4}$$

Para cada BB$_{\text{loc}}$, do conjunto E de BB$_{\text{loc}}$, busca-se no conjunto R de BB$_{\text{gt}}$ aquele de melhor casamento, ou seja, aquele que apresenta o maior valor de m_{p} em (5.4). Assim,

$$m(\text{BB}_{\text{loc}},R) = \max m_{\text{p}}(\text{BB}_{\text{loc}},\text{BB}_{\text{gt}}) \mid \text{BB}_{\text{gt}} \in R. \tag{5.5}$$

Obtidos os BB$_{\text{gt}}$ que possuem melhor casamento com os BB$_{\text{loc}}$, definem-se as equações de precisão e reconhecimento desse método. Portanto,

$$P_{\mathrm{ICDAR}} = \frac{\sum_{\mathrm{BB}_{\mathrm{loc}} \in E} m(\mathrm{BB}_{\mathrm{loc}}, R)}{|E|} \tag{5.6}$$

e

$$R_{\mathrm{ICDAR}} = \frac{\sum_{\mathrm{BB}_{\mathrm{gt}} \in R} m(\mathrm{BB}_{\mathrm{gt}}, E)}{|R|} \tag{5.7}$$

onde $|E|$ e $|R|$ são o número de BBs localizados ($\mathrm{BB}_{\mathrm{loc}}$) e referência ($\mathrm{BB}_{\mathrm{gt}}$), respectivamente.

Para compor os resultados de precisão e reconhecimento em uma única medida de qualidade a ICDAR adota o *f-score* (f) padrão. Assim,

$$f = \frac{1}{\frac{\alpha}{P_{\mathrm{ICDAR}}} + \frac{1-\alpha}{R_{\mathrm{ICDAR}}}}. \tag{5.8}$$

Os pesos relacionados a cada medida (P_{ICDAR} e R_{ICDAR}) são definidas pela constante α. A conferência atribui a tal constante o valor $0{,}5$, ou seja, pesos iguais para ambos, precisão e reconhecimento.

A grande desvantagem do método de avaliação proposto pela ICDAR é a dificuldade em interpretar os resultados. Simon Lucas [45] destaca que um reconhecimento de $0{,}9$ pode significar que todos os BBs de referência ($\mathrm{BB}_{\mathrm{gt}}$) são identificados com uma precisão de 90%, ou que 90% dos $\mathrm{BB}_{\mathrm{gt}}$ são perfeitamente identificados e 10% completamente perdidos. Além disso, tal método de avaliação somente tem significado quando o formato de saída[2] do algoritmo sob avaliação coincide com o formato dos BBs de referência, não podendo assim ser utilizado de maneira genérica.

5.3.1.3 Método Proposto de Avaliação - Mínimo BB

Analisando as deficiências dos métodos apresentados anteriormente propõe-se um método de avaliação visando atender as seguintes especificações:

1. O método deve ser capaz de avaliar algoritmos de localização cuja saída são delimitações de palavras ou caracteres.

2. O método não deve atribuir qualquer perda de desempenho pela diferença de posição entre as caixas limítrofes de referência e as localizadas devido à dificuldade em localizar regiões exatamente coincidentes com a referência.

[2] Caractere, palavras ou linhas de texto.

3. O método de avaliação deve verificar se as caixas limítrofes obtidas pelo localizador contemplam os caracteres da imagem de maneira satisfatória. É improvável que caracteres identificados parcialmente sejam reconhecidos corretamente pelo sistema de OCR. Logo, o método de avaliação deve levar em consideração apenas os caracteres com alguma chance de reconhecimento.

A base do método de avaliação proposto está na demarcação dos BBs de referência BB_{gt}. Cada BB_{gt} delimita apenas um caractere da imagem (Fig. 5.8 em verde), o que explica a denominação do método: *mínimo BB*. A delimitação individual dos caracteres atende a especificação 1, uma vez que diferentes algoritmos de localização, independente do tipo de saída (palavras ou caracteres), podem utilizar tal referência. Independente do tipo de saída do algoritmo, pode-se determinar os caracteres localizados corretamente por meio da intersecção dos BB_{gt} e os BBs resultantes do processo de localização. Tal capacidade permite atender a especificação 3, pois é possível determinar a área corretamente identificada de cada caractere.

Para atender a especificação 3, exige-se que as demarcações da referência (BB_{gt}) delimitem cada caractere individualmente. Tal demarcação ainda facilita a implementação da especificação 1, visto que para avaliar algoritmos de localização cuja saída são palavras, verificam-se diretamente os BBs de referência (BB_{gt}) englobados pelo BB obtido do algoritmo de localização (BB_{loc}).

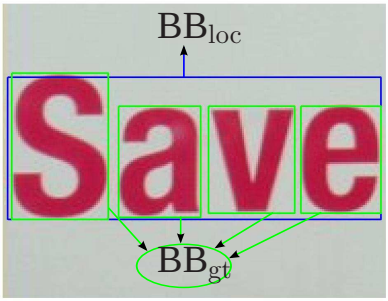


Fig. 5.8: BBs de referência (BB_{gt}) para o método mínimo BB.

A contagem dos BB_{gt} identificados pelos BB_{loc} é baseada em 2 critérios que são avaliados seqüencialmente: BB_{aceito} e $BB_{reconhecido}$. Um BB_{loc} é considerado como BB_{aceito} se 40% dos seus pixels possuem intersecção

com os pixels dos $\mathrm{BB_{gt}}$. Caso isso não ocorra, o $\mathrm{BB_{loc}}$ é considerado falso-positivo. A condição de aceitação é criada para determinar os $\mathrm{BB_{loc}}$ que não possuem caracteres em seu interior ou que 60% da sua área contemple apenas regiões não textuais.

Caso o $\mathrm{BB_{loc}}$ seja considerado um $\mathrm{BB_{aceito}}$, buscam-se os $\mathrm{BB_{gt}}$ que possuem mais de 70% dos seus pixels contemplados pelo $\mathrm{BB_{loc}}$. Tais $\mathrm{BB_{gt}}$ são considerados corretamente localizados e denominados $\mathrm{BB_{reconhecido}}$, enquanto os outros como não-localizados. A condição de aceitação do BB ($\mathrm{BB_{aceito}}$) associada à condição de reconhecimento ($\mathrm{BB_{reconhecido}}$) contemplam as restrições 2 e 3. Tais condições não associam perda de desempenho para variações de posição entre o $\mathrm{BB_{loc}}$ e $\mathrm{BB_{gt}}$, e evitam que caracteres sem qualquer chance de reconhecimento pelo sistema de OCR sejam considerados corretamente localizados.

Uma vez definidos os $\mathrm{BB_{reconhecido}}$ e os falsos-positivos, pode-se determinar a avaliação de precisão (P_{minBB}) e reconhecimento (R_{minBB}) do método mínimo BB. Assim,

$$P_{\mathrm{minBB}} = \frac{\text{Nº de } \mathrm{BB_{reconhecido}}}{\text{Nº de } \mathrm{BB_{reconhecido}} + \text{Nº de Falsos-Positivos}} \tag{5.9}$$

e

$$R_{\mathrm{minBB}} = \frac{\text{Nº de } \mathrm{BB_{reconhecido}}}{\text{Nº Total de } \mathrm{BB_{gt}}}. \tag{5.10}$$

5.3.2 Métodos de Avaliação - Etapas de Extração e OCR

A avaliação da etapa de extração geralmente é realizada de forma indireta, devido à ausência de figuras de mérito que indiquem a qualidade do extrator. Uma vez que os algoritmos de extração são responsáveis por adequar a imagem (binarização) para o reconhecimento por um sistema de OCR, é comum inferir-se a qualidade do extrator pela taxa de reconhecimento obtida pelo sistema de OCR. Tal método é questionável, visto que um conjunto imagens binárias alimentando diferentes sistemas de OCR resultam em diferentes taxas de reconhecimento. Dessa forma, não se pode quantificar até que ponto a melhoria da taxa de reconhecimento está associada ao desempenho do algoritmo extrator. Tal método apenas define o desempenho do algoritmo de extração associado a determinado OCR, dificultando a comparação de desempenho entre diferentes algoritmos extratores.

Neste trabalho, a avaliação do algoritmo de extração foi realizada por meio de uma medida de discrepância denominada probabilidade de erro (P_{E}), como descrito no Capítulo 4. Tal medida é utilizada para verificar o desempenho dos algoritmos de segmentação para o caso de duas classes (binária)

e possui a vantagem de quantificar a qualidade do extrator de maneira direta, independente do sistema de OCR utilizado posteriormente. No entanto, a medida P_{E} exige imagens binárias de referência, cuja criação é um processo laborioso e subjetivo.

5.4 Resultados Experimentais

Os algoritmos apresentados neste trabalho, bem como as interfaces criadas durante o desenvolvimento do sistema TIE proposto, são implementados em Matlab R2007a. As implementações e testes são realizadas em um computador possuindo um processador Intel core 2 duo de $1{,}6$ GHz, 1 Gb de RAM e sistema operacional Ubuntu 8.04.

Para a avaliação de desempenho do sistema TIE, 80 imagens coloridas de diferentes características são selecionadas para o conjunto de teste. Tal conjunto é composto por 15 capas de livro, 15 capas de CD, 15 *banners* da internet e 35 *imagens-cena*. Dentre as *imagens-cena*, 20 são do campeonato promovido pela ICDAR e 15 de fotografias da Universidade Federal de Santa Catarina (UFSC). As dimensões das imagens do conjunto de teste variam entre 196×138 até 1500×1500 pixels (largura $\times$ altura), contendo uma ampla variedade de planos de fundo e texto. O conjunto de *imagens-teste* possui no total 5090 caracteres de diferentes orientações, fontes, cores e cujas dimensões variam entre 3×3 até 265×285 pixels (largura $\times$ altura).

Embora o conjunto de *imagens-teste* possua uma extensa variedade de dimensões, orientações, cores e fontes de caracteres embutidos em diversos tipos de plano de fundo, nenhuma heurística ou limiar foram adaptados manualmente. A proposta do trabalho é localizar e extrair caracteres de imagens independente da aplicação. Dessa forma, todos os resultados experimentais do sistema TIE proposto foram obtidos utilizando os mesmos parâmetros.

A etapa de verificação do sistema TIE proposto é implementado de duas formas para propósito de comparação: uma utilizando o classificador SVM, denominado *proposto (SVM)* e outra utilizando limiares obtidos manualmente dos histogramas de cada atributo, denominado *proposto (Histograma)*, como descrito no Capítulo 3. O desempenho das duas formas de implementação do sistema TIE são comparados ao método de Jain e Yu utilizando os três métodos de avaliação. Os resultados de desempenho de cada algoritmo são obtidos através de uma interface de avaliação de sistemas TIE que integra algoritmos de localização, verificação, extração e avaliação, permitindo compará-los sobre o mesmo conjunto de imagens. A interface pode ser visualizada no Apêndice D.

Na Fig. 5.9 são apresentados os resultados da localização das regiões textuais utilizando o método de avaliação mínimo BB para os 3 algoritmos

implementados. Nota-se que a taxa de reconhecimento de caracteres dos dois

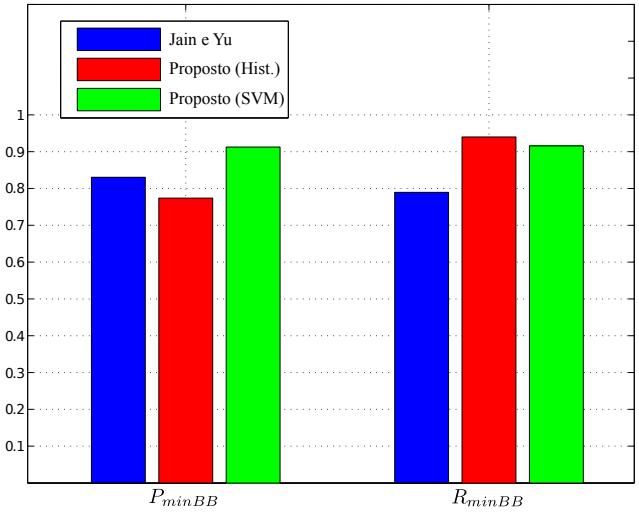


Fig. 5.9: Avaliação de desempenho dos algoritmos de localização utilizando o método mínimo BB.

algoritmos propostos supera em mais de 10% o método de Jain e Yu; no entanto a precisão do método de Jain e Yu supera em cerca de 5% o método *proposto (Histograma)*. Na Tabela 5.2 é apresentado o desempenho dos algoritmos utilizando o método mínimo BB para cada grupo de imagens testadas.

Tabela 5.2: Desempenho dos algoritmos de localização pelo método mínimo BB

Método Proposto de Avaliação - mínimo BB						
Imagens	**Métodos**	**Nº Caracteres**	**Reconhecidos**	**Falsos-Positivos**	$\mathbf{R}_{\mathrm{minBB}}$	$\mathbf{P}_{\mathrm{minBB}}$
	Jain e Yu		500	61	67,57%	89,13%
Banner WWW	**Proposto (Hist.)**	740	658	129	88,92%	83,61%
	Proposto (SVM)		655	45	88,51%	93,57%
	Jain e Yu		670	137	67,95%	83,02%
Capa Livro	**Proposto (Hist.)**	986	908	152	92,09%	85,66%
	Proposto (SVM)		888	55	90,06%	94,17%
	Jain e Yu		2033	246	84,46%	89,21%
Capa CD	**Proposto (Hist.)**	2407	2350	213	97,63%	91,69%
	Proposto (SVM)		2217	132	92,11%	94,38%
	Jain e Yu		299	223	79,10%	57,28%
ICDAR	**Proposto (Hist.)**	378	370	294	97,88%	55,72%
	Proposto (SVM)		371	77	98,15%	82,81%
	Jain e Yu		516	153	89,12%	77,13%
UFSC	**Proposto (Hist.)**	579	498	610	86,01%	44,95%
	Proposto (SVM)		529	137	91,36%	79,43%
	Jain e Yu		4018	820	78,94%	83,05%
Conjunto de Teste	**Proposto (Hist.)**	5090	4784	1398	93,99%	77,39%
	Proposto (SVM)		4660	446	91,55%	91,27%

Avaliando a Tabela 5.2, nota-se que os algoritmos propostos possuem uma taxa de reconhecimento significativamente superior ao método de Jain e Yu. Observa-se da Fig. 5.9 que a utilização do classificador SVM durante a etapa de verificação promove uma perda de aproximadamente 2,5% na taxa de reconhecimento de caracteres quando comparada à implementação via histogramas. No entanto, tal implementação aumenta a precisão em aproximadamente 14%, melhorando consideravelmente a robustez no processo de definição das regiões textuais.

O resultado obtido pela avaliação baseada na área de intersecção é similar ao método mínimo BB, como mostrado na Fig. 5.10. O método de avaliação indica que as duas implementações propostas possuem desempenho superior ao método de Jain e Yu, tanto em precisão quanto em taxa de reconhecimento. Além disso, torna-se evidente que a implementação *proposto (SVM)* apresenta uma maior robustez na determinação das áreas textuais (precisão) sobre a implementação via histogramas. Tal robustez é obtida às custas de uma taxa de reconhecimento menor do que a implementação *proposto (Histogramas)*. No entanto, exibe uma perda da taxa reconhecimento de aproximadamente $0{,}01$ quando comparado a implementação via histogramas, como havia destacado o método mínimo BB. Apresenta-se na Fig. 5.10 os valores de desempenho obtidos pelo método de avaliação área de intersecção. Os resultados de desempenho dos algoritmos sob o método de avaliação ICDAR estão representados graficamente na Fig. 5.11. Os resultados indicam, similarmente aos métodos de avaliação anteriores, uma superioridade das abordagens propostas sobre o método de Jain e Yu. A implementação utilizando o classificador SVM possui uma maior precisão às custas de uma redução da taxa de reconhecimento quando comparada à implementação via histogramas.

A complexidade computacional do sistema TIE proposto é dependente da resolução da imagem, do número de caracteres presentes e da complexidade do plano de fundo em que os caracteres estão embutidos, não permitindo assim estimar o número de operações realizadas. No entanto, pode-se inferir a complexidade das etapas por meio da média de tempo despendida à realização de cada etapa. O sistema TIE proposto (SVM) destinou em média 10,11% do tempo para a etapa de localização, 82,23% para a etapa de verificação e 7,66% para a etapa de extração.

Na seção seguinte, são apresentados exemplos de localização e extração do sistema TIE proposto (SVM) sobre imagens pertencentes ao conjunto de teste.

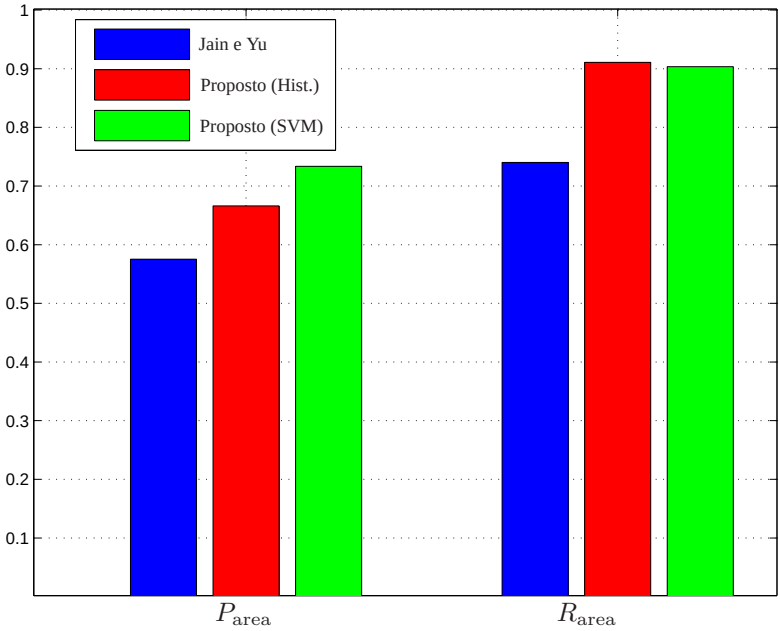


Fig. 5.10: Desempenho dos algoritmos de localização pelo método área de intersecção.

5.5 Exemplos de Resultados

Esta seção mostra os resultados obtidos pelo sistema TIE proposto sobre algumas imagens do conjunto de teste. Para cada *imagem-teste* são apresentadas quatro figuras:

a) Imagem original.

b) Resultado após as etapas de localização e verificação.

c) Desempenho do sistema TIE proposto sob método de avaliação mínimo BB, em que os caracteres considerados localizados, não-localizados e falsos-positivos são destacados em verde, vermelho e amarelo, respectivamente. Esta figura visa demonstrar que o método de avaliação mínimo BB classifica cada BB obtido pelo sistema TIE como contendo ou não caracteres, além de identificar os caracteres não-localizados. Tal classificação não é obtida pelos métodos de avaliação ICDAR e área de intersecção. Esses apenas apresentam os valores de precisão e taxa de reconhecimento para cada imagem.

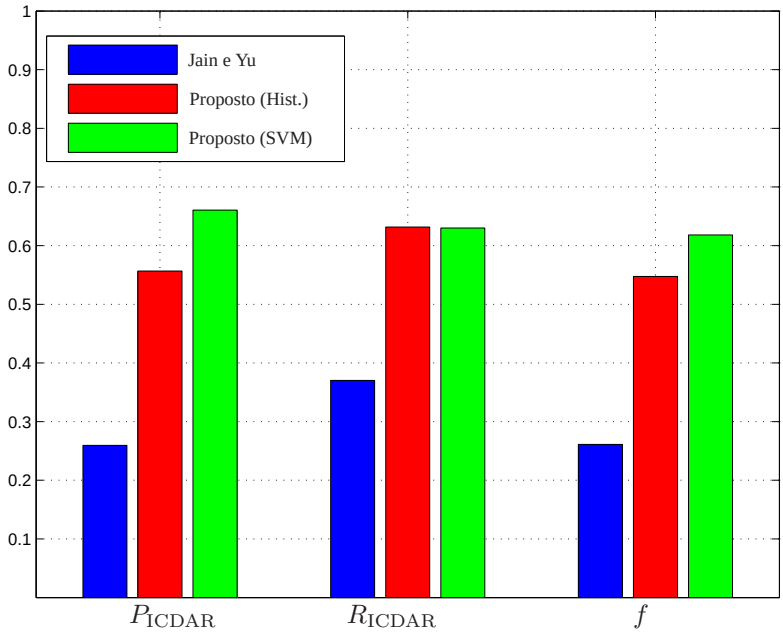


Fig. 5.11: Desempenho dos algoritmos de localização pelo método ICDAR.

d) Resultado após a etapa de extração utilizando o método proposto *K-means recorrente*.

Os exemplos que possuem texto passível de reconhecimento pelo sistema de OCR Tesseract apresentam uma tabela contendo as palavras existentes na imagem e as reconhecidas pelo sistema TIE proposto (SVM) para três tipos de interpolação (*nearest neightbor*, bilinear e bicúbica). Os sistemas de OCR geralmente são incapazes de reconhecer palavras não alinhadas horizontalmente, com rupturas e distorção de perspectiva, como, por exemplo, caracteres artísticos.

5.5.1 Exemplo 1

(a) (b)

(c) (d)

Fig. 5.12: *Imagem-cena* contendo caracteres em um plano de fundo texturizado. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.3: Reconhecimento do texto da imagem referente à Fig. 5.12

Tag Real	**OCR (bilinear)**	**OCR (nearest)**	**OCR (bicubic)**
COLONNADE E14	COLONNADE E14	COLONNADE E14	COLONNADE E14
THE SOUTH	THE SOUTH	THE SOUTH	THE SOUTH

5.5.2 Exemplo 2

Fig. 5.13: Imagem-artificial contendo fontes possuindo ascendência e descendência, além de caracteres com tamanho reduzido. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

5.5.3 Exemplo 3

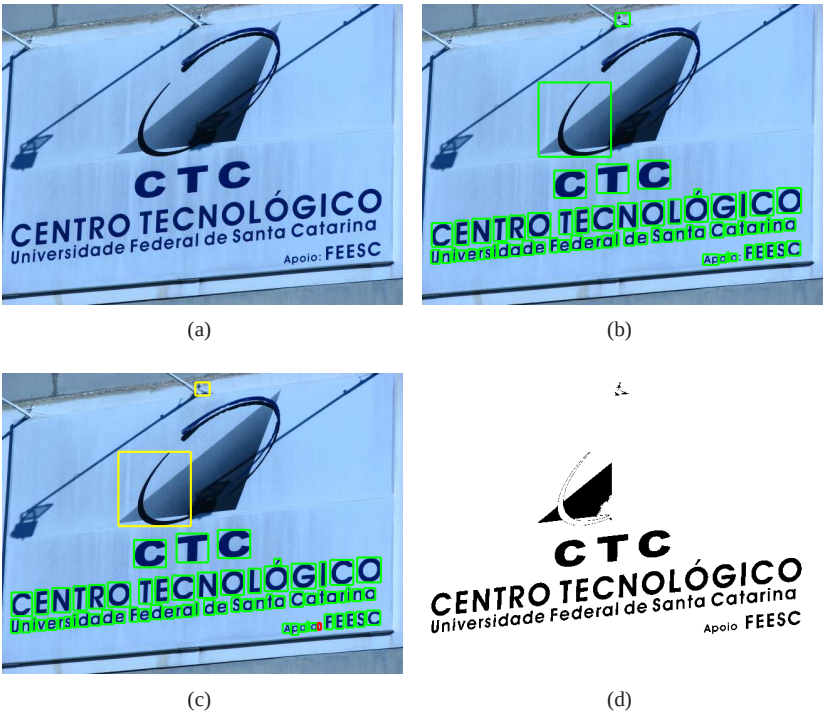


Fig. 5.14: Imagem-cena contendo caracteres de diferentes dimensões e com um pequeno desvio de orientação do eixo horizontal. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

5.5.4 Exemplo 4

(a) (b) (c)

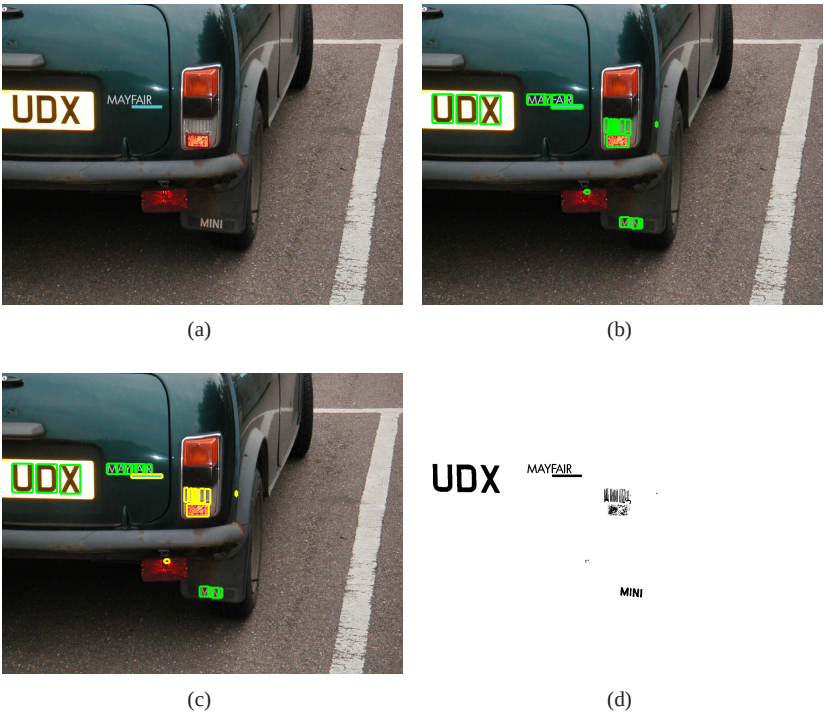
(d)

Fig. 5.15: Imagem-artificial contendo caracteres com orientações distintas e variadas dimensões sobre um plano de fundo complexo. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.4: Reconhecimento do texto da imagem referente à Fig. 5.15

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
het	HET	HET	HET
speelveld	SPEELVELD	SPEELVELD	SPEELVELD
van	VAN	VAN	VAN
de	DE	DE	DE
cio	CIO	CIO	CIO
omgaan	O M C A A N	O M C A A N	O M C A A N
met	M E T	hl E T	hl E T
dilemma's	D |. E M M A 7 S	D I. E M M A- '.| S	D |. E M M A 7 S

5.5.5 Exemplo 5



(a) (b)

(c) (d)

Fig. 5.16: Imagem-cena contendo caracteres de variadas dimensões. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Resultado do método de avaliação mínimoBB. (d) Resultado da etapa de extração.

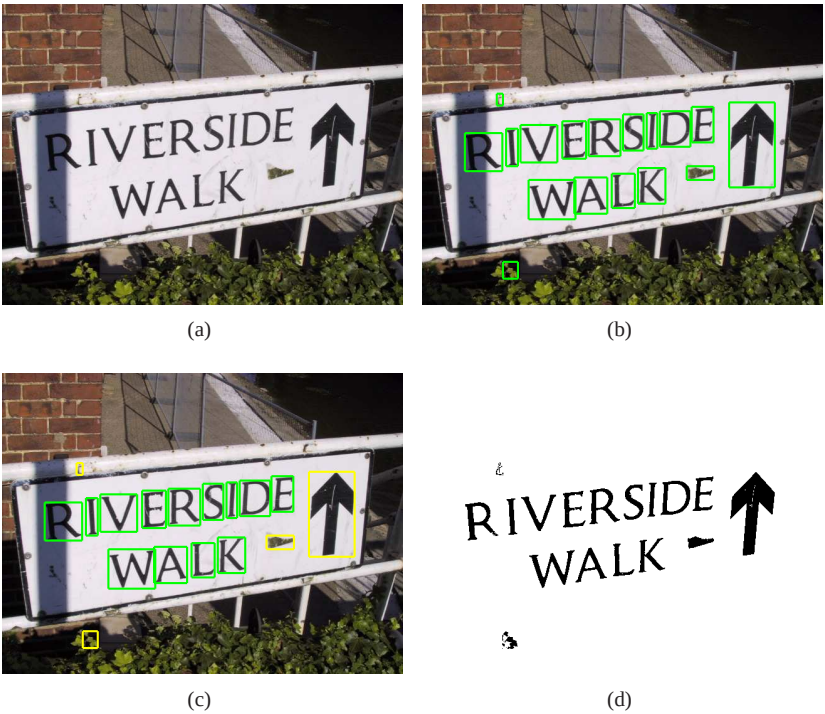
Tabela 5.5: Reconhecimento do texto da imagem referente à Fig. 5.16

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
udx	UDX	UDX	UDX
mayfair	MAYFAIR	MAYFAIR	MAYFAIR
mini	MINI	MINI	MINI

5.5.6 Exemplo 6

Fig. 5.17: Imagem-artificial contendo caracteres de variadas dimensões e com distorção de perspectiva. Nota-se o erro de avaliação cometido pelo método mínimo BB sobre a palavra 'suaempresa.com'. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

5.5.7 Exemplo 7



(a) (b)

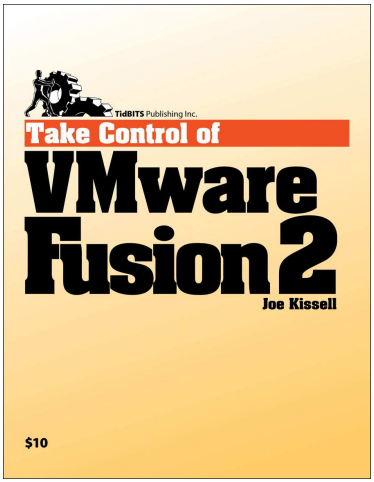
(c) (d)

Fig. 5.18: Imagem-cena contendo caracteres não alinhados horizontalmente sob variação de iluminação (‘R’ da palavra ‘RIVERSIDE’). (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

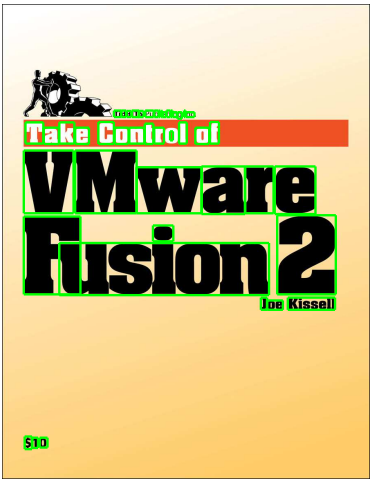
Tabela 5.6: Reconhecimento do texto da imagem referente à Fig. 5.18

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
riverside walk	R IVERSIDE WALK	R IV ERSIDE WALK	R IV ERSIDE WALK

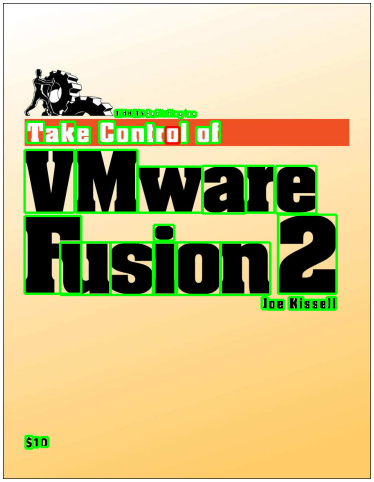
5.5.8 Exemplo 8



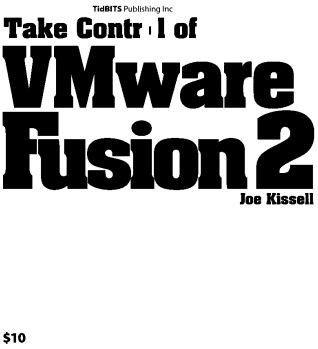
(a)



(b)



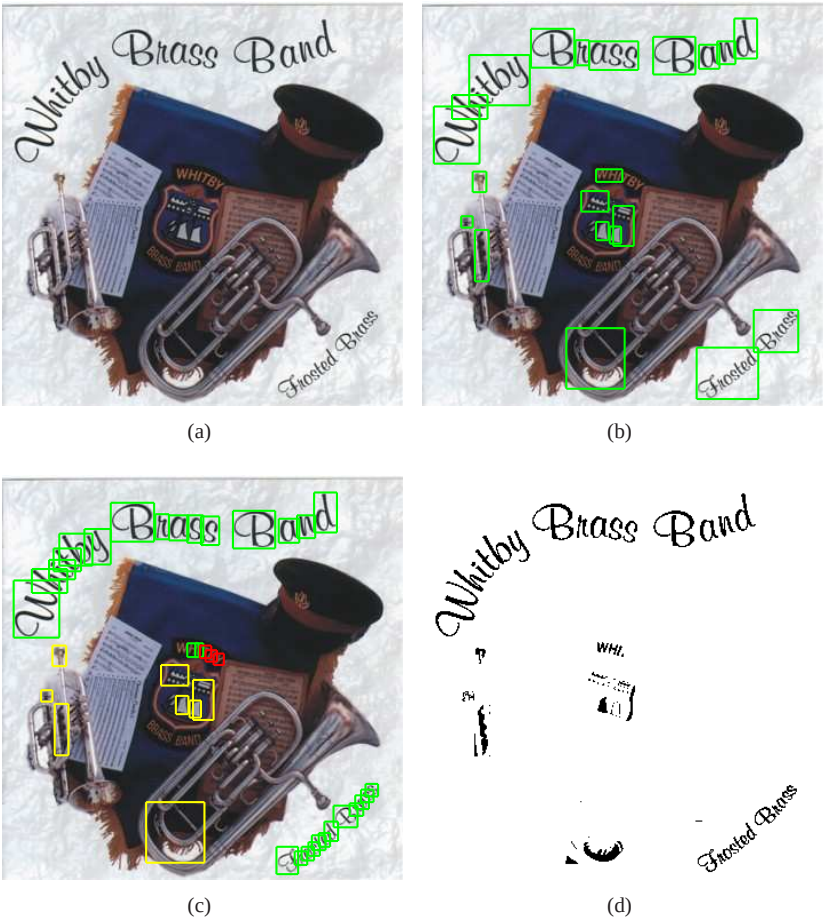
(c)



(d)

Fig. 5.19: Imagem-artificial contendo caracteres de variadas dimensões, além de possuir contraste positivo e negativo na mesma imagem. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

5.5.9 Exemplo 9

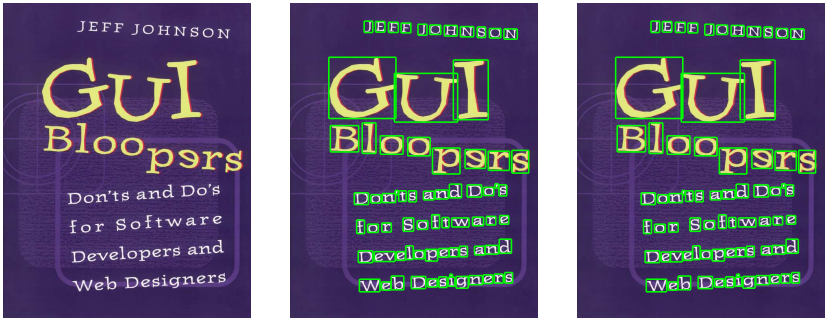


(a) (b)

(c) (d)

Fig. 5.20: Imagem-artificial contendo caracteres sem alinhamento. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

5.5.10 Exemplo 10



(a) (b) (c)

(d)

Fig. 5.21: Imagem-artificial contendo caracteres de variadas dimensões e orientações. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.7: Reconhecimento do texto da imagem referente à Fig. 5.21

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
jeff johnson	JEFF JOHNSON	JEFF JOHNSON	JEFF JOHNSON
gui	GUI	GUI	GUI
bloopers	BI00psrs	Blovpsrs	BI00psrs
don'ts and do's	Dorfts and D0'S	Dorfts and D0'S	Dcrxfts and D0'S
for	for	for	for
software	Softwar	Softwar	Softwar
developers and	Developers and	Developers and	Developers and
web designers	xgweb D esigners	wieb Designers	wieb Designers

5.5.11 Exemplo 11

(a) (b) (c)

(d)

Fig. 5.22: Imagem-cena contendo caracteres de dimensões similares em um plano de fundo texturizado. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.8: Reconhecimento do texto da imagem referente à Fig. 5.22

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
dry	Dry	Dry	Dry
riser	I'IS9I?	I?IS9I?	I'IS9I?

5.5.12 Exemplo 12

(a)

(b)

(c)

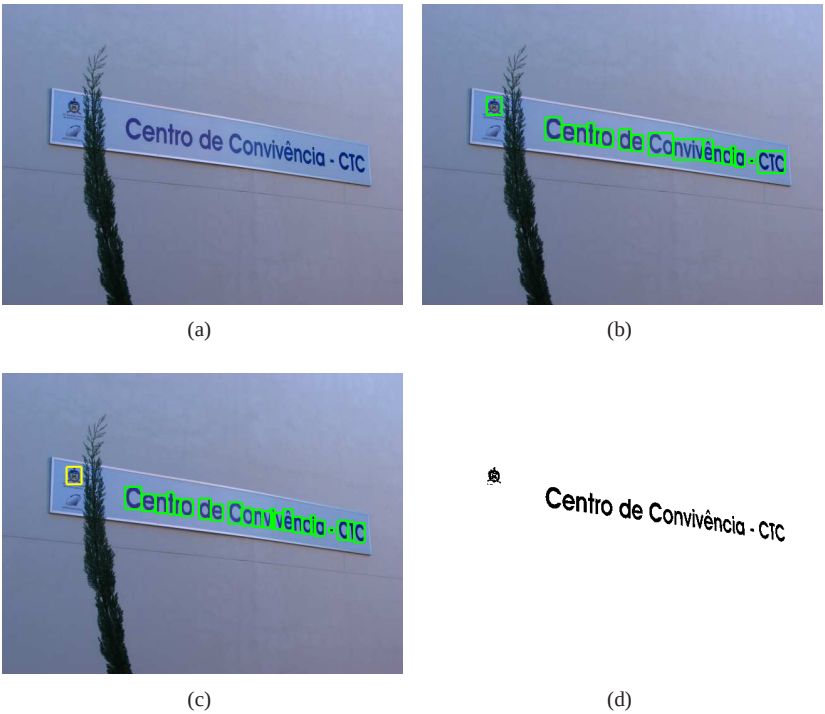
(d)

Fig. 5.23: Imagem-cena contendo caracteres apresentando contraste com o plano de fundo. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.9: Reconhecimento do texto da imagem referente à Fig. 5.23

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
while	WHILE	WHILE	WHILE
you	YOU	YOU	YOU
wait	WAIT	WAIT	WAIT
shoe	SHOE	SHOE	SHOE
repair	REPAIR	REPAIR	REPAIR
available	AVAILABLE	AVAILABLE	AVAILABLE

5.5.13 Exemplo 13

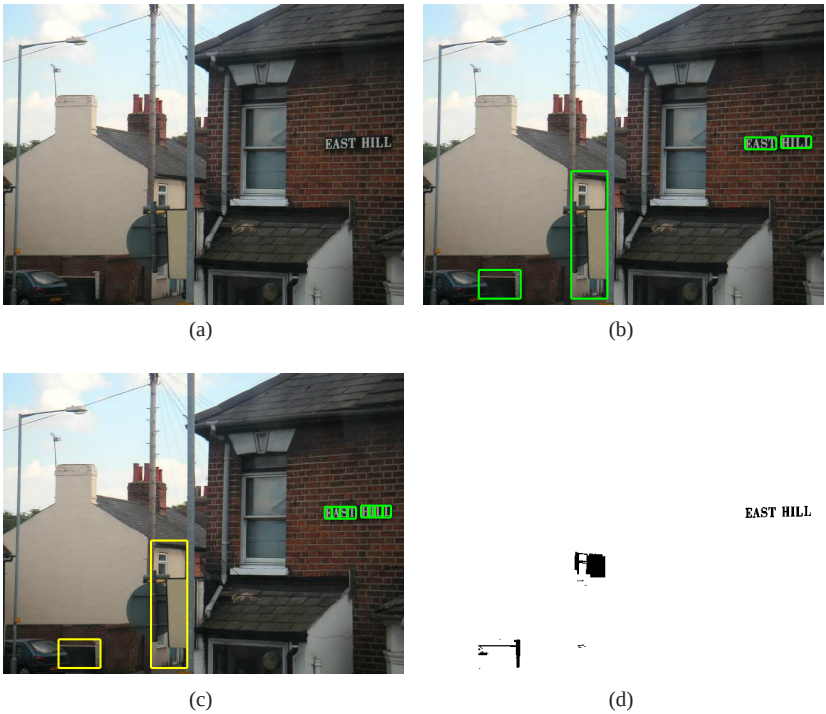


(a) (b)

(c) (d)

Fig. 5.24: Imagem-cena contendo caracteres com variação de orientação e distorção de perspectiva. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

5.5.14 Exemplo 14



(a) (b)

(c) (d)

Fig. 5.25: Imagem-cena contendo caracteres de dimensões reduzidas em um plano de fundo texturizado. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.10: Reconhecimento do texto da imagem referente à Fig. 5.25

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
east hill	EAST HILL	EAST HILL	EAST HILL

5.5.15 Exemplo 15

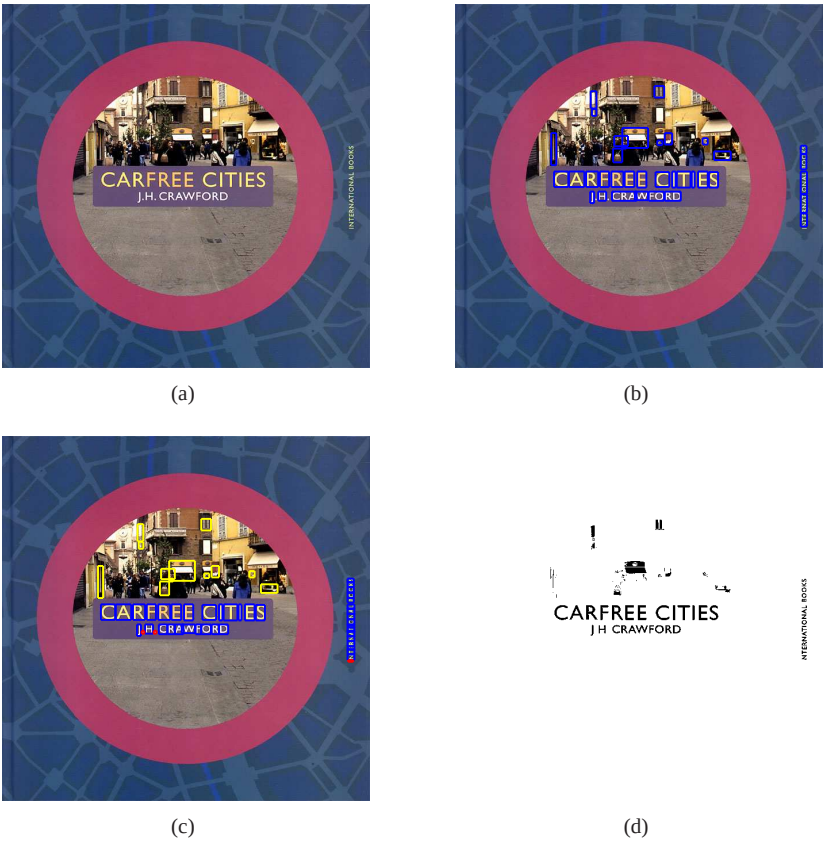
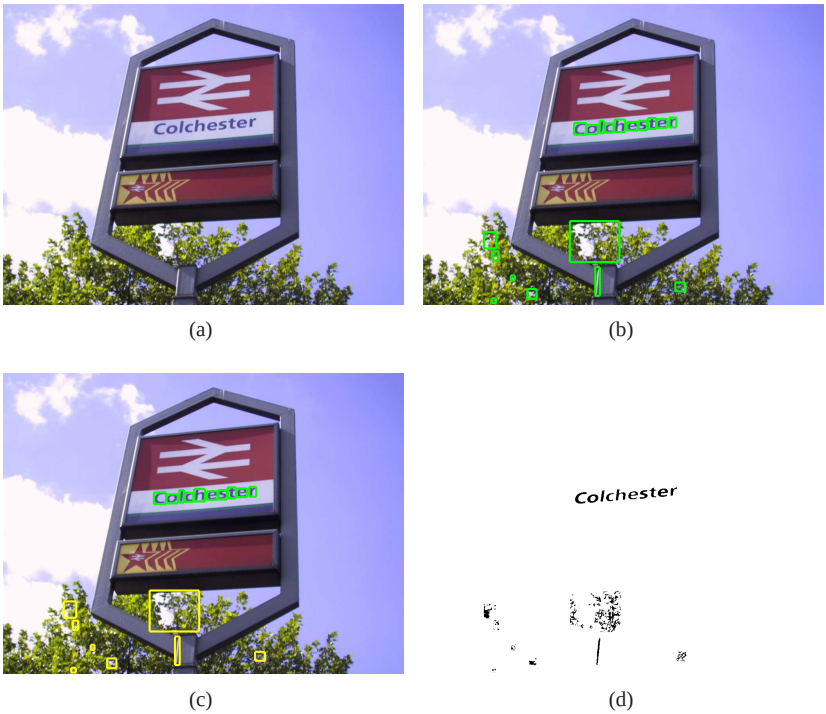


Fig. 5.26: Imagem-artificial contendo caracteres de cores e diferentes orientações em um plano de fundo complexo. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.11: Reconhecimento do texto da imagem referente à Fig. 5.26

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
carfree cities	CARFREE CITIES	CARFREE CITIES	CARFREE CITIES
j.h. crawford	] H CRAWFORD	j I-I CRAWFORD	] I-1 CRAWFORD

5.5.16 Exemplo 16



(a) (b)

(c) (d)

Fig. 5.27: Imagem-cena contendo caracteres de dimensões reduzidas. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.12: Reconhecimento do texto da imagem referente à Fig. 5.27

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
Colchester	Colchester	Colchester	Colchester

5.5.17 Exemplo 17

(a) (b) (c)

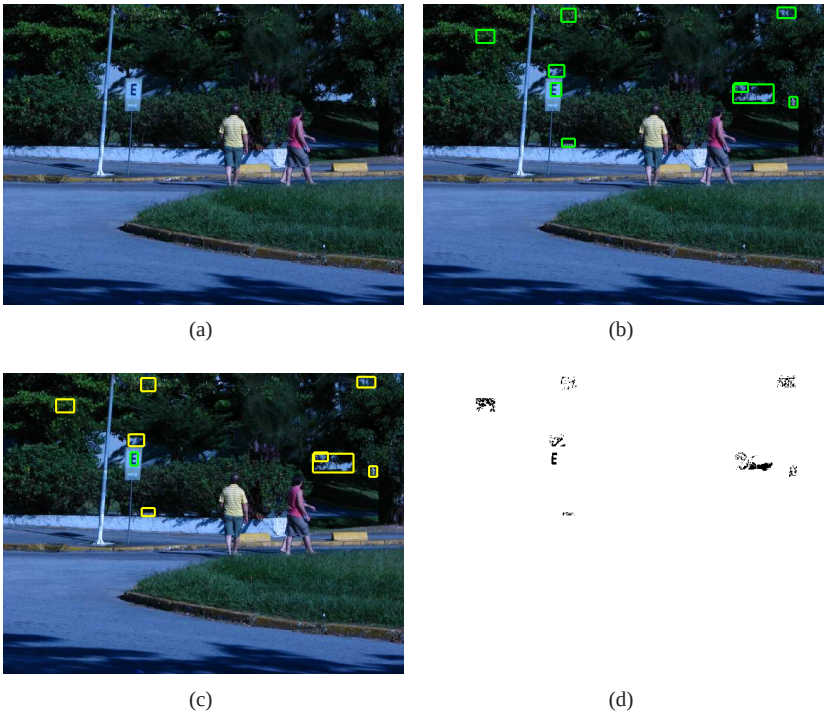
(d)

Fig. 5.28: Imagem-artificial contendo caracteres de contraste positivo e negativo. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.13: Reconhecimento do texto da imagem referente à Fig. 5.28

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
corruption:	CORRPTION :	CORRWTION :	CORR§P?|'|0N :
who	[HIE]	[HIE]	[HIE]
cares?	cnnns?	cnnns?	CARES?
brian lightowler	Brian Lightowler	Brian Lightowler	Brian Lightowler

5.5.18 Exemplo 18



(a) (b)

(c) (d)

Fig. 5.29: Imagem-cena contendo apenas um caractere isolado. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.14: Reconhecimento do texto da imagem referente à Fig. 5.29

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
E	E	E	E

5.5.19 Exemplo 19

Fig. 5.30: Imagem-artificial contendo caracteres com variação de cor. (a) Imagem original. (b) Resultado após as etapas de localização e verificação. (c) Desempenho do sistema sob método de avaliação mínimo BB. (d) Resultado da etapa de extração.

Tabela 5.15: Reconhecimento do texto da imagem referente à Fig. 5.30

Tag Real	OCR (bilinear)	OCR (nearest)	OCR (bicubic)
internet access	Internet Access	lntsrnst Access	Internet Access
get connected.	Get Connected.	Get Connected.	Get Connected.

5.6 Considerações Finais

Este capítulo destinou-se a apresentar o pré-processamento proposto da imagem binária, obtida após a etapa de extração, visando promover a melhoria da taxa de reconhecimento pelo sistema de OCR. Além disso, o método de Jain e Yu é comparado ao proposto sob três formas de avaliação, sendo uma dessas prosposta neste trabalho.

O capítulo evidenciou a necessidade da clusterização e normalização dos caracteres para adaptar as palavras extraídas ao formato que os sistemas de OCR foram projetados. Visto que o sistema de localização e extração de caracteres não impõe qualquer restrição quanto às dimensões da imagem e dos caracteres, propôs-se um método de clusterização hierárquica para agrupar caracteres alinhados horizontalmente independente das dimensões da imagem e dos caracteres de entrada.

Avaliações foram realizadas para definir o desempenho do sistema proposto na localização das regiões textuais em imagens complexas. Duas implementações foram propostas, uma classifica as regiões textuais por meio do

classificador SVM e a outra mediante limiares obtidos diretamente dos histogramas de atributos (Capítulo 3). O desempenho das implementações foi comparado ao método de Jain e Yu sob 3 formas de avaliação, sendo uma proposta neste trabalho, denominada mínimo BB. A principal vantagem da avaliação é determinar, de maneira objetiva, os caracteres localizados, os não-localizados e os falsos positivos independente do tipo de saída do algoritmo de localização (caracteres, palavras ou frases). Além disso, o método de avaliação proposto não computa perda de desempenho quando não existe casamento perfeito entre os BBs localizados e os BBs de referência. O desempenho na determinação das regiões textuais das duas implementações propostas se mostraram superiores ao método de Jain e Yu, tanto em precisão quanto em taxa de reconhecimento. Contudo, notou-se que a implementação via histogramas, apesar de possuir uma taxa de reconhecimento de caracteres um pouco superior à da implementação utilizando o classificador SVM, apresenta uma precisão consideravelmente inferior. Isso demonstra que a etapa de verificação utilizando o classificador SVM aumenta a robustez na determinação das regiões textuais às custas de uma pequena redução na taxa de reconhecimento.

Capítulo 6

Comentários e Conclusões

Neste trabalho foi apresentado um sistema TIE para imagens complexas. Quando nenhuma restrição é imposta à imagem de entrada e aos caracteres nela embutidos, torna-se desafiador encontrar técnicas de localização que contemplem a variedade de composições existente entre caracteres e plano de fundo. Por essa razão, a técnica de localização desenvolvida neste trabalho busca características comuns à maioria das regiões textuais, cuja principal premissa é assumir que os pixels de borda (contorno) dos caracteres estão entre os pixels da imagem que possuem maior magnitude do gradiente. Tal proposta permite a localização individual dos caracteres e reduz a grande quantidade de heurísticas impostas pelos métodos baseados em CCs na determinação das áreas textuais, tais como limiares de alinhamento, espaçamento, orientação, similaridade de dimensões e cores. O método proposto, além de localizar caracteres isolados, é indiferente ao tipo de fonte, orientação, distorção de perspectiva, espaçamento, alinhamento, positividade (ou negatividade) do contraste, ascendência ou descendência e dimensões dos caracteres. Possui como vantagem adicional possuir tal capacidade sem a utilização de técnicas multiresolução, essas computacionalmente mais complexas. No entanto, o método falha em reconhecer caracteres sobrepostos a planos de fundo com bordas evidentes[1] ou com caracteres fora de foco.

Após a realização da etapa de localização, diversas regiões textuais e não-textuais (com características textuais) são identificadas e delimitadas, re-

[1] A magnitude do gradiente dos pixels da borda do plano de fundo é comparável a dos caracteres.

duzindo assim, a área de análise da imagem para as etapas posteriores. Isso permite que a etapa de verificação utilize técnicas estruturais e texturais, computacionalmente mais complexas, para selecionar, dentre as regiões localizadas, as regiões textuais. Dessa forma, obtém-se uma maior robustez, visto que o método de identificação textual utiliza técnicas baseadas em região (etapa de localização) e textura (etapa de verificação) seqüencialmente.

A etapa de verificação proposta baseou-se em encontrar atributos com capacidade classificatória entre regiões textuais e não-textuais. Tal capacidade foi determinada pela extração de valores de um dado atributo para 3 classes de imagem: *caractere*, *mais de um caractere* e *não-caractere*. Três histogramas, referentes a cada classe, são obtidos para cada atributo, permitindo assim identificar possíveis limiares de cada atributo capazes de diferenciar regiões textuais das não-textuais. Dentre os 16 atributos avaliados, 11 são estruturais e 5 texturais.

Para avaliar a melhor forma de identificar as regiões textuais dentre as localizadas, foram implementadas duas estratégias de verificação. Uma utilizando os limiares obtidos diretamente da sobreposição dos histogramas de cada atributo e outra treinando um classificador SVM com os atributos mais relevantes.

A estratégia via histogramas permite obter diretamente os limiares de cada atributo capazes de filtrar as regiões não-textuais localizadas. Além disso, auxilia o projetista na escolha direta dos melhores atributos para uma determinada aplicação. A estratégia utilizando o classificador SVM se utiliza de um seletor de atributos da abordagem *wrapper* para obter o melhor subconjunto de atributos para o treinamento. Uma vez determinado o subconjunto de atributos, buscam-se os melhores parâmetros para o treinamento do algoritmo SVM. Tal busca é realizada por meio da técnica *grid-search* associada à técnica *cross-validation*, visando produzir um modelo generalista.

O desempenho na determinação das regiões textuais das duas estratégias propostas mostrou-se superior ao método método de Jain e Yu, tanto em precisão quanto em taxa de reconhecimento. Todavia, notou-se que a implementação via histogramas, apesar de possuir uma taxa de reconhecimento superior à implementação utilizando o classificador SVM, apresenta uma precisão consideravelmente inferior. Notou-se que o classificador SVM introduz uma maior robustez na identificação das regiões textuais às custas de uma redução da taxa de reconhecimento. Tal avaliação de desempenho dos algoritmos foi realizada por meio de três métodos: área de intersecção, ICDAR e mínimo BB. O método mínimo BB proposto neste trabalho não computa perda de desempenho ao não casamento perfeito entre os BBs localizados e BBs referência, além de determinar, de maneira objetiva, os caracteres presentes na imagem que foram localizados corretamente independente do formato

de saída do algoritmo de localização (caracteres ou palavras).

Para a etapa de extração, a qual é de fundamental importância para o aumento da taxa de reconhecimento pelos sistemas de OCR, propôs-se uma técnica que associa características do sistema visual humano para a clusterização das cores da imagem. Associado à clusterização, existe um mecanismo de recorrência que evita o erro de clusterização nas bordas dos caracteres (região mais degradada durante o processo de compressão). Obtida a imagem binária, um método de normalização e clusterização hierárquica agrupa os caracteres com o objetivo de adequá-los ao reconhecimento por um sistema de OCR.

Sendo assim, o trabalho apresenta duas propostas de sistema TIE completo com resultados equiparáveis ao estado-da-arte no reconhecimento automático de caracteres em imagens complexas sem restrições; além disso, fornece um conjunto de ferramentas (banco de dados e interfaces) capazes de realizar uma rápida e eficaz comparação de desempenho entre algoritmos em cada uma das etapas de um sistema TIE.

Apêndice $\mathcal{A}$

Localização

A.1 Componentes Conectados - CCs

Componentes conectados são grupos de pixels que recebem um rótulo baseado na similaridade de intensidade e na conectividade entre eles. Todos os pixels em um CC compartilham valores de intensidade similares e estão de alguma forma conectados uns aos outros.

A.1.1 Identificação dos Componentes Conectados (CCs)

A identificação dos componentes conectados é gerada rotulando-se cada pixel de um CC com um mesmo valor. Os rótulos dos CCs são feitos pixel a pixel de maneira a identificar regiões de pixels conectados, ou seja, regiões de pixels adjacentes que compartilham o mesmo conjunto de valores de intensidade V. Em uma imagem binária o conjunto V possui um único elemento $V = \{1\}$ (Tabela A.1), contudo em uma imagem em níveis de cinza o conjunto V pode tomar uma faixa de valores (por exemplo, $V = \{20, 21, 22,..., 30\}$).

Os CCs da imagem apresentada na Tabela A.1 são apresentados por meio da Tabela A.2, em que cada CC está diferenciado por uma cor.

Tabela A.1: Imagem binária e sua correspondente representação matricial

Imagem Binária	Matriz Correspondente
	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{bmatrix}$

Tabela A.2: Imagem contendo os CCs identificados (destacados em cores) e a sua correspondente representação matricial

Imagem (CCs Rotulados)	Matriz Correspondente
	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 2 & 2 & 2 & 0 & 3 & 3 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 & 0 & 0 & 4 & 4 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 4 & 4 & 0 & 0 \end{bmatrix}$

Apêndice *B*

Verificação

B.1 Interface para Criação de Banco de Dados

A interface para a criação de base de dados tem como objetivo facilitar o cadastro manual de regiões de imagens em 3 classes: *caractere*, *mais-de-um-caractere* e *não-caractere*. Para obter um conjunto de regiões-exemplo relevantes capazes de determinar os atributos com poder classificatório e facilitar o treinamento de classificadores entre regiões textuais e não textuais, a interface realiza os seguintes passos:

- Após o carregamento da imagem, existe um painel em que o usuário pode demarcar as áreas da imagem que não possuem caracteres. Caso o localizador obtenha como saída algum BB contido em tais demarcações, as regiões delimitadas pelos BBs são automaticamente extraídas e cadastradas como *não-caractere*.

- A interface apresenta ao usuário, seqüencialmente, cada BB obtido pelo localizador que não está contido nas regiões demarcadas no passo anterior. Para cada BB apresentado, o usuário pode classificá-lo como *caractere*, *mais de um caractere* ou *não-caractere*. Caso a região delimitada seja textual, o usuário é obrigado a cadastrar no campo *'tag'* o caractere (ou caracteres) contidos no BB. Além disso, o usuário pode atribuir diversas informações adicionais a cada BB, tais como tipo de orientação, qualidade do caractere (bom ou ruim), tipo de não-caractere [parecido (P) ou não-parecido (NP) com caractere]. A interface automa-

ticamente cadastra a posição de cada região na imagem. Após o término do cadastramento das regiões obtidas pelo localizador, o usuário pode demarcar possíveis regiões textuais não reconhecidas seguindo um procedimento similar ao descrito anteriormente.

Gera-se, desta forma, um banco de regiões de imagens contendo informações detalhadas de 3 classes. Tais informações são utilizadas pela interface de avaliação de atributos, treinamento do classificador SVM e pela interface de avaliação de sistemas TIE completos. A Fig. B.1 ilustra as opções presentes na interface de criação do banco de dados.

B.1.1 Interface de Criação de Banco de Dados

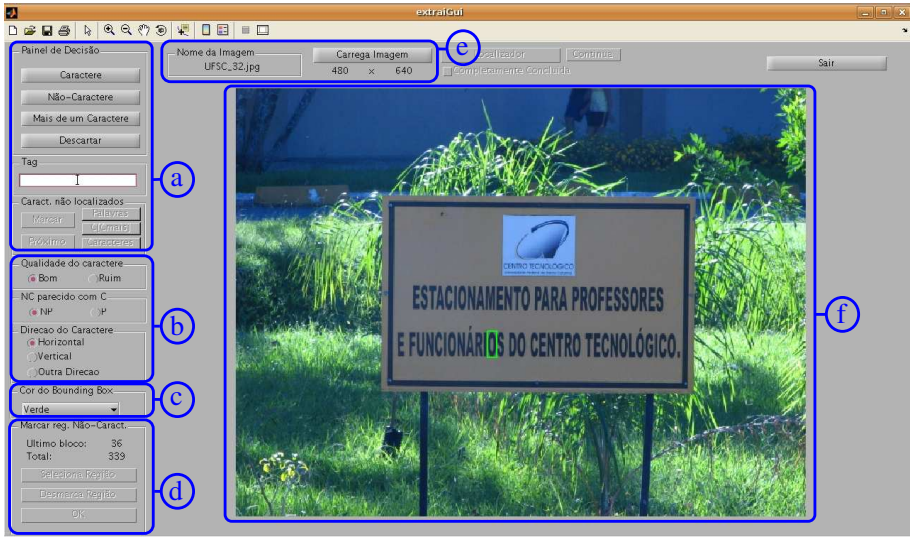


Fig. B.1: Interface para criação de banco de dados. (a) Painel de seleção da classe da região delimitada pelo BB localizado. Caso a região seja textual, existe um campo para cadastramento da palavra ou caractere contido no BB. (b) Painel que atribui rótulos de qualidade (subjetiva) às regiões delimitadas por BBs, tais como, bom, ruim, etc. (c) *Pop-up* que muda a cor da caixas limítrofes dos BBs. (d) Painel em que se pode demarcar as regiões que não possuem caracteres para cadastramento automático das regiões não-textuais. (e) Botão de seleção da imagem que se deseja realizar o cadastramento. (f) Região de visualização da imagem.

B.2 Interface de Avaliação de Atributos

Esta interface permite avaliar, por meio de histogramas, a eficiência de um determinado atributo na classificação de uma região qualquer da imagem como textual ou não-textual (Fig. B.2). O banco de dados possui imagens cadastradas em 3 classes: *caractere*, *não-caractere* e *mais de um caractere*. Uma vez escolhido o atributo a ser avaliado, o usuário pode filtrar as imagens do banco de dados de acordo com a aplicação, tais como, dimensões das imagens do banco, qualidade das imagens (bom ou ruim), imagens contendo apenas um caractere (ou mais de um caractere), dentre outros. Tal filtragem permite ao usuário avaliar o atributo para uma aplicação específica (por exemplo, caso o usuário saiba que determinada aplicação possui apenas caracteres de dimensões maiores do que 20 pixels de altura, pode-se avaliar tal atributo para regiões superiores a essa). Após aplicar as restrições de filtragem, um histograma de cada classe é desenhado em um mesmo gráfico para um dado atributo, em que o usuário pode incluir linhas de avaliação que separam o histograma em regiões (região sombreada Fig. B.2). A interface indica a porcentagem de ocorrência de cada classe sob a região sombreada delimitada pela linha de avaliação. As linhas de avaliação podem ser facilmente deslocadas, permitindo a seleção do melhor limiar do atributo para a classificação das regiões textuais e não-textuais. A interface permite acesso a um banco de dados de exemplos de variadas fontes e dimensões possuindo 3598 amostras da classe *caractere*, 1227 da classe *mais de um caractere* e 32740 exemplos da classe *não-caractere*, além de possuir 16 atributos implementados.

B.2.1 Interface de Avaliação de Atributos

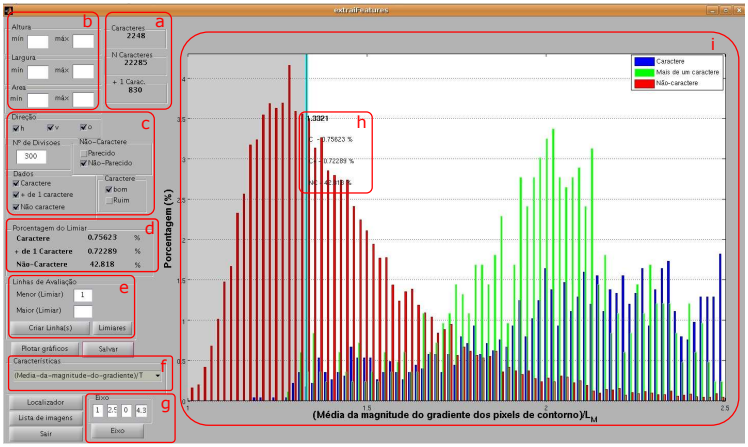


Fig. B.2: Interface de avaliação de atributos. (a) Indica o número de regiões de cada classe que foram avaliadas. (b) Campos para a filtragem do banco de dados de imagens de acordo com as dimensões desejadas. (c) Campos para a seleção de características dos caracteres e não-caracteres, tais como qualidade do caractere, orientação, etc; além dos dados e resolução que são apresentados no histograma. (d) Porcentagem da classe *caractere*, *não-caractere* e *mais de um caractere* presentes sob a região sombreada delimitada pela linha de avaliação. (e) Campos para a criação da linha de avaliação identificada por uma linha azul que delimita a região sombreada no histograma. (f) Seleção do atributo que deseja-se avaliar. (g) Ajuste dos limites da abscissa e ordenada para facilitar a avaliação da eficiência do atributo. (h) Limiares da região sombreada e porcentagem de cada classe sob tal região. (i) Área de apresentação do histograma do atributo.

Apêndice C

Extração

C.1 *Imagens-teste* da Etapa de Extração

(a) (b) (c)

(d) (e)

(f)

Fig. C.1: Conjunto de *imagens-teste* utilizadas na etapa de extração de textos para comparação de desempenho dos algoritmos. As imagens de (a) a (f) correspondem às imagens de 1 a 8, respectivamente.

(a)

(b)

(c)

(d)

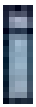
(e)

(f)

(g)

(h)

(i)



(j)

Fig. C.2: Conjunto de *imagens-teste* utilizadas na etapa de extração de textos para comparação de desempenho dos algoritmos. As imagens de (a) a (j) correspondem às imagens de 7 a 18, respectivamente.

(a)

(b) (c) (d) (e)

(f) (g)

(h)

(i)

Fig. C.3: Conjunto de *imagens-teste* utilizadas na etapa de extração de textos para comparação de desempenho dos algoritmos. As imagens de (a) a (i) correspondem às imagens de 17 a 25, respectivamente.

C.2 Interface de Extração de Caracteres

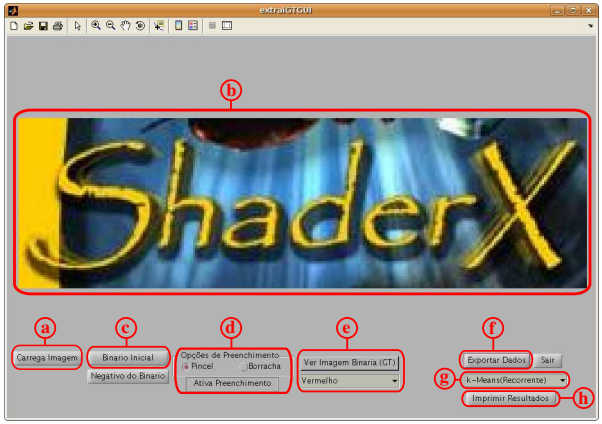
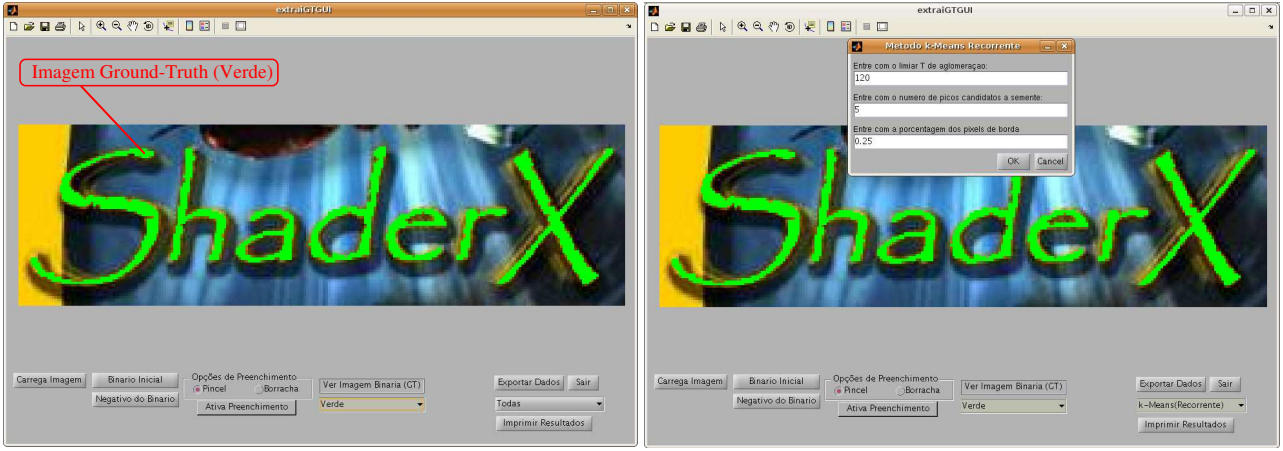


Fig. C.4: Interface de extração de caracteres. (a) Botão que permite carregar uma imagem. (b) Região da interface onde a imagem é apresentada. (c) Botão que permite a criação de uma imagem *ground-truth* inicial evitando que o processo seja completamente manual. (d) Ferramentas (pincel e borracha) que permitem marcar os pixels que representam a região do caractere para a criação da imagem *ground-truth*. (e) Botão que permite visualizar a imagem *ground-truth* em destaque com uma cor escolhida pelo usuário. (f) Botão que calcula o desempenho de um algoritmo selecionado. (g) *Pop-up* para escolha do algoritmo que se deseja avaliar o desempenho da binarização. (h) Botão que imprime os resultados na tela.



(a) (b)

Fig. C.5: Interface de extração de dados. (a) Imagem *ground-truth*, em que todos os pixels dos caracteres são destacados em verde. (b) Avaliação do algoritmo *K-means recorrente*. Um menu aparece em destaque para a configuração das constantes que influenciam no desempenho do algoritmo.

Apêndice D

Interface de Avaliação de Desempenho

Os resultados de avaliação final apresentados no Capítulo 5 são obtidos utilizando a interface de avaliação de sistemas TIE. Essa interface foi criada com o objetivo de integrar os algoritmos de localização, verificação, extração, OCR e avaliação para permitir a comparação de diferentes sistemas TIE sobre um mesmo banco de dados.

A interface, além de permitir a comparação de desempenho de um conjunto de algoritmos sob 3 diferentes métodos de avaliação (área de intersecção, ICDAR, mínimo BB) a um mesmo banco de imagens, ainda permite acoplar os algoritmos de localização-verificação com diversos algoritmos extratores. Tal flexibilidade permite selecionar a melhor combinação possível dos algoritmos.

Um método de clusterização proposto (Capítulo 5) associado ao sistema de OCR Tesseract [34] incluído também na interface, permite ao usuário clusterizar e reconhecer as palavras presentes na imagem através de 3 métodos de normalização dos caracteres (*nearest neighbor*, bilinear e bicúbica).

A interface é, então, um ambiente para uma rápida avaliação de desempenho de sistemas TIE futuros, possibilitando a identificação das vantagens e desvantagens de cada etapa da implementação.

D.1 Interface de Avaliação de Sistemas TIE

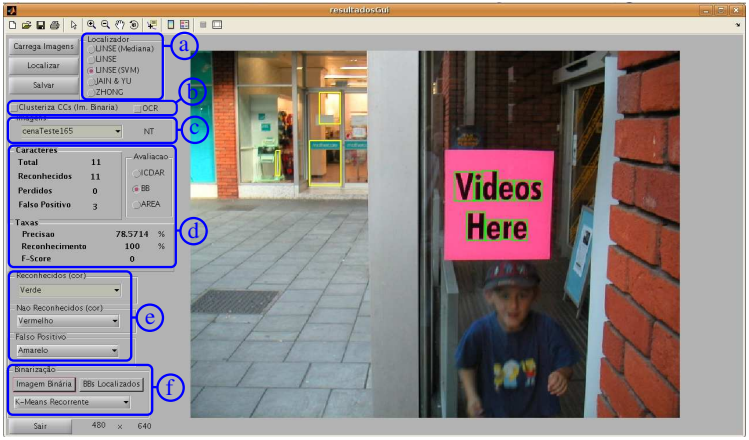


Fig. D.1: Interface de avaliação de sistemas TIE. (a) Quadro de seleção do sistema localização-verificação. (b) Botões que permitem selecionar a clusterização dos BBs localizados e reconhece as palavras por meio do sistema de OCR Tesseract. (c) *Pop-up* para seleção da imagem que se deseja realizar a localização e extração dos caracteres. Existe um símbolo ao lado do *pop-up* que indica se a imagem participou do conjunto de treinamento (‘T’) ou não (‘NT’). (d) Quadro de escolha do método de avaliação e resultados correspondentes. (e) Seleção das cores dos BBs localizados, não-localizados e falsos-positivos. (f) *Pop-up* de seleção do algoritmo extrator, bem como os botões para apresentar o resultado obtido por cada algoritmo.

Referências Bibliográficas

[1] Search engine watch, 2006.

[2] ANAGNOSTOPOULOS, C.-N.; ANAGNOSTOPOULOS, I.; PSOROULAS, I.; LOUMOS, V. & KAYAFAS, E. License plate recognition from still images and video sequences: A survey. IEEE Transactions on Intelligent Transportation Systems 9, 3 (Sept. 2008), 377–391.

[3] ANTANI, S.; CRANDALL, D. & KASTURI, R. Robust extraction of text in video. In ICPR '00: Proceedings of the International Conference on Pattern Recognition, Washington, DC, USA, 2000, IEEE Computer Society, p. 1831.

[4] ANTONACOPOULOS, A.; KARATZAS, D. & LOPEZ, J. O. Accessing textual information embedded in internet images. In SPIE - Internet Imaging II, San Jose, USA, 2001, p. 198–205.

[5] BECCARI, C. Typesetting mathematics for science and technology according to ISO 31/XI. TUGboat 18, 1 (march 1997), 39–48.

[6] BERNSEN, J. Dynamic thresholding of grey-level images. In Proc. International Conference on Pattern Recognition, Paris, França, 1986, p. 1251–1255.

[7] BLANCHET, G. & CHARBIT, M. Digital Signal and Image Processing Using MATLAB (Digital Signal and Image Processing series). ISTE, 2006.

[8] CAI, M.; SONG, J. & LYU, M. A new approach for video text detection. International Conference on Image Processing 1 (2002), I–117–I–120.

[9] CANNY, J. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 6 (1986), 679–698.

[10] CHANG, S.; CHEN, L.; CHUNG, Y. & CHEN, S. Automatic license plate recognition. IEEE Transactions on Intelligent Transportation Systems 5, 1 (March 2004), 42–53.

[11] CHEN, D. Text Detection and Recognition in Images and Video Sequences. Tese de Doutorado, École Polytechnique Fêdêrale de Lausanne, 2003.

[12] CHEN, D.; BOURLARD, H. & THIRAN, J.-P. Text identification in complex background using svm. IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2 (2001), 621.

[13] CHEN, X. & YUILLE, A. Detecting and reading text in natural scenes. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2 (June-2 July 2004), II–366–II–373 Vol.2.

[14] CHEW H-G, CRISP D. J., B. R. E. Target detection in radar imagery using support vector machines with training size biasing. In Proceedings of the Sixth International Conference on Control Automation Robotics and Vision, 2000.

[15] CLARK, P. & MIRMEHDI, M. Combining statistical measures to find image text regions. International Conference on Pattern Recognition 01 (2000), 1450.

[16] CRISTIANINI, N. & SHAWE-TAYLOR, J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, March 2000.

[17] CUI, Y. T. & HUANG, Q. Character extraction of license plates from video. In CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97), Washington, DC, USA, 1997, IEEE Computer Society, p. 502.

[18] DASH, M. & LIU, H. Feature selection for classification. Intelligent Data Analysis 1 (1997), 131–156.

[19] DIMITRIS MANOLAKIS, VINAY K. INGLE, S. M. K. Statistical and Adaptive Signal Processing. Artech House Publishers, 2005.

[20] DUE; JAIN, A. K. & TAXT, T. Feature extraction methods for character recognition-a survey. Pattern Recognition 29, 4 (April 1996), 641–662.

[21] EIDENBERGER, H. A new perspective on visual information retrieval. In SPIE Electronic Imaging Symposium, 1997, vol. 5304.

[22] GLLAVATA, J.; EWERTH, R. & FREISLEBEN, B. Text detection in images based on unsupervised classification of high-frequency wavelet coefficients. In ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1, Washington, DC, USA, 2004, IEEE Computer Society, p. 425–428.

[23] GONZALEZ, R. & WOODS, R. Digital Image Processing, Second Edition. Prentice Hall, 2002.

[24] GUYON, I. & ELISSEEFF, A. An introduction to variable and feature selection. J. Mach. Learn. Res. 3 (2003), 1157–1182.

[25] HE, J.; DO, Q. D. M.; DOWNTON, A. C. & KIM, J. H. A comparison of binarization methods for historical archive documents. In ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition, Washington, DC, USA, 2005, IEEE Computer Society, p. 538–542.

[26] JAIN, A. & BHATTACHARJEE, S. Address block location on envelopes using gabor filters. Pattern Recognition 25, 12 (December 1992), 1459–1477.

[27] JAIN, A. & YU, B. Automatic text location in images and video frames. Pattern Recognition 31, 12 (Aug 1998), 2055–2076.

[28] JAIN, A. & ZHONG, Y. Page segmentation using texture discrimination masks. Image Processing, International Conference on 3 (1995), 308–311.

[29] JAIN, A. K. & BHATTACHARJEE, S. Text segmentation using gabor filters for automatic document processing. Mach. Vision Appl. 5, 3 (1992), 169–184.

[30] JAIN, A. K. & BHATTACHARJEE, S. K. Address block location on envelopes using gabor filters. Pattern Recognition 25, 12 (1992), 1459 – 1477.

[31] JAIN, A. K. & DUBES, R. C. Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[32] JUNG, K.; KIM, K. I. & JAIN, A. K. Text information extraction in images and video: a survey. Pattern Recognition 37, 5 (2004), 977–997.

[33] KARATZAS, D. & ANTONACOPOULOS, A. Colour text segmentation in web images based on human perception. Image Vision Comput. 25, 5 (2007), 564–577.

[34] KAY, A. Tesseract: an open-source optical character recognition engine. Linux J. 2007, 159 (2007), 2.

[35] KITTLER, J. & ILLINGWORTH, J. On threshold selection using clustering criteria. SMC 15, 5 (September 1985), 52–655.

[36] KOHAVI, R. & JOHN, G. H. Wrappers for feature subset selection. Artificial Intelligence 97, 1-2 (1997), 273–324.

[37] LEE, H. & PARK, R.-H. Comments on an optimal multiple threshold scheme for image segmentation. IEEE Transactions on Systems, Man and Cybernetics 20, 3 (June 1990), 741–742.

[38] LEE, H.-C. & COK, D. Detecting boundaries in a vector field. IEEE Transactions on Signal Processing 39, 5 (May 1991), 1181–1194.

[39] LI, H.; DOERMANN, D. & KIA, O. Automatic text detection and tracking in digital video. IEEE Transactions on Image Processing 9, 1 (Jan 2000), 147–156.

[40] LI, J. & GRAY, R. Text and picture segmentation by the distribution analysis of wavelet coefficients. p. 790–794 vol.3.

[41] LIENHART, R. & WERNICKE, A. Localizing and segmenting text in images and videos. IEEE Transactions on Circuits and Systems for Video Technology 12, 4 (Apr 2002), 256–268.

[42] LIM, Y. W. & LEE, S. U. On the color image segmentation algorithm based on the thresholding and the fuzzy c-means techniques. Pattern Recognition 23, 9 (1990), 935–952.

[43] LIU, Y.; GOTO, S. & IKENAGA, T. A contour-based robust algorithm for text detection in color images. IEICE - Trans. Inf. Syst. E89-D, 3 (2006), 1221–1230.

[44] LOO, P. K. & TAN, C. L. Adaptive region growing color segmentation for text using irregular pyramid. In Document Analysis Systems, 2004, p. 264–275.

[45] LUCAS, S. M. Text locating competition results. International Conference on Document Analysis and Recognition 0 (2005), 80–85.

[46] LUCCHESE, L. & MITRA, S. Color image segmentation: a state-of-the-art survey. In Proc. of the Indian National Science Academy (INSA-A), New Delhi, India, 2001, p. 207–221.

[47] MESSELODI, S. & MODENA, C. M. Automatic identification and skew estimation of text lines in real scene images. Pattern Recognition 32, 5 (1999), 791–810.

[48] NIBLACK, W. An Introduction to Digital Image Processing. Prentice Hall, 1986.

[49] OTSU, N. A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics 9, 1 (January 1979), 62–66.

[50] PETROU, M. & BOSDOGIANNI, P. Image Processing: The Fundamentals. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[51] PRATT, W. K. Digital Image Processing: PIKS Inside, 3rd Edition. Wiley-Interscience, July 2001.

[52] QIAO, Y.-L.; LI, M.; LU, Z.-M. & SUN, S.-H. Gabor filter based text extraction from digital document images. In IIH-MSP '06: Proceedings of the 2006 International Conference on Intelligent Information Hiding and Multimedia, Washington, DC, USA, 2006, IEEE Computer Society, p. 297–300.

[53] RUMELHART, D. E.; HINTON, G. E. & WILLIAMS, R. J. Learning internal representations by error propagation. Parallel distributed processing: explorations in the microstructure of cognition 1 (1986), 318–362.

[54] RUMELHART, D. E.; WIDROW, B. & LEHR, M. A. The basic ideas in neural networks. Commun. ACM 37, 3 (1994), 87–92.

[55] SAOI, T.; GOTO, H. & KOBAYASHI, H. Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features. In ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition, Washington, DC, USA, 2005, IEEE Computer Society, p. 690–694.

[56] SATO, T.; KANADE, T.; HUGHES, E. K.; SMITH, M. A. & SATOH, S. Video ocr: indexing digital news libraries by recognition of superimposed captions. Multimedia Systems 7, 5 (April 1999), 385–395.

[57] SAUVOLA, J. & INEN, M. P. Adaptive document image binarization. Pattern Recognition 33 (2000), 225–236.

[58] SERRA, J. Image Analysis and Mathematical Morphology. Academic Press, London, 1982.

[59] SIN, B.-K.; KIM, S.-K. & CHO, B.-J. Locating characters in scene images using frequency features. In ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition, Washington, DC, USA, 2002, IEEE Computer Society, p. 30489.

[60] TAHIM, A. P. N. & SEARA, R. Método k-means recorrente para extração de texto em imagens web. In Anais do XVII Congresso Brasileiro de Automática, 2008, SBA.

[61] THE, P.; RICE, S. V.; RICE, S. V.; JENKINS, F. R.; JENKINS, F. R.; NARTKER, T. A. & NARTKER, T. A. The fifth annual test of ocr accuracy. Relatório técnico., 1996.

[62] TRIER, O. D. & JAIN, A. K. Goal-directed evaluation of binarization methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 12 (1995), 1191–1201.

[63] TRIER, O. D. & TAXT, T. Evaluation of binarization methods for document images. IEEE Transactions on Pattern Analysis and Machine Intelligence 17, 3 (1995), 312–315.

[64] VAPNIK, V. N. The nature of statistical learning theory. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[65] WESTON, J.; MUKHERJEE, S.; CHAPELLE, O.; PONTIL, M. & VAPNIK, V. Feature selection for svms. In Advances in Neural Information Processing Systems 13, 2000, vol. 13, p. 668–674.

[66] WU, V.; MANMATHA, R. & RISEMAN, E. Textfinder: An automatic system to detect and recognize text in images. IEEE Transactions on Pattern Analysis and Machine Intelligence 21, 11 (November 1999), 1224–1229.

[67] WU, V.; MANMATHA, R. & RISEMAN, E. M. Finding text in images. In DL '97: Proceedings of the second ACM international conference on Digital libraries, New York, NY, USA, 1997, ACM, p. 3–12.

[68] YU, B.; JAIN, A. K. & MOHIUDDIN, M. Address block location on complex mail pieces. Document Analysis and Recognition, International Conference on 0 (1997), 897.

[69] ZHANG, Y. J. A survey on evaluation methods for image segmentation. Pattern Recognition 29, 8 (August 1996), 1335–1346.

[70] ZHONG, Y.; KARU, K. & JAIN, A. K. Locating text in complex color images. In ICDAR '95: Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1), Washington, DC, USA, 1995, IEEE Computer Society, p. 146.