

**RODOLFO VALIENTE ROMERO**

**PROCESSO AUTOMÁTICO DE RECONHECIMENTO DE TEXTO EM  
IMAGENS DE DOCUMENTOS DE IDENTIFICAÇÃO GENÉRICOS**

São Paulo  
2018

**RODOLFO VALIENTE ROMERO**

**PROCESSO AUTOMÁTICO DE RECONHECIMENTO DE TEXTO EM  
IMAGENS DE DOCUMENTOS DE IDENTIFICAÇÃO GENÉRICOS**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

São Paulo  
2018

**RODOLFO VALIENTE ROMERO**

**PROCESSO AUTOMÁTICO DE RECONHECIMENTO DE TEXTO EM  
IMAGENS DE DOCUMENTOS DE IDENTIFICAÇÃO GENÉRICOS**

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Ciências.

Área de Concentração:  
Engenharia de Computação

Orientadora:  
Prof<sup>a</sup>. Dra. Graça Bressan

São Paulo  
2018

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, \_\_\_\_\_ de \_\_\_\_\_ de \_\_\_\_\_

Assinatura do autor: \_\_\_\_\_

Assinatura do orientador: \_\_\_\_\_

#### Catalogação-na-publicação

Romero, Rodolfo Valiente

Processo automático de reconhecimento de texto em imagens de documentos de identificação genéricos / R. V. Romero -- versão corr. -- São Paulo, 2018.

168 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Reconhecimento de texto 2. Documentos de identificação I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II.t.

“A persistência é o caminho do êxito.” (Charles Chaplin).

## **Dedicatoria.**

---

*A quien debo todo cuanto soy y seré en la vida... quienes son mi orgullo y razón de ser... mis padres.*

*A mi hermano porque sé que me quiere con la vida.*

*A mi abuela que siempre ha sacrificado todo por mí, por ser tan cariñosa y extremadamente buena.*

*A mi familia, en la cual están incluidos mis grandes amigos...*

## **AGRADECIMENTOS**

Agradeço especialmente à minha orientadora Graça, por me aceitar como seu aluno, por a paciência, por sua ajuda durante todo o meu mestrado, pela disponibilidade frequente, pelas respostas rápidas e pelo caminho bem orientado abrindo novas oportunidades e me orientando sempre. Estou infinitamente agradecido.

Ao Brasil, e especialmente à Universidade de São Paulo que me abriram as portas e deram a oportunidade de aumentar meus conhecimentos e desenvolver este projeto de pesquisa.

Aos meus colegas do LARC e da USP, aos meus professores que ao longo da minha vida me ensinaram e formaram a base dos meus conhecimentos, a todos os colegas e ex-colegas com quem sempre contatei pelos ensinamentos e pelas experiências, a todos o meu muito obrigado.

Ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Scopus Soluções em TI, e também Fundação de Apoio à Universidade de São Paulo (FUSP) pelo apoio financeiro.

Aos professores e membros da minha banca de qualificação, Prof. Dr. Alain e Prof. Dr. Flavio.

Finalmente agradecer ao leitor, muito obrigado pelo tempo investido em ler esta dissertação, espero seja útil para você.

## **AGRADECIMIENTOS**

A mis padres, que con tanto amor han estado pendientes de mí y me han ayudado a hacer realidad mi sueño. A mi Hermano porque forma parte fundamental de mi existencia. Los quiero más que a nada en este mundo...

A José Carlos el doctor y José Carlos el ingeniero, mis hermanos de la vida.

A toda mi familia, por sus consejos, su confianza y por todo el apoyo. En especial a Baby mi madre aquí en Brasil.

A Carla, a Eduardo, a Celio, a Carlos y a Rolando, por su amistad incondicional.

A Marcelo, una persona genial, reímos mucho, porque este trabajo también es de él y resultado de este proyecto surgió una gran amistad.

A Sandra, Luisa, Heydi, Claudia, Nelson e José Enrique a todos por hacer feliz mi vida en Brasil. A todos los amigos que han estado a mi lado que me cuidan y los cuido a distancia, e a todas las personas especiales que pasaron por mi vida. Gracias a los cuales me torne la persona que soy hoy.

A todos y todas las personas que de una forma u otra han tenido la amabilidad de brindarme su ayuda e que contribuyeron directa o indirectamente a la conclusión de este trabajo. A todos gracias infinitas....

## RESUMO

Existe uma busca crescente por métodos de extração de texto em imagens de documentos. O uso de imagens digitais tem se tornado cada vez mais frequente em diversas áreas. O mundo moderno está cheio de texto, que os seres humanos usam para identificar objetos, navegar e tomar decisões. Embora o problema do reconhecimento de texto tenha sido amplamente estudado dentro de determinados domínios, detectar e ler texto em documentos de identificação, continua sendo um desafio aberto. Apresenta-se uma arquitetura que integra os diferentes algoritmos de localização, extração e reconhecimento aplicados à extração de texto em documentos de identificação genéricos.

O método de localização proposto usa o algoritmo MSER junto com uma melhoria do contraste e a informação das bordas dos objetos da imagem, para localizar os possíveis caracteres. A etapa de seleção desenvolveu-se mediante a busca de heurísticas, capazes de classificar as regiões localizadas como textuais e não-textuais. Na etapa de reconhecimento é proposto um método iterativo para melhorar o desempenho do OCR.

O processo foi avaliado usando as métricas precisão e revocação e foi realizada uma prova de conceito do sistema em um ambiente real. A abordagem proposta é robusta na detecção de textos oriundos de imagens complexas com diferentes orientações, dimensões e cores. O sistema de reconhecimento de texto proposto apresenta resultados competitivos, tanto em precisão e taxa de reconhecimento, quando comparados com outros sistemas. Mostrando excelente desempenho e viabilidade de sua implementação em sistemas reais.

**Palavras-chave:** Documentos de identificação. Seleção e reconhecimento de texto. MSER. OCR.

## **ABSTRACT**

The use of digital images has become more and more frequent in several areas. The modern world is full of text, which humans use to identify objects, navigate and make decisions. Although the problem of text recognition has been extensively studied within certain domains, detecting and recognizing text in identification documents remains an open challenge. We present an architecture that integrates the different localization, extraction and recognition algorithms applied to extracting text in generic identification documents.

The proposed localization method uses the MSER algorithm together to contrast enhance and edge detection to find the possible characters. The selection stage was developed through the search for heuristics, capable of classifying the located regions in textual and non-textual. In the recognition step, an iterative method is proposed to improve OCR performance.

The process was evaluated using the metrics precision and recall and a proof of concept of the system was performed in a real environment. The proposed approach is robust in detecting texts from complex images with different orientations, dimensions and colors. The text recognition system presents competitive results, both in accuracy and recognition rate, when compared with other systems in the current technical literature. Showing excellent performance and feasibility of its implementation in real systems.

**Keywords:** Identification documents. Text recognition. MSER. OCR.

## LISTA DE ILUSTRAÇÕES

Figura 1-1: Exemplos de imagens-documento: (a) trecho de um artigo, (b) notícia de jornal. Fonte: Autor.....	27
Figura 1-2: Exemplo de imagem-artificial (imagem colorida com caracteres). Fonte: Autor.....	28
Figura 1-3: Exemplos de imagens-cena: (a) imagem contendo variações de iluminação sobre os caracteres devido às distorções geométricas, (b) imagem contendo caracteres com distorção de perspectiva e iluminação não-uniforme. Fonte: Autor.....	28
Figura 1-4: (a) imagem-artificial, (b) imagem-cena, de documentos de identificação. Fonte: Autor.....	29
Figura 1-5: Transformação realizada por um sistema de extração de texto com o objetivo de separar os caracteres do plano de fundo complexo. Fonte: Autor.....	30
Figura 1-6: Arquitetura de um sistema de extração da informação textual. Fonte: Autor.....	31
Figura 1-7: Sequência de resultados das 3 primeiras etapas de um Sistemas de extração da informação textual: (a) imagem original, (b) resultado da etapa de localização, (c) resultado após a etapa de seleção, (d) imagem binária após a etapa de extração. Fonte: Autor.....	32
Figura 1-8: 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). Fonte: Autor.....	34
Figura 2-1: Imagem original e demonstração de parte dos pixels. Fonte: Autor.....	42
Figura 2-2 Ilustração da conversão de uma imagem colorida para uma em escala de cinza (a) imagem colorida, (b) imagem em escala de cinza. Fonte: Autor.....	43
Figura 2-3: Histogramas correspondentes a Figura 2-2 (b). a) $p(r_k)$ , b) $s_k$ Fonte: Autor.....	45
Figura 2-4: a) Figura 2-2 equalizada e b) histograma correspondente. Fonte: Autor.....	46

Figura 2-5: Limiarizações da Figura 2-2 (b), usando a) Otsu global, b) método adaptativo, c) método local, d) limiar selecionado a partir do histograma manualmente. Fonte: Autor.	50
Figura 2-6 Resultado dos algoritmos de detecção de borda: a) detector Sobel, b) detector Canny. Fonte: Autor.	56
Figura 2-7 Os quatro pontos colineares em (a) são mapeados em quatro retas que se cruzam no mesmo ponto no espaço de parâmetros em (b). Fonte: (Gonzalez and Woods 2008).	57
Figura 2-8 Diferentes transformações do espaço da imagem para o espaço de Hough. Fonte: (Gonzalez and Woods 2008).	58
Figura 2-9: Correção de rotação da imagem usando a transformada de Hough. Fonte: Autor.	60
Figura 2-10 a) Exemplo de traço e a sua largura. Fonte: (Epshtain, Ofek et al. 2010).	60
Figura 2-11: Regiões MSER. Fonte: Autor.	64
Figura 2-12: Sequencia dos limiares aplicados a uma imagem para calcular as regiões MSER. Fonte: Autor.	64
Figura 2-13 : Arvore de componente da imagem para o algoritmo MSER. Fonte: Autor.	65
Figura 3-1: Processo de reconhecimento com modelo , a) documento do qual existe o modelo, b) reconhecimento do documento, c) extração do documento segundo o modelo, d) documento do qual não existe modelo, e) o documento não foi reconhecido, não tem pontos em comum com o modelo. Fonte: Autor.	74
Figura 3-2 Criação de Modelo. Fonte: Autor.	74
Figura 3-3: Processo de criação de modelo em MATLAB. Fonte: Autor.	75
Figura 3-4: Sistema proposto por (Ryan and Hanafiah 2015). Fonte: adaptado de (Ryan and Hanafiah 2015).	85
Figura 3-5: Imagens de teste. Fonte: (Ryan and Hanafiah 2015).	85
Figura 3-6: Resultados da extração. Fonte: (Ryan and Hanafiah 2015).	86
Figura 3-7 Imagem usada na etapa de reconhecimento. Fonte: (Ryan and Hanafiah 2015).	87
Figura 3-8: Resultados após o reconhecimento. Fonte: (Ryan and Hanafiah 2015).	88
Figura 4-1: Arquitetura do sistema proposto. Fonte: Autor.	93

Figura 4-2: a) Imagem original , b) imagem em níveis de cinza e filtrada. Fonte: Autor.	94
Figura 4-3: Retificação da imagem usando a transformada de Hough. Fonte: Autor.	95
Figura 4-4: Retificação da imagem e seleção da região de interesse na qual está o documento. Fonte: Autor.	95
Figura 4-5 : a) Resultado do MSER, b) Resultado do algoritmo Canny. Fonte: Autor.	96
Figura 4-6: Uso das heurísticas para eliminar possíveis regiões não textuais. Fonte: Autor.	100
Figura 4-7 Transformada da distância (b) e o esqueleto da imagem (c) para 3 componentes. Fonte: Autor.	101
Figura 4-8 Resultado após da etapa de seleção. Fonte: Autor.	103
Figura 4-9: União dos BBs e formação das linhas de palavras b). Fonte: Autor.	104
Figura 4-10: Arquitetura para o melhoramento do OCR. Fonte: Autor.	106
Figura 4-11 Imagens geradas: Q <sub>f1t1</sub> , Q <sub>f2t1</sub> , Q <sub>f3t1</sub> , Q <sub>f4t1</sub> , Q <sub>f1t2</sub> , Q <sub>f2t2</sub> , Q <sub>f3t2</sub> , Q <sub>f4t2</sub>	108
Figura 5-1: Amostra do banco de imagens criado. Fonte: Autor.	111
Figura 5-2: Imagens de regiões textuais tomadas de imagens de IDs. Fonte: Autor.	112
Figura 5-3: Processo automático de extração das imagens textuais usando modelo. Fonte: Autor.	113
Figura 5-4: a,b) formação de linhas usando o processo de seleção; c,d) formação sem o processo de seleção. Fonte: Autor.	120
Figura 5-5: Número de caracteres selecionados usando como heurística a proporção. Fonte: Autor.	122
Figura 5-6: Percentual de acertos para diferentes resoluções. Fonte: Autor.	123
Figura 5-7: Percentual de acertos para variação de brilho. Fonte: Autor.	124
Figura 5-8: Imagem original e a imagem com as linhas de texto selecionadas. Fonte: Autor.	125
Figura 5-9: Diagrama do sistema de teste. Fonte: Autor.	128
Figura 5-10: Resultado do reconhecimento das palavras numa imagem de teste. 12 linhas (100% do documento) com um 100% de precisão do OCR. Fonte: Autor.	129

Figura 5-11: Palavras corretamente lidas após o uso do algoritmo 2. Fonte: Autor.	.....	131
Figura 5-12: Palavras incorretamente lidas. Fonte: Autor.	.....	131
Figura 5-13: Treinamento do OCR. Fonte: Autor.	.....	131
Figura 5-14: Resultados do OCR com e sem treinamento para letras e números. Fonte: Autor.	.....	132
Figura 5-15: Tempo de execução dos algoritmos 1 e 2, em segundos. Fonte: Autor.	.....	133
Figura 7-1: Similaridades geométricas dos caracteres: alinhamento (marcador em azul), altura (marcadores em vermelho) e espaçamento (marcadores em verde). Fonte: (Tahim 2010).	.....	150
Figura 8-1: Exemplo de segmentação baseada na amplitude: (a) imagem original; (b) segmentação com um único limiar da imagem em (a). Fonte: (da Conceição Palma 2004)	.....	153
Figura 8-2: 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. Fonte: Autor.	.....	155
Figura 8-3: Exemplos da detecção de fronteiras: (a) imagem original; (b), (c) e (d) resultado da detecção de fronteiras utilizando os operadores de Prewitt, Roberts e Robison, respectivamente, para imagem em (a). Fonte: Adaptação de(Gonzalez and Woods 2008).	.....	157
Figura 8-4: Exemplo de segmentação espacial baseada na textura: (a) imagem original constituída por vários tipos de textura; (b) regiões correspondentes à segmentação da imagem em (a). Fonte: (Jain and Yu 1998, da Conceição Palma 2004).	.....	160
Figura 8-5: Exemplos de texturas: (a) textura aleatória; (b) textura determinística. Fonte: (Jain and Yu 1998, da Conceição Palma 2004).	.....	160
Figura 8-6: (a) Imagem com um objeto; (b) contorno do objeto em (a). Fonte: (Jain and Yu 1998, da Conceição Palma 2004).	.....	163
Figura 8-7: Exemplos de objetos simples e complexos, com as respectivas regiões e buracos. Fonte: (Jain and Yu 1998, da Conceição Palma 2004).	.....	164

Figura 9-1: a) Imagem Original, b) após a etapa de Localização, c) após a etapa de Seleção, d) após a etapa de Extração. Fonte: Autor ..... 165

Figura 9-2: Amostra do banco de imagens usado. Fonte: Autor ..... 166

## LISTA DE TABELAS

Tabela 3-1: Palavras extraídas e número de caracteres. Fonte: (Ryan and Hanafiah 2015). .....	86
Tabela 3-2: Sumário das vantagens e desvantagens das técnicas de localização apresentadas. Fonte: Autor. ....	90
Tabela 3-3: Sumário das vantagens e desvantagens dos métodos de seleção de texto. Fonte: Autor.....	91
Tabela 3-4: Resumo das vantagens e desvantagens dos abordagens usadas nos sistemas de reconhecimento de IDs. Fonte: Autor. ....	91
Tabela 3-5: : Resumo das características dos sistemas de extração de texto sem modelos. Fonte: Autor. ....	92
Tabela 4-1 : Porcentagem de regiões filtradas para cada propriedade usada independentemente. Fonte: Autor. ....	98
Tabela 4-2: Porcentagem de regiões filtradas para as propriedade heurísticas quando usadas uma a seguir da outra. Fonte: Autor.....	99
Tabela 4-3: Propriedades heurísticas. t = número de pixels da imagem. Fonte: Autor. ....	100
Tabela 5-1: Avaliação da etapa de seleção de texto. Fonte: Autor. ....	116
Tabela 5-2: Regiões detectadas no banco de imagens 1. Fonte: Autor. ....	117
Tabela 5-3: Desempenho em termos da classificação das regiões para as várias condições heurísticas. Fonte: Autor.....	117
Tabela 5-4: Avaliação da etapa de reconhecimento de texto para caracteres. Fonte: Autor. ....	118
Tabela 5-5: Avaliação da etapa de reconhecimento de texto para palavras. Fonte: Autor.....	118
Tabela 5-6: Avaliação do sistema geral. Fonte: Autor. ....	119
Tabela 5-7: Desempenho em termos da seleção de linhas de texto. Fonte: Autor .120	
Tabela 5-8: Comparação dos resultados da etapa de seleção com (Ryan and Hanafiah 2015). Fonte: Autor. ....	121
Tabela 5-9: Comparação dos resultados da etapa de reconhecimento com (Ryan and Hanafiah 2015). Fonte: Autor. ....	121

Tabela 5-10: Número de iterações: n* para Acurácia do OCR desejado Ac*=85%.	
Fonte: Autor.....	124
Tabela 5-11: Número de iterações: n* para Acurácia do OCR desejado Ac*=95%.	
Fonte: Autor.....	125
Tabela 5-12: Reconhecimento para cada linha nas imagens Q <sub>f1t1</sub> , Q <sub>f2t1</sub> , e Q <sub>f4t2</sub> . Fonte:	
Autor.....	126
Tabela 5-13: Seleção das melhores palavras. Fonte: Autor.....	126
Tabela 5-14: Comparativa dos resultados de saída da função OCR. Fonte: Autor.	130
Tabela 5-15: Distancia Levenshtein do resultado do OCR com e sem treinamento.	
Fonte: Autor.....	132
Tabela 5-16: Tempo de execução dos algoritmos 1 e 2, em segundos. Fonte: Autor.	
.....	133
Tabela 9-1: Resultados experimentais etapa de seleção. Fonte: Autor.	166
Tabela 9-2: Resultados experimentais Etapa de reconhecimento. Fonte: Autor....	167

## LISTA DE ABREVIATURAS E SIGLAS

<b>BB</b>	Caixa delimitadora ( <i>Bounding boxes</i> )
<b>CC</b>	Componentes Conexos ( <i>Connected Component</i> )
<b>CDF</b>	Função de distribuição acumulada ( <i>Cumulative Distribution Function</i> )
<b>CNN</b>	Redes Neurais Convolutivas ( <i>Convolutional Neural Network</i> )
<b>HOG</b>	Histograma de Gradientes Orientados ( <i>Histogram Of Oriented Gradients</i> )
<b>ID</b>	Documento de identificação ( <i>Identity document</i> )
<b>KNN</b>	K vizinhos mais próximos ( <i>K-Neest Neighbors</i> )
<b>MSER</b>	<i>Maximally Stable Extremal Region</i>
<b>OCR</b>	Reconhecimento óptico de caracteres ( <i>Optical character recognition</i> )
<b>ROI</b>	Regiões de interesse ( <i>Region of Interest</i> )
<b>SIFT</b>	<i>Scale-Invariant Feature Transform</i>
<b>SURF</b>	<i>Speeded Up Robust Feature</i>
<b>SVM</b>	Máquina de vetores de suporte ( <i>Support Vector Machine</i> )

## LISTA DE SÍMBOLOS

$\mathbb{Z}$	Conjunto dos números inteiros
<b>E</b>	Domínio das imagens, subconjunto de $\mathbb{Z} \times \mathbb{Z}$
<b>B</b>	Exemplo de elemento estruturante
$\mathbb{Z} \times \mathbb{Z}$	Plano cartesiano discreto
$N_G^{\leftarrow}(\mathbf{p})$	Pixels com conectividade G à esquerda
$N_G^{\rightarrow}(\mathbf{p})$	Pixels com conectividade G à direita
$\delta_B(A)$	Dilatação de uma imagem A por um elemento estruturante B
$\oplus$	Soma de Minkowski
$\varepsilon_B(A)$	Erosão de uma imagem A por um elemento estruturante B
$\ominus$	Subtração de Minkowski
$\gamma_B(A)$	Abertura de uma imagem A por um elemento estruturante B
$\varphi_B(A)$	Fechamento de uma imagem A por um elemento estruturante B
$\Psi d$	Transformada da distância

## SUMARIO

AGRADECIMENTOS.....	7
AGRADECIMIENTOS.....	8
RESUMO .....	9
ABSTRACT .....	10
LISTA DE ILUSTRAÇÕES.....	11
LISTA DE TABELAS .....	16
LISTA DE ABREVIATURAS E SIGLAS.....	18
LISTA DE SÍMBOLOS .....	19
Capítulo 1.....	24
1    Introdução.....	24
1.1    Domínio do problema.....	26
1.2    Extração da informação textual.....	29
1.2.1    Localização.....	31
1.2.2    Seleção .....	32
1.2.3    Extração .....	33
1.2.4    Reconhecimento.....	34
1.3    Motivações e justificativas.....	34
1.4    Objetivos do trabalho .....	36
1.5    Contribuições .....	36
1.5.1    Etapa de localização.....	37
1.5.2    Etapa de seleção.....	38
1.5.3    Etapa de extração e reconhecimento.....	39
1.6    Metodologia .....	39
1.7    Estrutura do trabalho.....	40
Capítulo 2.....	42
2    Fundamentação teórica: Processamento de imagem .....	42

2.1	Introdução .....	42
2.2	Fundamentos de imagens digitais .....	42
2.2.1	Um modelo simples de imagem .....	42
2.2.2	Conversão da imagem para tons de cinza .....	43
2.3	Pré-processamento da Imagem.....	44
2.3.1	Histograma .....	44
2.3.2	Filtragem .....	46
2.4	Limiarização .....	48
2.4.1	Abordagem de limiarização global.....	51
2.4.2	Abordagem de limiarização local .....	52
2.5	Operações morfológicas.....	53
2.6	Detector de bordas.....	54
2.7	Transformada de Hough.....	56
2.7.1	Rotação .....	59
2.8	Largura do traçado (Stroke width) .....	60
2.9	MSER (Maximally Stable Extremal Regions).....	62
2.10	Reconhecimento ótico de caracteres - Tesseract.....	66
	Capítulo 3.....	67
3	Trabalhos relacionados. Sistemas de extração de texto mais relevantes .....	67
3.1	Etapas fundamentais no processo de reconhecimento de texto.....	67
3.1.1	Etapa de localização .....	68
3.1.2	Etapa de seleção .....	75
3.1.3	Etapa de extração .....	78
3.1.4	Etapa de reconhecimento .....	80
3.2	Trabalhos selecionados.....	82
3.3	Comentários finais.....	88
	Capítulo 4.....	93
4	Sistema de extração automática de texto proposto .....	93

4.1	Etapa de localização das regiões candidatas a serem texto.....	93
4.1.1	Pré-processamento.....	94
4.2	Etapa de seleção das regiões que possuem realmente caracteres.....	96
4.3	Etapa de reconhecimento .....	103
4.3.1	Mesclar caracteres em palavras .....	104
4.3.2	Melhoria do OCR e escolha das melhores palavras.....	104
4.4	Comentários finais.....	109
	Capítulo 5.....	110
5	Validação e resultados .....	110
5.1	Elaboração do banco de imagens .....	110
5.2	Métodos de avaliação do desempenho do sistema .....	113
5.2.1	Distância de Levenshtein .....	115
5.3	Avaliação da etapa de seleção de texto .....	115
5.4	Avaliação da etapa de reconhecimento de texto.....	117
5.5	Avaliação do sistema geral.....	119
5.6	Experimentos com as heurísticas e os parâmetros .....	122
5.7	Realização de uma prova de conceito do sistema em um ambiente real .....	128
5.7.1	Experimentos e resultados da prova de conceito .....	128
5.8	Análise de desempenho .....	132
5.9	Considerações finais.....	134
6	Conclusões e trabalhos futuros.....	136
6.1	Próximas etapas .....	137
	Publicações .....	139
	Referências .....	140
7	Apêndice A - Características textuais .....	150
8	Apêndice B - Etapa de localização .....	153
	Segmentação espacial .....	153
	Correspondência de modelos.....	160

Classes principais de descritores de forma.....	163
9 Apêndice C – Teste do sistema com imagens da Web.....	165
Imagens da Web.....	165
Avaliação da etapa de seleção de texto em imagens da Web .....	165
Avaliação da etapa de reconhecimento de texto em imagens da Web.....	167

# Capítulo 1

## 1 Introdução

A quantidade de dados disponível em formato digital na rede mundial de computadores tem aumentado incessantemente. De acordo com estimativas realizadas em 2014, de 2013 a 2020 o universo digital irá aumentar de 4,4 trilhões de gigabytes para 44 trilhões de gigabytes (Turner, Gantz et al. 2014). Parte dos dados no universo digital está no formato textual, como: e-mails, relatórios, boletins, artigos, registros de pacientes e conteúdo de páginas Web. Grande parte dessas informações em formato textual são imagens digitalizadas de documentos, nos diversos setores da atividade humana.

Segundo a AIIM International (Association for Information and Image Management International) nos últimos 50 anos, a humanidade gerou a mesma quantidade de informação que nos 5 mil anos anteriores. Além disso, diversas organizações mantêm enormes bases de dados na forma de imagens e vídeos com interesse em pesquisas médicas, entretenimento, comércio, segurança, etc. Cada vez mais estamos gerando maiores quantidades de documentos. Contudo, os sistemas de busca de tais arquivos são baseados em texto, consequentemente, para que a pesquisa e recuperação de um determinado arquivo seja eficiente, cada arquivo deve ser textualmente descrito por intervenção humana. Processar, organizar ou gerenciar essa grande quantidade de dados textuais manualmente exige um grande esforço humano, sendo muitas vezes impossível de ser realizado. Além disso, há conhecimento embutido nos dados textuais, e analisar e extrair conhecimento de forma manual também torna-se inviável devido à grande quantidade de textos.

Uma vez que a descrição manual de cada arquivo é inviável e subjetiva, torna-se necessário para gerenciamento, indexação e recuperação de tais arquivos, sistemas capazes de descrever o conteúdo de imagens automaticamente. Obter a informação rapidamente ou prover “a informação correta, à pessoa correta, no tempo correto” é difícil e custoso e demorado com processos manuais. Com isso, técnicas computacionais que requerem pouca intervenção humana e que permitem a extração de conhecimento de grandes quantidades de textos têm ganhado destaque nos

últimos anos e vêm sendo aplicadas tanto na academia quanto em empresas e organizações (Biemann and Mehler 2014).

Reconhecimento de texto automático é um dos problemas mais difíceis em visão computacional. Embora muitos métodos de detecção de texto foram estudados no passado, o problema permanece em aberto (Jain and Yu 1998, Epshtain, Ofek et al. 2010, Yao, Bai et al. 2012, Lukas Neumann 2015, Ryan and Hanafiah 2015, Sun, Huo et al. 2015, Jaderberg, Simonyan et al. 2016). Um pré-requisito essencial para o reconhecimento de texto é localizar o texto dentro da imagem. Esta continua sendo uma difícil tarefa por causa da ampla variedade na formatação do texto, como: variações de fonte, espessura, cor, tamanho, textura e distorções geométricas (Gonzalez, Bergasa et al. 2012). Recentemente a extração de texto tem sido amplamente abordada, porém a maioria dos sistemas descritos na literatura são dedicados a contextos específicos, tais como: o reconhecimento de endereços em envelopes (Jain and Bhattacharjee 1992, Palumbo, Srihari et al. 1992), identificação de placas veiculares permitindo o monitoramento e fiscalização de possíveis infratores (Arth, Limberger et al. 2007, Anagnostopoulos, Anagnostopoulos et al. 2008, Gonçalves, da Silva et al. 2016), busca de cenas específicas por meio das legendas e créditos em bancos de dados de vídeo (Luccheseyz and Mitray 2001), pesquisas na Web (Antonacopoulos, Karatzas et al. 2001), dentre muitas outras.

Relativamente poucos sistemas consideram a extração de texto em documentos genéricos e menos ainda tem abordado o problema de reconhecimento de texto em documentos de identificação (IDs). Os IDs são uma das principais fontes para a obtenção de informações sobre um cidadão. A informação contida nos documentos de identificação é usada em processos de registro, sistemas de verificação de dados, abertura de contas, etc. Em geral, são usados formulários preenchidos de acordo com os dados dos documentos de identificação, que são convertidos em dados digitais através de um processo manual de digitação das informações. No entanto, o processo manual é demorado e propenso a erro ou até fraude. Com a tecnologia ficando cada vez mais sofisticada, as exigências dos usuários estão cada vez maiores, sendo necessário soluções automáticas que diminuam o trabalho humano.

A informação extraída precisa ser exata, uma vez que um erro de reconhecimento pode gerar um erro durante o registo, fazendo com que o sistema se

torne complexo. A maioria das abordagens hoje existentes consideram separadamente as etapas dos sistemas de extração e reconhecimento de texto, trabalhos relacionados concentram-se em: localização do texto (Li, Lu et al. 2012, Risnumawan, Shivakumara et al. 2014, Yin, Yin et al. 2014), retificação do texto (Yin, Chen et al. 2011, Yu-peng Gao 2011, Yonemoto 2014), extração e reconhecimento de texto (Chang 2013, Gonzalez and Bergasa 2013, Lukas Neumann 2015, Jaderberg, Simonyan et al. 2016). Só alguns poucos trabalhos abordam o problema como um conjunto (Sonia Bhaskar 2011, Ryan and Hanafiah 2015). Neste cenário, técnicas de reconhecimento de textos em imagens de documentos de identificação continuam sendo amplamente pesquisadas.

Neste trabalho é apresentado um sistema automático de reconhecimento de texto em imagens de IDs, assim como as aplicações e a metodologia para a validação dos resultados. Finalmente, é realizada uma prova de conceito do sistema proposto em um ambiente real no Laboratório de Arquitetura e Redes de Computadores (LARC) da Universidade de São Paulo.

### **1.1 Domínio do problema**

Os sistemas capazes de extrair a informação textual de imagens são conhecidos como sistemas de extração e reconhecimento de texto. As abordagens para a localização e reconhecimento de texto utilizados nestes sistemas estão intimamente relacionadas à maneira que o texto está inserido na imagem e ao plano de fundo ao qual o texto está sobreposto. A seguir se descrevem os tipos de imagem de acordo com o plano de fundo e o tipo de texto que possuem. Os pesquisadores costumam definir três categorias de imagens quanto ao tipo de texto que elas possuem (Jung, Kim et al. 2004): 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 e possuindo um alto contraste com o 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 separando o texto dos gráficos e figuras. Após tal identificação, sistemas conhecidos como 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 Figura 1-1.

As imagens-artificiais são caracterizadas por textos que são sobrepostos a uma determinada imagem 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, frequentemente 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 livros e revistas, como ilustrado na Figura 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.

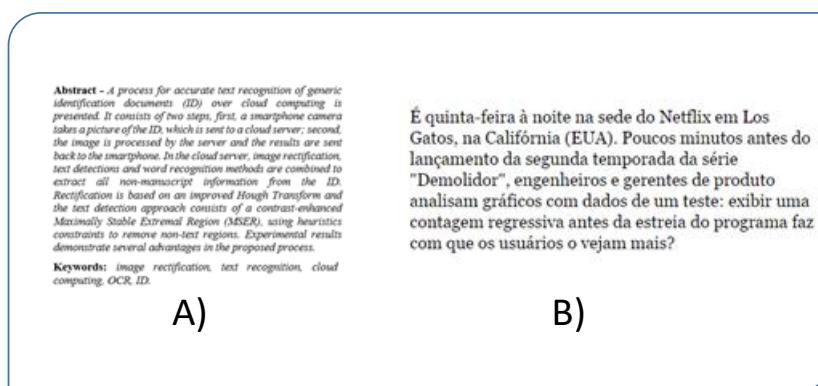


Figura 1-1: Exemplos de imagens-documento: (a) trecho de um artigo, (b) notícia de jornal. Fonte: Autor.

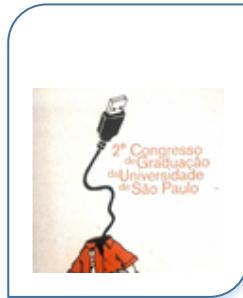


Figura 1-2: Exemplo de imagem-artificial (imagem colorida com caracteres).

Fonte: Autor.

As imagens em que os caracteres fazem naturalmente parte da cena são definidas como imagens-cena. A extração de texto de tais imagens apresenta 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 são apresentadas na Figura 1-3.

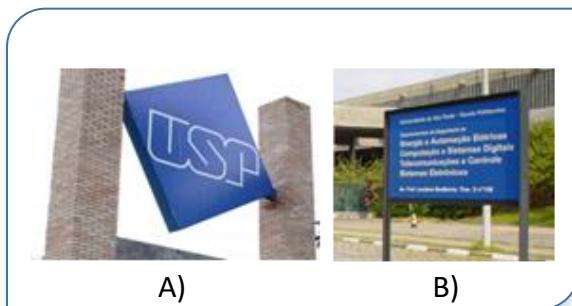


Figura 1-3: Exemplos de imagens-cena: (a) imagem contendo variações de iluminação sobre os caracteres devido às distorções geométricas, (b) imagem contendo caracteres com distorção de perspectiva e iluminação não-uniforme. Fonte: Autor.

Esses dois conjuntos de imagem (imagens-artificiais e imagens-cena) são englobados por um conjunto ainda mais amplo denominado imagens complexas, definido por Zhong et al. (Zhong, Karu et al. 1995) como: “Imagens em que os caracteres não podem ser segmentados do plano de fundo usando simples técnicas de limiarização, e a cor, tamanho, fonte e orientação do texto são desconhecidos”.

Este trabalho visa a extração de texto em imagens complexas de documentos, especificamente de IDs e o objetivo é ler e reconhecer o conteúdo deles. Assim, a saída desejada é a informação do ID. Exemplos são apresentadas na Figura 1-4.



Figura 1-4: (a) imagem-artificial, (b) imagem-cena, de documentos de identificação. Fonte: Autor.

## 1.2 Extração da informação textual

Os sistemas de OCR atuais possuem uma alta taxa de reconhecimento para imagens-documento simples, porém tais sistemas são incapazes de reconhecer a informação textual em imagens complexas (Zhong, Karu et al. 1995). Um sistema de OCR convencional aplica binarizações locais ou globais à imagem em níveis de cinza de alta resolução (100-300 dpi) (Trier, Jain et al. 1996), 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).

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 do reconhecimento de padrões e inteligência artificial (Chen, Bourlard et al. 2001). Existem diversos sistemas de OCR com taxas de reconhecimento que variam entre 95% e 99% para imagens-documento. Em decorrência disso, a solução apresentada pela maioria dos sistemas de extração de texto 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 de extração de texto 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 Figura 1-5 apresenta a transformação de uma imagem complexa em uma imagem binária adequada ao reconhecimento dos caracteres por sistemas de OCR convencionais.

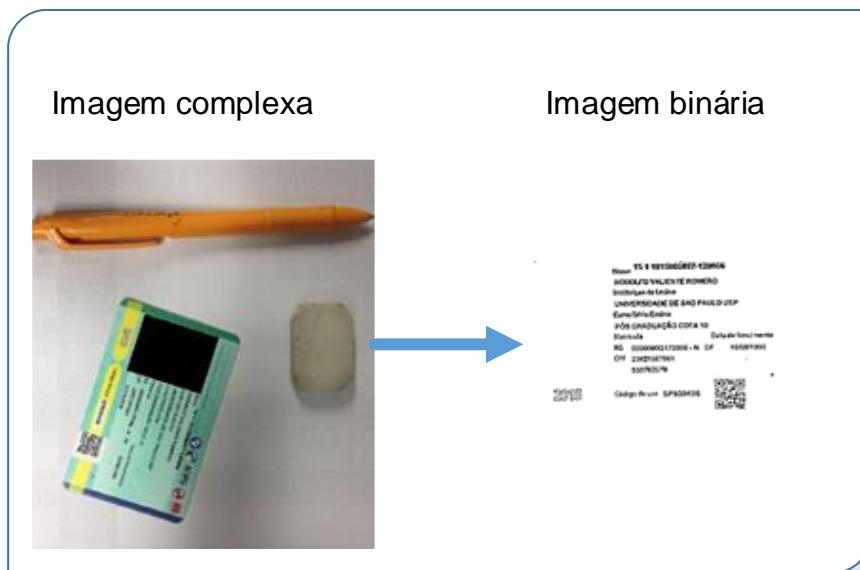


Figura 1-5: Transformação realizada por um sistema de extração de texto com o objetivo de separar os caracteres do plano de fundo complexo. Fonte: Autor.

Sistemas de extração da informação textual em imagens complexas recebem como entrada uma imagem ou sequência de imagens possuindo texto (em que tais imagens podem ser coloridas ou em níveis de cinza), retornando texto plano como saída (Jung, Kim et al. 2004).

Esses sistemas geralmente são divididos em quatro subsistemas ou etapas: (i) localização das regiões candidatas a serem texto; (ii) seleção das regiões que possuem realmente caracteres; (iii) extração e correção do texto selecionado; (iv) reconhecimento do texto. Os três primeiros são responsáveis pela adaptação da imagem complexa a ser reconhecida pelo sistema de OCR, como ilustrado na Figura 1-5. A arquitetura completa de um sistema de extração da informação textual está ilustrada no diagrama de blocos da Figura 1-6.

As subseções seguintes descrevem o propósito de cada etapa (veja Figura 1-6).

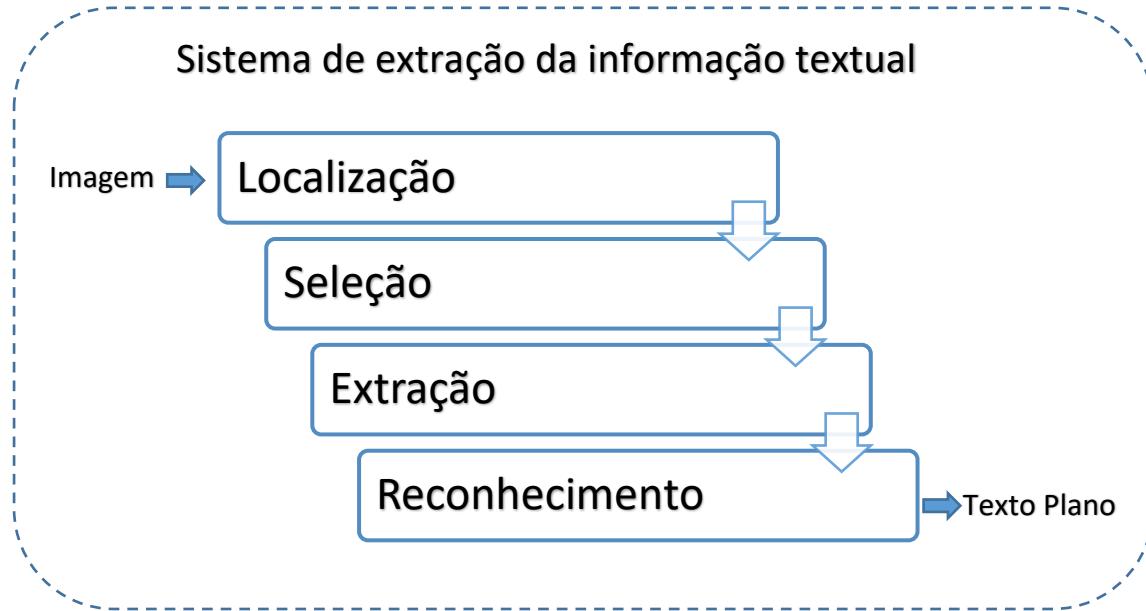


Figura 1-6: Arquitetura de um sistema de extração da informação textual. Fonte: Autor.

### 1.2.1 Localização

A etapa de localização das regiões candidatas a serem texto tem como propósito responder à pergunta: “onde está o texto na imagem?”. Segundo Zhong et al. (Zhong, Karu et al. 1995), é impraticável reconhecer 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 Figura 1-5, 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 exemplificar

melhor o processo de localização, a Figura 1-7 (a), é submetida a um algoritmo de localização que delimita os caracteres individualmente, como apresentado na Figura 1-7(b).

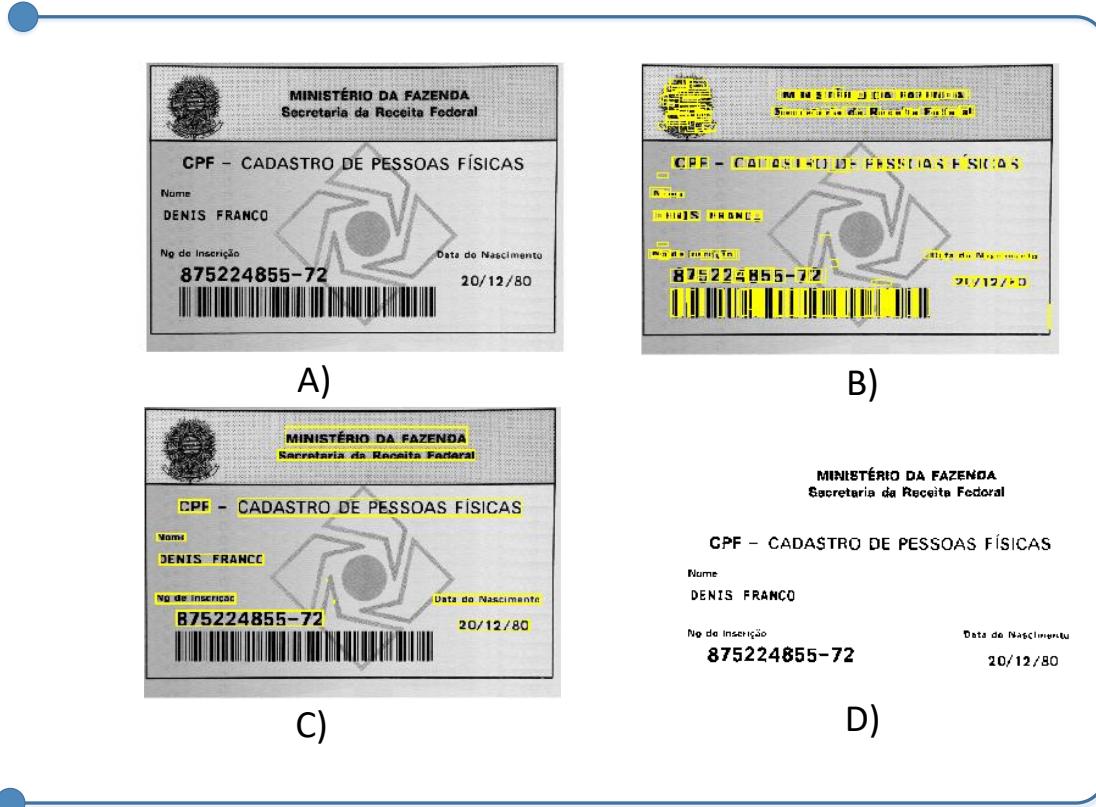


Figura 1-7: Sequência de resultados das 3 primeiras etapas de um Sistemas de extração da informação textual: (a) imagem original, (b) resultado da etapa de localização, (c) resultado após a etapa de seleção, (d) imagem binária após a etapa de extração. Fonte: Autor.

### 1.2.2 Seleção

Uma vez que os algoritmos de localização geralmente utilizam poucas características texturais para selecionar as regiões da imagem candidatas a texto, após a etapa de localização, diversas regiões que não representam caracteres estão delimitadas por BBs (falsos positivos) Figura 1-7(b). A etapa de seleção tem como objetivo fazer uma seleção refinada dos BBs obtidos na etapa de localização, selecionando as regiões que possuem realmente caracteres, 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 seleção 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 seleção 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 seleção 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 Figura 1-7(c) apresenta o resultado da etapa de seleção , em que os BBs remanescentes foram os classificados como textuais.

Muitos autores consideram a seleção (também chamada verificação por alguns autores) como parte integrante da etapa de localização, visto que as regiões localizadas e consideradas textuais só são conhecidas após a etapa de seleção. No entanto, neste trabalho considera-se a subdivisão em duas etapas (localização e seleção) visando facilitar a compreensão.

### **1.2.3 Extração**

Após as etapas de localização e seleção , as regiões consideradas textuais estão delimitadas por BBs, Figura 1-7(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 Figura 1-7(d).

### 1.2.4 Reconhecimento

Após as três primeiras etapas do sistema de extração da informação textual, a imagem apresenta-se na forma binária com os caracteres sobre um plano de fundo homogêneo, Figura 1-7(d). A última etapa geralmente é constituída de um OCR convencional e tem como objetivo responder à 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, retornando os caracteres em texto plano (geralmente ASCII ou HTML), como mostrado na Figura 1-8.

No presente trabalho é utilizado o OCR Tesseract (código aberto) (Smith 2007) atualmente mantido pela empresa Google Inc. O Tesseract assume que a imagem fornecida é binária com regiões de texto poligonais opcionais definidas.



Figura 1-8: 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). Fonte: Autor.

Mesmo uma vez extraído o texto e binarizado, o sistema OCR pode ter erro, como consequência de uma imagem muito complexa, ruidosa ou em baixa resolução. Existem melhorias possíveis a serem aplicadas antes do uso do OCR (Burie, Chazalon et al. 2015, Urbschat, Meier et al. 2015, Walha, Drira et al. 2015).

### 1.3 Motivações e justificativas

Atendendo que os documentos de identificação tornam-se a referência principal na obtenção de informações de um cidadão. Sendo que em muitas situações, o cliente

é obrigado a mostrar o ID e as informações contidas nele são coletadas manualmente (O cliente ou um funcionário precisa digitar os dados um a um no computador ou outro meio), um processo lento e ineficiente. Em vez de digitar manualmente os dados, um sistema pode ser proposto para extrair a informação, fornecendo a imagem do cartão de identificação para ser processada e produzindo dados textuais como resultado.

Os sistemas automáticos de reconhecimento de texto permitem processar, organizar e gerenciar os dados textuais, que manualmente exigem um grande esforço, sendo muitas vezes impossível de ser realizado (Ryan and Hanafiah 2015, Rossi 2016, Walha, Drira et al. 2016). Os algoritmos de localização, extração e reconhecimento de texto fazem possível a extração da informação existente nos documentos, tornando viável analisar, e extrair conhecimento embutido dessas informações textuais. Os resultados recentes apresentados na literatura têm demonstrado grandes avanços na área, diminuído consideravelmente o trabalho manual (Yin, Yin et al. 2014, Yonemoto 2014, Lukas Neumann 2015).

Ainda assim, alguns desafios precisam ser enfrentados para tornar os algoritmos mais eficientes e aplicáveis no cenário de reconhecimento de texto em IDs. Dentre eles podemos citar:

- Os métodos de reconhecimento de texto atuais estão dedicados a contextos específicos, tais como: o reconhecimento de endereços, identificação de placas veiculares e busca de cenas, porém, relativamente poucos sistemas consideram a extração de texto em IDs genéricos.
- Os métodos de reconhecimento de texto em IDs devem integrar de forma inteligente e eficiente métodos existentes e novos para prover um resultado de alta qualidade.
- Os métodos de reconhecimento de texto em IDs devem prover o melhor resultado possível na etapa de OCR. Os sistemas de OCR atuais possuem uma alta taxa de reconhecimento, porém para imagens complexas não oferece o resultado desejado.

Tendo em conta que a informação contida nos documentos de identificação é usada em processos de registro, sistemas de verificação de dados, abertura de contas, etc. Nos quais a informação é transformada em dados digitais através de um processo manual de digitação, processo demorado e propenso a erro ou até fraude. Neste

cenário, com a tecnologia ficando cada vez mais sofisticada e as exigências dos usuários estão cada vez maiores, precisam-se soluções automáticas que diminuam o trabalho humano.

#### **1.4 Objetivos do trabalho**

Considerando as motivações e justificativas expostas, o objetivo do trabalho é desenvolver um sistema automático de reconhecimento de texto em imagens de IDs, e para alcançá-lo propõe-se uma arquitetura que integra eficientemente os diferentes algoritmos de reconhecimento de imagens e na etapa final um método iterativo para melhorar o resultado do OCR.

O trabalho também tem os seguintes objetivos específicos:

- Descrever as principais abordagens empregadas no reconhecimento de texto em imagens de documentos.
- Avaliar o método proposto em diversos conjuntos de imagens, que contenham variação de luminosidade, dimensão, fonte e orientação.
- Comparar os desempenhos com outros trabalhos e analisar os resultados.

#### **1.5 Contribuições**

Uma vez que este trabalho propõe um sistema extração e reconhecimento de texto para imagens de IDs, diversos problemas surgem em praticamente todas as etapas descritas anteriormente. Tais problemas são devido ao grande número de fatores que variam em imagens, tais como: dimensão, fonte e orientação dos caracteres, textura do plano de fundo, dimensões da imagem, etc. Esta seção apresenta os diversos problemas relacionados ao reconhecimento textual em imagens complexas como também as contribuições do trabalho em cada etapa da arquitetura proposta que visam solucioná-los.

As principais contribuições deste trabalho são:

- Desenvolvimento de uma arquitetura para o reconhecimento de texto em IDs genéricos; a arquitetura é apresentada no capítulo 4 e oferece melhorias na implementação das diferentes etapas do reconhecimento de texto, que serão apresentadas a seguir.

As contribuições parciais foram publicadas em: (Valiente, Sadaike et al. 2016); (Valiente and Bressan 2016). (Valiente, Gutiérrez et al. 2017) e (Gutiérrez, Valiente et al. 2017).

- Uma abordagem alternativa para o problema de reconhecimento de texto em imagens de IDs através de implementação de melhorias na etapa final da arquitetura, usando um método iterativo para melhorar o resultado do OCR (Valiente, Sadaike et al. 2016).

Contribuições colaterais deste trabalho são:

- Uma revisão bibliográfica detalhada do estado da arte.
- Uma investigação para facilitar o reconhecimento de texto em imagens de IDs que serve como referência em outras áreas de aplicação.
- Realização de uma prova de conceito do sistema em um ambiente real, na qual foi implementado e testado seu uso na leitura automática das informações de documentos de identificação. (Valiente, Gutiérrez et al. 2017) e (Gutiérrez, Valiente et al. 2017).
- Uso do sistema proposto para reconhecimento de imagens com texto em páginas Web (Valiente, Gutiérrez et al. 2017).

Os desafios e contribuições nas diferentes etapas são apresentados a seguir. Todas as contribuições que serão apresentadas foram divulgadas à comunidade acadêmica em congressos nacionais e internacionais anteriormente citados, futuros trabalhos e artigos decorrentes desta dissertação, estão em fase de realização.

### **1.5.1 Etapa de localização**

A construção de um sistema extração de texto em IDs é desafiador devido à diversidade de imagens de entrada. Como consequência, o sistema deve localizar as regiões textuais por meio de características menos dependentes dos parâmetros variantes em uma imagem genérica.

O método de localização proposto explora a característica de contraste existente entre os pixels de contorno e o plano de fundo nos caracteres legíveis. É usado o algoritmo MSER junto com uma melhoria do contraste e aproveitando a informação das bordas dos objetos da imagem, para localizar os possíveis caracteres da imagem.

O método de localização proposto é baseado no trabalho de (Yin, Yin et al. 2014), no qual os autores demostram que o MSER tem um ótimo desempenho para localizar regiões que possuam texto pois a consistência da cor e o alto contraste do texto resultam em perfis de intensidade estáveis. MSER é invariante para contínuas transformações geométricas e mudanças de intensidade e escalas, é invariante ao tipo de fonte e dimensão dos caracteres, permitindo que o método de localização obtenha êxito independentemente do tipo de caractere presente na imagem.

No entanto, diferentemente do trabalho de Yin et al., é acrescentada a detecção de bordas e a melhoria do contraste adaptados a IDs. Tal método é menos sensível ao ruído e oferece melhores resultados no processo de localização.

Os resultados demonstram 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. Além disso, possui a vantagem de identificar os caracteres individualmente.

### **1.5.2 Etapa de seleção**

Uma vez que o método proposto de localização identifica as regiões candidatas a texto usando um método baseado em MSER, o método de seleção proposto utiliza métodos heurísticos e estruturais para a certificação de tais regiões como textuais. A utilização de diferentes abordagens promove robustez na identificação de regiões textuais, porém não inserindo grande aumento de complexidade computacional, visto que os algoritmos de seleção são apenas aplicados às áreas previamente delimitadas na etapa de localização.

A etapa de seleção proposta neste trabalho desenvolveu-se mediante a busca de heurísticas, extraídos da imagem, capazes de classificar as regiões localizadas como textuais e não-textuais. O método é baseado no trabalho de (Gonzalez, Bergasa et al. 2012) , onde são usadas um conjunto de heurísticas para verificar os caracteres.

No entanto, diferentemente do trabalho de Gonzalez et al., são calculados e propostos novos valores das heurísticas, específicos para a seleção do texto em IDs, obtendo uma melhor classificação das regiões localizadas. É desenvolvido também, aproveitando as heurísticas e a informação da largura do traço de cada caractere, um classificador binário, texto e não-texto que melhora o desempenho do sistema.

### **1.5.3 Etapa de extração e reconhecimento**

Nesta dissertação não são consideradas contribuições na etapa de extração, esta etapa é incluída como pré-processamento antes do reconhecimento. A solução apresentada pela maioria dos sistemas de extração de texto para imagens complexas é transformá-las em imagens com as características de imagens-documento, acoplando ao final do processo um sistema de OCR. Os métodos desenvolvidos resolvem, em certas condições, o problema do pré-processamento antes do OCR, porém existem ainda algumas limitações. As maiores dificuldades advêm da existência de: caracteres com diferentes tamanhos, orientações e perspectivas; diferente espaçamento entre os caracteres na mesma linha (que dificulta o seu agrupamento em palavras) e fraco contraste dos caracteres em relação ao fundo.

Diante dos problemas citados, este trabalho propõe um método de organização dos caracteres alinhados horizontalmente independente das dimensões dos caracteres e da imagem. Tal método é utilizado como uma etapa de pré-processamento na etapa de reconhecimento dos caracteres.

Além disso, tendo em conta que o reconhecimento de texto é diferente para cada imagem é usado um processo iterativo de criação de imagens e escolha das melhores palavras para aperfeiçoar o OCR. Finalmente é realizada uma prova de conceito do sistema proposto em um cenário real, demonstrando a viabilidade e funcionalidade do sistema.

## **1.6 Metodologia**

Para a realização e validação deste trabalho, as seguintes etapas são contempladas de acordo com a sequência a seguir:

- Revisão de literatura das principais técnicas disponíveis para a localização, extração e reconhecimento automático de texto em imagens, de forma a identificar algumas oportunidades de melhoria no aspecto de qualidade e eficiência no processo final.
- Estudo e análise dos principais trabalhos relacionados a extração automática de texto em imagens de IDs. A partir desta análise, foi possível identificar os problemas e dificuldades associadas aos métodos de reconhecimento, considerando as diversas etapas do processo.

- Implementação de uma arquitetura que integra eficientemente os diferentes algoritmos de localização, extração e reconhecimento aplicado ao reconhecimento de texto de IDs genéricos, superando algumas das limitações anteriormente identificadas.
- Realização de várias simulações computacionais para diferentes cenários de aplicação, utilizando a ferramenta MATLAB.
- Implementação de um método iterativo para melhorar o resultado do OCR.
- Análise dos resultados obtidos e realização de novos testes para diferentes cenários de aplicação em MATLAB.
- Avaliação do desempenho do sistema, para vários tipos de cenários tendo em conta métricas objetivas.
- Implementação de uma prova de conceito do sistema em um cenário real.
- Extensão do algoritmo para outras aplicações.

## **1.7 Estrutura do trabalho**

Para melhor situar o leitor no que se refere à estrutura deste trabalho será feita uma breve apresentação do conteúdo do mesmo. Esta Dissertação é composta, além deste capítulo introdutório (Capítulo 1), de mais cinco capítulos (Capítulo 2 - 6) e as referências bibliográficas, conforme detalhamento feito a seguir:

No Capítulo 2, apresenta-se a fundamentação teórica e conceitos necessários para o entendimento do trabalho, explica-se com mais detalhes os tópicos de maior relevância para a compreensão do trabalho na área de processamento de imagens.

No Capítulo 3 descrevem-se os principais trabalhos de reconhecimento automático de texto em IDs, buscando correlacionar e contextualizar com o tema proposto nesta dissertação.

No Capítulo 4, é apresentada e explicada a arquitetura proposta, descrevendo o funcionamento.

No Capítulo 5 descrevem-se a implementação e validação da arquitetura proposta, e são apresentadas as melhorias e aperfeiçoamentos obtidos. Também são

realizadas simulações computacionais em diferentes cenários e comparados com os outros métodos propostos na literatura, utilizando métricas objetivas e subjetivas.

No Capítulo 6, são apresentadas as conclusões e pontos para futuras pesquisas.

Ao longo deste trabalho serão usados os termos precisão (por *precision*), revocação (por *recall*) e *F-score* (ou *F-measure* tomado do inglês).

## Capítulo 2

### 2 Fundamentação teórica: Processamento de imagem

#### 2.1 Introdução

Neste capítulo são revisados alguns conceitos fundamentais para o entendimento do que será apresentado a seguir. Inicialmente são abordados brevemente os fundamentos das imagens digitais como a definição do que é imagem digital e suas características, mais informações podem ser encontradas em (Gonzalez and Woods 2008). Também são abordadas técnicas voltadas ao tratamento de imagens, e é realizado um estudo dos algoritmos usados neste trabalho. A seguir são definidos e apresentados os principais conceitos e algoritmos de processamento de imagens úteis para a extração automática de texto em imagens de documentos.

#### 2.2 Fundamentos de imagens digitais

##### 2.2.1 Um modelo simples de imagem

Uma imagem pode ser definida através de uma função bidimensional  $f(m,n)$ , onde para qualquer par  $(m,n)$  existe um valor  $f$  proporcional à intensidade do brilho da imagem naquele ponto. As coordenadas espaciais  $(m,n)$  localizam qualquer ponto pertencente a imagem em questão (Gonzalez and Woods 2008). Numa imagem digital  $m,n$  e  $f(m,n)$  são quantias finitas e discretas. Estes pontos são chamados elementos da imagens ou "pixels" que formam as imagens digitais como pode ser observado na Figura 2-1.



Figura 2-1: Imagem original e demonstração de parte dos pixels. Fonte: Autor.

## 2.2.2 Conversão da imagem para tons de cinza

A conversão de uma imagem colorida para tons de cinza é o primeiro passo de inúmeros algoritmos de análise de imagens, já que reduz a quantidade de informação de uma imagem. Embora haja redução, a maioria das informações relacionadas às características da imagem se preservam, tais como: bordas, regiões, junções, etc.

Uma imagem no espaço de cores RGB, é convertida para uma imagem em tons de cinza, Icinza, por meio da transformação mostrada pela Equação 2-1.

$$I_{cinza}(m, n) = \alpha I_{RGB}(m, n, r) + \beta I_{RGB}(m, n, g) + \gamma I_{RGB}(m, n, b) \quad (\text{Equação 2-1})$$

Na qual  $(m, n)$  são os índices de um pixel na imagem em escala de cinza,  $(m, n, c)$  diz respeito ao pixel com localização  $(m, n)$  no canal  $c$  da imagem colorida, sendo que  $c$  assume as variáveis  $r$ ,  $g$  e  $b$ , que se referem, respectivamente, aos canais de cores vermelho, verde e azul.

Nota-se, portanto, que uma imagem em escala de cinza é uma combinação linear dos canais de cores de uma imagem no espaço de cores RGB. O peso dos coeficientes ( $\alpha$ ,  $\beta$  e  $\gamma$ ) são atribuídos com a finalidade de que o olho humano perceba a imagem em tons de cinza da mesma forma que a imagem colorida. A Figura 2-2 ilustra o resultado da conversão de uma imagem colorida para uma em escala de cinza.



Figura 2-2 Ilustração da conversão de uma imagem colorida para uma em escala de cinza (a) imagem colorida, (b) imagem em escala de cinza. Fonte: Autor.

## 2.3 Pré-processamento da Imagem

O pré-processamento de imagens procura corrigir os defeitos na imagem através de algoritmos eficientes. Para melhorar a qualidade das imagens é necessário utilizar técnicas de realce de modo que a imagem resultante seja mais adequada que a imagem original (Bovik 2009).

As técnicas de melhoria da qualidade de imagens podem ser divididas em duas famílias: as de realce e as de restauração de imagens. Quando a melhoria é usada para combater um processo de degradação conhecido ou avaliado por métodos da teoria da filtragem, a palavra usada é restauração de imagens. A restauração difere de realce pelo fato de que a primeira procura obter a imagem “real” tendo, se possível, um conhecimento a priori da degradação. Neste trabalho são usadas técnicas de realce da imagem, nas etapas de pré-processamento, que são necessárias para melhorar o resultado das seguintes etapas.

### 2.3.1 Histograma

De acordo com (Gonzalez and Woods 2008), o histograma é uma relação que mapeia, para cada valor de intensidade que um pixel possivelmente possa ter, o número de vezes em que ela aparece na imagem, é uma tabela das frequências de cada valor ou faixa de valores de intensidade nos pixels da imagem. Estes valores são normalmente representados por um gráfico de barras que fornece para cada nível de cinza o número (ou percentual) de pixels correspondentes na imagem. Através da visualização do histograma de uma imagem obtém-se uma indicação de sua qualidade quanto ao nível de contraste e quanto ao seu brilho médio (se a imagem é predominantemente clara ou escura). O histograma normalizado é dado pela Equação 2-2.

$$p_r(r_k) = \frac{n_k}{n} \quad (\text{Equação 2-2})$$

sendo:  $0 \leq r_k \leq 1$ ;  $k = 0, 1, \dots, L-1$ , onde  $L$  é o número de níveis de cinza da imagem digitalizada;  $n$  = número total de pixels na imagem;  $p_r(r_k)$  = probabilidade do  $k$ -ésimo nível de cinza;  $n_k$  = número de pixels cujo nível de cinza corresponde a  $k$  (Gonzalez and Woods 2008). Na Figura 2-3a, segue o histograma normalizado da Figura 2-2 (b).

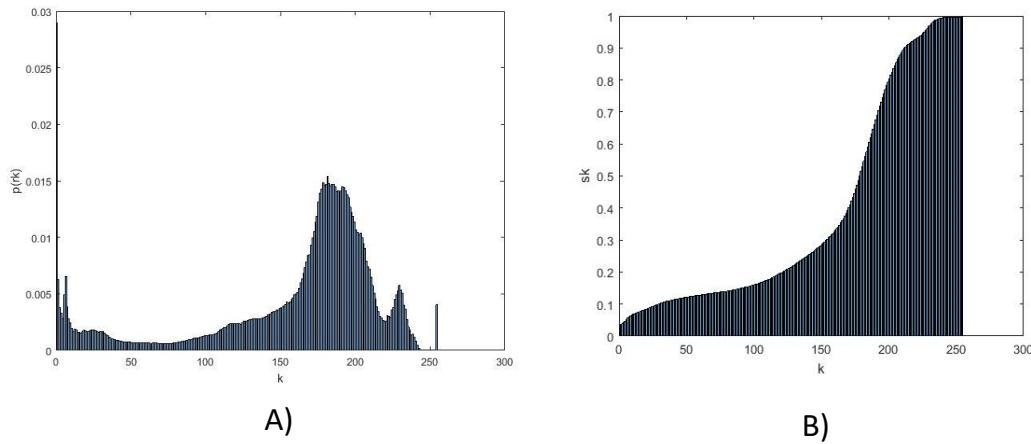


Figura 2-3: Histogramas correspondentes a Figura 2-2 (b). a)  $p(r_k)$ , b)  $s_k$  Fonte: Autor.

A Figura 2-3 mostra que os níveis de cinza estão concentrados em direção à extremidade clara do intervalo de níveis de cinza, ou seja, esse histograma corresponde a uma imagem com características predominantemente claras. Se uma imagem não está utilizando todos os níveis de cinza disponíveis, pode-se alterá-la, para melhorar o contraste. Para manipular o histograma de uma maneira consistente e significativa usa-se o processo de equalização. Equalizar o histograma significa obter a máxima variância do histograma de uma imagem, obtendo assim uma imagem com o melhor contraste. A forma mais usual de se equalizar um histograma é utilizar a função de distribuição acumulada (CDF - cumulative distribution function) segundo a Equação 2-3. Na Figura 2-3b, segue a CDF normalizada da Figura 2-2 (b). O resultado da Figura 2-2 equalizada e seu histograma equalizado são mostrados na Figura 2-4 :

$$s_k = T(r_k) = (L - 1) \sum_{j=0}^k \frac{n_j}{n} = \sum_{j=0}^k p_r(r_j) \quad (\text{Equação 2-3})$$

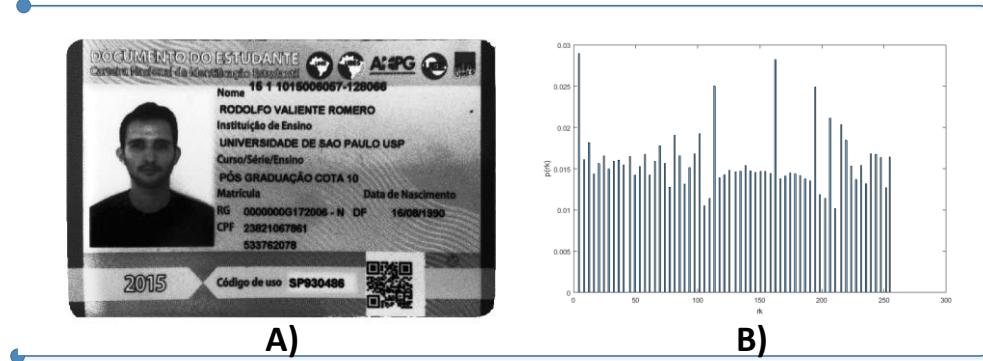


Figura 2-4: a) Figura 2-2 equalizada e b) histograma correspondente. Fonte: Autor.

Nesta dissertação a equalização de histograma é usada como uma etapa de pré-processamento antes da aplicação do OCR.

### 2.3.2 Filtragem

Nas técnicas de filtragem, o processamento de um nível de cinza de um pixel depende dos valores de nível de cinza desse pixel e de seus pixels vizinhos. Em geral, na vizinhança, os pixels mais próximos contribuem mais na definição do novo valor de nível de cinza do que os pixels mais afastados (Gonzalez, Bergasa et al. 2012). As principais abordagens para tratar esses tipos de problemas envolvem métodos no domínio espacial e no domínio da frequência. O termo domínio espacial refere-se ao agregado de pixels que compõem uma imagem, e métodos no domínio espacial são métodos que operam diretamente sobre estes pixels. O uso de máscaras espaciais para processamento de imagens é usualmente chamado filtragem espacial.

O fato de que as operações no domínio espacial são realizadas diretamente com os pixels da imagem é uma vantagem, pois a imagem não sofre transformações prévias e posteriores para poder ser processada, ao contrário do que ocorre com as operações realizadas no domínio frequência onde a imagem deve ser transformada do domínio espacial para o domínio frequência para poder ser tratada e, então, transformada novamente para o domínio espacial. Na prática, pequenas máscaras espaciais são mais frequentemente usadas do que a transformada de Fourier, devido a sua simplicidade de implementação. Entretanto, uma compreensão dos conceitos do domínio da frequência é essencial para a solução de problemas que não são facilmente tratáveis por técnicas espaciais.

Por tanto, neste trabalho usaremos apenas o domínio espacial, técnicas no domínio da frequência podem ser estudadas em (Gonzalez and Woods 2008).

Dentro da filtragem espacial, ainda existem as abordagens linear e não-linear. Em relação à abordagem linear, a forma como ela se dá é determinada pela máscara do filtro, que consiste de uma matriz de dimensões  $N \times N$ , cujos componentes são os pesos da combinação linear a ser feita. Por questões de simetria, geralmente, são utilizadas janelas quadradas com  $N$  ímpar. Além disso, normalmente, para maior eficiência computacional, os valores de  $N$  são pequenos.

Basicamente, o elemento central da matriz da máscara coincide com o pixel a ser modificado, sendo posicionado sucessivamente sobre cada pixel de interesse da imagem. Portanto, o processo pode ser entendido como uma máscara deslizante sobre os pontos de interesse da imagem, que assumirão um novo valor de acordo com a combinação linear dos pixels vizinhos ponderados pelos pesos  $w$  da matriz máscara.

Esse processo pode ser representado por uma operação de convolução da imagem  $I(m, n)$  com a máscara de pesos  $w$ , resultando na imagem filtrada  $I_{filtrada}(m, n)$ . A Equação 2-4 representa o processo em sua forma matemática.

$$I_{filtrada}(m, n) = I * w = \sum_{i=I_{min}}^{I_{max}} \sum_{j=J_{min}}^{J_{max}} w(i, j) I(m - i, n - j) \quad (\text{Equação 2-4})$$

Situação em que os índices  $i = 0$  e  $j = 0$  se referem ao pixel central da máscara, que possui dimensões  $(I_{max} - I_{min} + 1, J_{max} - J_{min} + 1)$ .

Em relação aos tipos, geralmente os filtros se dividem em três: passa-baixos, passa-faixa e passa-altos.

Filtros de suavização ou filtro passa-baixos são usados para borramento e redução de ruído. O borramento é utilizado em pré-processamento para remoção de pequenos detalhes de uma imagem antes da extração de objetos (grandes), e conexão de pequenas descontinuidades em linhas e curvas. A redução de ruídos pode ser conseguida pelo borramento com filtro linear assim como por filtragem não linear. Se o objetivo for alcançar a redução de ruído em vez de borrar, uma abordagem alternativa consiste no uso de filtros por mediana. Isto é, o nível de cinza de cada pixel

é substituído pela mediana dos níveis de cinza na vizinhança daquele pixel, ao invés da média (Gonzalez and Woods 2008). Neste trabalho é usado um filtro de mediana como primeira etapa de pré-processamento, antes da etapa de localização, com o objetivo de diminuir possíveis ruídos no documento.

Filtro passa-altos - atenuam ou eliminam os componentes de baixa-frequência, como esses componentes são responsáveis pelas características que variam lentamente em uma imagem, tais como o contraste total e a intensidade média, o efeito resultante da filtragem passa-altos é uma redução destas características, correspondendo a uma aparente intensificação bordas e outros detalhes finos.

Neste trabalho é aplicado um realce de contraste que é basicamente a aplicação de um filtro passa-altos à imagem. A seguinte matriz é o kernel para o filtro passa-altos comum usado para esta tarefa (Gonzalez and Woods 2008).

$$\begin{matrix} -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \\ \frac{1}{9} & 1 & -\frac{1}{9} \\ -\frac{1}{9} & -\frac{1}{9} & -\frac{1}{9} \end{matrix}$$

## 2.4 Limiarização

A limiarização, ou thresholding, é uma das mais importantes abordagens para a segmentação de imagens (Gonzalez and Woods 2008, Szeliski 2011), é uma técnica de análise por região particularmente útil para cenas que contêm objetos sobre um fundo contrastante. Sua implementação computacional geralmente é simples. Devido ao fato da limiarização produzir uma imagem binária na saída, o processo também é denominado, muitas vezes, binarização.

A conversão de uma imagem com níveis de cinza para uma imagem com representação binária (dois tons) é importante para uma série de objetivos, tais como identificar objetos e separá-los do fundo da imagem e analisar a forma da imagem quando é mais importante a forma que a intensidade dos pixels. A forma mais simples de limiarização consiste na bipartição do histograma, convertendo os pixels cujo tom de cinza é maior ou igual a um certo valor de limiar ( $T$ ) em brancos e os demais em pretos.

Matematicamente, a operação de limiarização pode ser descrita como uma técnica de processamento de imagens na qual uma imagem de entrada  $I_{cinza}(m, n)$  de N níveis de cinza produz na saída uma imagem  $I_B(m, n)$  chamada de imagem limiarizada, cujo número de níveis de cinza é menor que N. Normalmente,  $I_B(m, n)$  apresenta 2 níveis de cinza, sendo:

$$I_B(m, n) = \begin{cases} 0 & \text{if } I_{cinza}(m, n) < T \\ 1 & \text{caso contrário} \end{cases} \quad (\text{Equação 2-5})$$

No qual os pixels rotulados com 1 (um) correspondem aos objetos e os pixels etiquetados com 0 (zero) correspondem ao fundo (background) e T é um valor de tom de cinza predefinido, ao qual denomina-se limiar.

A limiarização pode ser vista como uma operação que envolve um teste com relação a uma função T do tipo  $T = T[m, n, p(m, n), f(m, n)]$ , onde  $f(m, n)$  é o tom de cinza original no ponto  $(m, n)$  e  $p(m, n)$  indica alguma propriedade local deste ponto, por exemplo, a média de seus vizinhos. Quando T depende apenas de  $f(m, n)$ , o limiar é chamado global; quando T depende de  $f(m, n)$  e de  $p(m, n)$ , o limiar é chamado local. Se, além disso, T depende das coordenadas espaciais de  $(m, n)$ , o limiar é chamado dinâmico ou adaptativo.

A Figura 2-5 representa as imagens resultantes das diferentes limiarizações da imagem Figura 2-2.

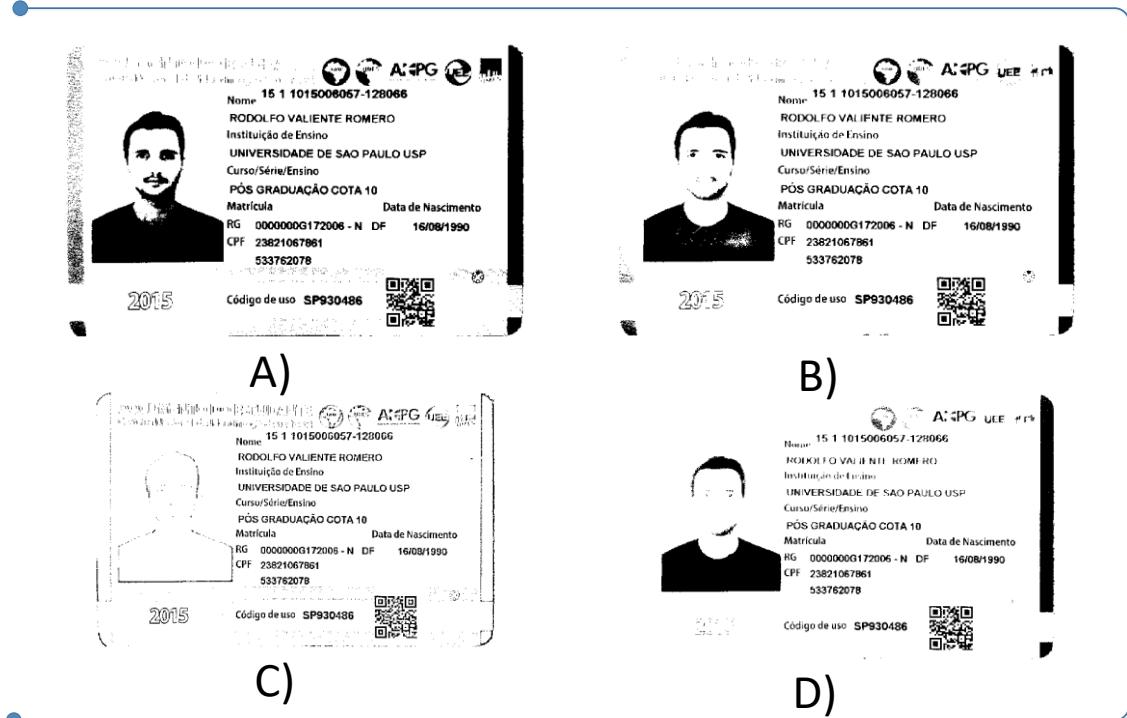


Figura 2-5: Limiarizações da Figura 2-2 (b), usando a) Otsu global, b) método adaptativo, c) método local, d) limiar selecionado a partir do histograma manualmente.  
Fonte: Autor.

O histograma da imagem, após sua binarização, terá apenas dois tons com número de pixels diferentes de zero. É evidente que a escolha adequada do valor de limiar é essencial para o bom funcionamento da técnica, e ainda, esta escolha é única para cada imagem. Nesta dissertação são usados os métodos de Otsu, local e adaptativo (como explicados no próximo epígrafe) como uma etapa de pré-processamento antes da aplicação do OCR.

Em muitos casos, principalmente quando não há um controle da iluminação sobre a imagem, o fundo não possui uma intensidade luminosa constante, e o contraste da imagem varia. Neste caso, um valor de limiar que fornece um bom resultado em uma determinada região pode não ser adequado em outra. Como a iluminação sobre o objeto não é homogênea, um limiar global não funciona bem, causando uma perda de informação. É necessário um limiar variável que se adapte às diferentes condições de iluminação.

### 2.4.1 Abordagem de limiarização global

Os métodos de limiarização global calculam um único valor de limiar  $T$  para todos os pixels da imagem. O método proposto por Otsu (Otsu 1975) está dentre os melhores métodos de limiarização global (Trier and Jain 1995), e é utilizado neste trabalho.

O algoritmo de Otsu propõe a segmentação da imagem em duas classes, em que o limiar ótimo  $T$  é aquele que minimiza a variância dentro da 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  $T$ .

Ele visa dividir uma imagem que possui  $L$  níveis de cinza em duas classes  $C_0$  e  $C_1$ , uma que contará com os pixels pretos, a outra, com os brancos (objeto e o fundo). Admitindo-se que essa divisão será estabelecida no nível de cinza  $t$ , tem-se que as duas classes são formadas pelos seguintes níveis de cinza:

$$C_0 = \{0, 1, 2, \dots, t\} \quad (\text{Equação 2-6})$$

$$C_1 = \{t + 1, t + 2, \dots, L\} \quad (\text{Equação 2-7})$$

Seja  $\sigma_w^2$  a variância dentro da classe,  $\sigma_B^2$  a variância entre as classes e  $\sigma_T^2$  a variância total. Um limiar ótimo pode ser obtido pela minimização de uma das funções critérios seguintes:

$$\zeta = \frac{\sigma_T^2}{\sigma_w^2} \quad \eta = \frac{\sigma_B^2}{\sigma_T^2} \quad \lambda = \frac{\sigma_B^2}{\sigma_w^2} \quad (\text{Equação 2-8})$$

Das três funções critérios apresentadas acima,  $\eta$  é a mais simples, o limiar ótimo  $t^*$  é definido por:  $t^* = \operatorname{Arg} \max \eta$

sendo que

$$\sigma_T^2 = \sum_{i=0}^{L-1} (i - \mu_T)^2 p_i \quad (\text{Equação 2-9})$$

$$\mu_T = \sum_{i=0}^{L-1} i p_i \quad (\text{Equação 2-10})$$

$$\omega_0 = \sum_{i=0}^t p_i \quad (\text{Equação 2-11})$$

$$\omega_1 = 1 - \omega_0 \quad (\text{Equação 2-12})$$

$$\mu_1 = \frac{\mu_T - \mu_0}{1 - \mu_0} \quad (\text{Equação 2-13})$$

$$\mu_0 = \frac{\mu_T}{\omega_0} \quad (\text{Equação 2-14})$$

$$\mu_T = \sum_{i=0}^t i p_i \quad (\text{Equação 2-15})$$

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \quad (\text{Equação 2-16})$$

$p_i$  é a probabilidade de um pixel qualquer da imagem pertencer ao nível de cinza  $i$ , como explicado anteriormente.

Em suma, a métrica  $\eta$  é calculada para todos os níveis possíveis da imagem, sendo que o  $t$  escolhido será aquele que a maximiza. Equivalentemente, é aquele que maximiza o valor de  $\sigma_B^2$ , que consiste da variância entre as duas classes. Apesar de envolver alguns cálculos, o método é simples, visto que são utilizados apenas os momentos cumulativos zero e de primeira ordem do histograma de níveis de cinza.

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, prejudicando o desempenho do método de Otsu. A Figura 2-5 (a) representa a imagem resultante da aplicação do método de Otsu.

#### 2.4.2 Abordagem de limiarização local

Os algoritmos de limiarização local atribuem um limiar  $T(m,n)$  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. Para coletar tais características, uma janela (ou máscara) deslizante  $W$  move-se extraiendo informações dos pixels sob a janela, tais como: média, variância, faixa

dinâmica, etc. Tais dados servem de suporte para a determinação de um limiar  $T(m,n)$  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.

O método de Niblack 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  $T(m,n)$  mediante a seguinte equação:

$$T(m,n) = \mu(m,n) + K \cdot s(m,n) \quad (\text{Equação 2-17})$$

No qual  $\mu(m,n)$  denota a média, e  $s(m,n)$ , 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 terminante no processo de binarização. Janelas subdimensionadas fazem com que os pixels de ruído influenciem na determinação dos valores de limiar, enquanto janelas superdimensionadas resultem em um limiar inadequado por não preservar os detalhes locais.

A equação possui valores fixos de  $K$  e  $W$  independentes da imagem de entrada. Como consequência, torna-se difícil encontrar valores para tais constantes que produzam resultados satisfatórios para diferentes imagens. Vários trabalhos usam o valor 0,6 para  $K$  e janela  $W$  de  $25 \times 25$ . Tais valores foram obtidos heuristicamente por He et al. (He, Do et al. 2005). A Figura 2-5 (c) representa a imagem resultante da aplicação do método de Niblack.

## 2.5 Operações morfológicas

“Operação morfológica” ou “morfologia matemática” é uma ferramenta de pré-processamento e para a extração de componentes de imagens que sejam úteis na representação e descrição da forma de uma região, como: fronteiras, esqueletos, etc (Gonzalez and Woods 2008).

A ideia base da morfologia é comparar os objetos que queremos analisar com um outro objeto de forma conhecida chamado elemento estruturante. A partir desse elemento estruturante, é possível testar e quantificar de que maneira o elemento

estruturante “está ou não contido” na imagem. Cada elemento estruturante fornece uma aparência nova do objeto, donde surge a importância de sua escolha.

A seguir são destacadas algumas operações morfológicas:

Dilatação - expande uma imagem, ou seja, os efeitos da dilatação são engordar as partículas, preencher os pequenos buracos. A dilatação de uma imagem A por um elemento estruturante B é definida através da equação:

$$\delta_B(A) := A \oplus B = \cup \{A + b : b \in B\} \quad (\text{Equação 2-18})$$

Erosão - reduz uma imagem, ou seja, aumenta os espaços entre os caracteres e consequentemente evita as sombras. O efeito da erosão é fazer desaparecer os elementos de tamanho inferior ao tamanho do elemento estruturante. A erosão do conjunto A pelo conjunto B, denotado por  $A \ominus B$ , é definida através da seguinte equação:

$$\varepsilon_B(A) := A \ominus B = \{x : B + x \subseteq A\} = \{x : B_x \subseteq A\} \quad (\text{Equação 2-19})$$

A abertura da imagem A por um elemento estruturante B, denotado  $A \circ B$ , é definida como a erosão de A por B seguida da dilatação por B. Precisamente, tem-se:

$$\gamma_B(A) := A \circ B = (A \ominus B) \oplus B \quad (\text{Equação 2-20})$$

O fechamento de A por B, denotado por  $A \bullet B$ , é definido como segue:

$$\varphi_B(A) := A \bullet B = (A \oplus B) \ominus B \quad (\text{Equação 2-21})$$

Um maior estudo das operações morfológicas é realizado em (Dougherty and Lotufo 2003) e (Gonzalez and Woods 2008).

Neste trabalho na etapa de pré-processamento antes do OCR é usada morfologia para a separação de alguns caracteres emendados, usando abertura da imagem com elemento estruturante 3x3. Os caracteres quebrados são tratados usando fechamento com o mesmo elemento estruturante.

## 2.6 Detector de bordas

Entre as técnicas utilizadas para segmentação de imagens as abordagens para detecção de bordas possuem um importante papel e são amplamente usadas. Primeiramente, entende-se borda como o limite ou a fronteira entre duas regiões com propriedades distintas de nível de cinza. De forma geral, a maioria das técnicas de detecção de bordas utiliza o cálculo de um operador diferencial. Partindo desse ponto,

pode-se utilizar a magnitude da primeira derivada para a detecção de uma borda em uma imagem. Por sua vez, o sinal da derivada segunda possui um cruzamento no zero, ou seja, indica que há transição dos níveis de cinza, o que permite a localização das bordas na imagem. Pontua-se que a primeira derivada em qualquer ponto da imagem é obtida a partir da magnitude do gradiente naquele ponto. Já a segunda derivada é obtida da mesma forma, mas a partir do operador Laplaciano.

Em termos contínuos, o gradiente de  $f(x,y)$  em um certo ponto  $(x,y)$  é definido como o vetor:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (\text{Equação 2-22})$$

E a força da borda é marcada pela magnitude do gradiente dada por:

$$\nabla f = \text{mag}(\nabla f) = \sqrt{\frac{\partial f^2}{\partial x} + \frac{\partial f^2}{\partial y}} \quad (\text{Equação 2-23})$$

Existem vários métodos de detecção de bordas, entre os quais estão: Método de Sobel, Prewitt, Canny, etc. Dentre eles, uns dos mais usados é o detector de borda Canny (Mohamed, Mahmoud et al. 2017), que pelas suas vantagens é o detector usado neste trabalho. O algoritmo é brevemente descrito a seguir, mais detalhes podem ser encontrados em (Canny 1986).

1. Suavizar a imagem usando um filtro Gaussiano para reduzir ruídos.
2. Computar o gradiente  $g(x,y)$  e a direção do gradiente em cada ponto. Os pontos de bordas calculados originam cristas na imagem de magnitudes do gradiente.
3. Supressão de Não-Máximos.
4. Estabelecer 2 limiares  $T_1$ ,  $T_2$ . Valores maiores que  $T_2$  são considerados Bordas Fortes e valores entre  $T_1$  e  $T_2$  são Bordas Fracas.
5. Incorporar às Bordas Fortes as Bordas Fracas que sejam 8-conectadas.

Os resultados da detecção de borda usando detector Sobel e Canny são mostrados a seguir.

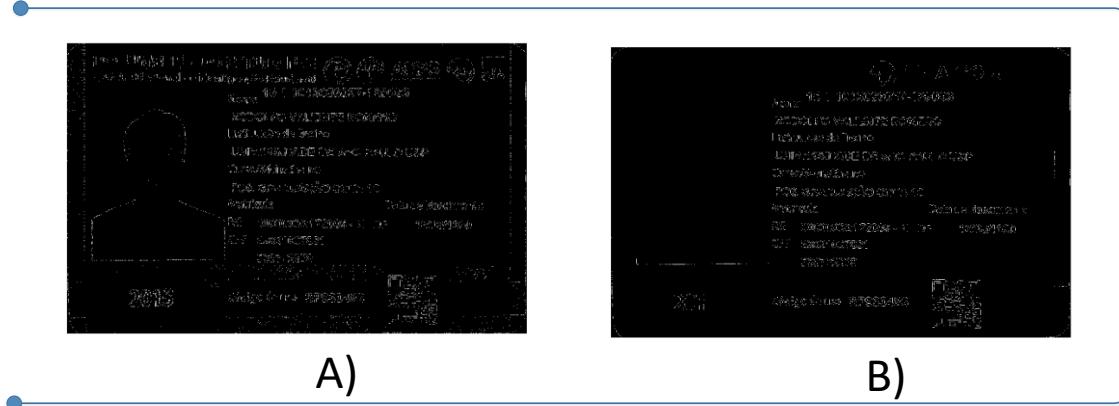


Figura 2-6 Resultado dos algoritmos de detecção de borda: a) detector Sobel, b) detector Canny. Fonte: Autor.

## 2.7 Transformada de Hough

Além da detecção de bordas, neste trabalho, a detecção de um conjunto de pontos em uma imagem que pertencem a um segmento de reta também é importante. Dito isso, o problema relacionado ao segmento de reta consiste basicamente em achar subconjuntos de pontos que sejam colineares. Uma solução possível, mas inviável computacionalmente para a maioria das aplicações, é encontrar todos os segmentos de retas formados entre cada par de pontos e procurar pelos conjuntos de pontos que estejam próximos desses segmentos. Para facilitar o processo, Hough (Hough 1962, Duda and Hart 1972) propôs um método, denominado transformada de Hough que é detalhado na sequência.

Uma reta pode ser representada pela equação  $y = mx + b$

Para diferentes valores de  $m$  e  $b$ , há infinitas retas que passam por um ponto  $p_1(x_1, y_1)$ , todas elas satisfazendo a equação  $y_1 = mx_1 + b$ , bem como há infinitas retas que passam por um ponto  $p_2(x_2, y_2)$ , todas elas satisfazendo a equação  $y_2 = mx_2 + b$ .

Reescrevendo a equação , tem-se que  $b = y - mx$

Assim como o plano  $xy$  é denominado de plano da imagem, o plano  $mb$  é chamado de espaço de parâmetros. Sendo assim, todas retas que passam pelo ponto  $p_1$  são representadas no espaço de parâmetros pela equação  $b = y_1 - mx_1$ . De forma análoga, a equação  $b = y_2 - mx_2$  representa todas as retas que passam pelo ponto  $p_2$  no plano da imagem.

É interessante observar que o ponto  $(m, b)$ , localizado no espaço de parâmetros, é comum a essas duas retas associadas aos pontos  $p_1$  e  $p_2$ . Mais interessante ainda é notar que todos os pontos que são colineares no plano da imagem se interceptam em um mesmo ponto no espaço de parâmetros. A Figura 2-7 ilustra essa afirmação.

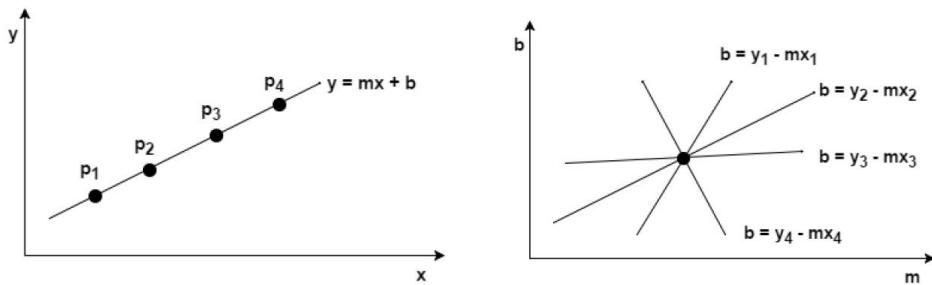
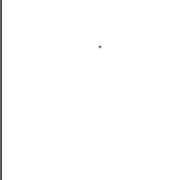
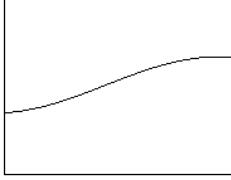


Figura 2-7 Os quatro pontos colineares em (a) são mapeados em quatro retas que se cruzam no mesmo ponto no espaço de parâmetros em (b). Fonte: (Gonzalez and Woods 2008).

Esse entendimento é a base para a transformada de Hough que, para melhor representação, utiliza a equação de uma reta em sua forma polar. Ou seja  $\rho = x \cdot \cos\theta + y \cdot \sin\theta$

Portanto, agora, passa-se do espaço  $(m, b)$  para o espaço  $(\rho, \theta)$ , denominado espaço de Hough. Isto associa cada reta da imagem a um único ponto  $(\theta, \rho)$  no espaço de Hough. Infinitas retas passam por um ponto no plano. Todas as retas que passam por esse ponto formam um senóide no plano de Hough. Mais do que isso, pontos colineares no espaço  $(x, y)$  correspondem a curvas senoidais que se interceptam no espaço de Hough. Na Figura 2-8 mostram-se diferentes transformações.

Imagen original	Espaço de Hough
 (a) 1 ponto	 (b) 1 senóide

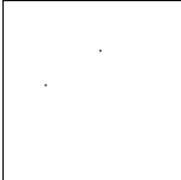
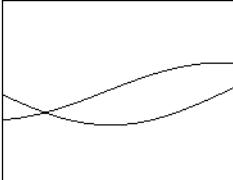
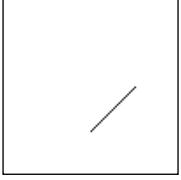
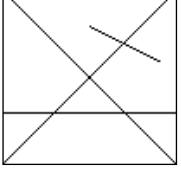
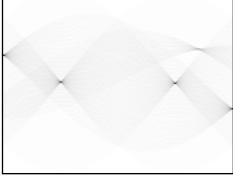
 (c) 2 pontos	 (d) 2 senóides. <p>O ponto de intersecção representa a reta que passa pelos 2 pontos.</p>
 (e) uma reta	 (f) Infinitos senóides
 (g) 4 retas	 (h) Infinitos senóides que se acumulam em 4 pontos

Figura 2-8 Diferentes transformações do espaço da imagem para o espaço de Hough. Fonte: (*Gonzalez and Woods 2008*).

Feita essa contextualização, a transformada Hough pode ser implementada por meio dos seguintes passos:

- 1) O espaço Hough  $(\rho, \theta)$  é discretizado em intervalos finitos, criando para cada uma das células resultantes um acumulador  $Ac(\rho, \theta)$ ;
- 2) Todas as células do acumulador  $Ac(\rho, \theta)$  são iniciadas com o valor zero;
- 3) Para cada ponto  $(x, y)$  no plano da imagem, calcula-se os valores  $\rho$  e  $\theta$

4) Encontrados os valores, incrementa-se de uma unidade o acumulador  $Ac(\rho, \theta)$ ;

5) Uma vez determinados os parâmetros de todos os pontos do plano da imagem, as células com os maiores valores do acumulador  $Ac(\rho, \theta)$  indicam potenciais retas na imagem.

Uma das limitações do método é determinar, dentre os maiores valores do acumulador, aqueles que são relevantes, ou seja, deve-se definir um limiar acima do qual uma célula será considerada como tendo os parâmetros de um segmento de reta na imagem. Neste trabalho a transformada de Hough é usada após uma limiarização e abertura morfológica para detectar grupos de pixels que pertencem a uma linha reta, o que serve para depois corrigir automaticamente a rotação de um documento de identificação.

### 2.7.1 Rotação

Caso o documento tenha sido mal posicionado, a imagem gerada poderá sofrer uma inclinação em seu eixo. Essa inclinação pode gerar falhas na etapa de reconhecimento. A rotação da imagem de um documento pode ser tratada principalmente pelo uso da Transformada de Hough anteriormente explicada (Matas, Galambos et al. 2000, Yu-peng Gao 2011).

Uma imagem pode ser rotacionada de um ângulo arbitrário, tanto no sentido horário quanto no anti-horário. Rotações com ângulos múltiplos de  $90^\circ$  são mais simples de implementar, pois consistem na cópia de pixels que estão organizados em linhas, reordenando-os em colunas na direção em que se deseja rotar a imagem. A rotação por ângulos quaisquer é uma tarefa mais complexa. Matematicamente, a rotação de cada ponto  $(X, Y)$  de uma imagem por um ângulo arbitrário  $\theta$ , mapeará este ponto na localidade de coordenadas  $(X', Y')$ , onde  $X'$  e  $Y'$  são calculados pelas equações:

$$X' = X \cos(\theta) + Y \sin(\theta) \quad (\text{Equação 2-24})$$

$$Y' = Y \cos(\theta) - X \sin(\theta) \quad (\text{Equação 2-25})$$

Na Figura 2-9 mostra-se a correção de rotação de um documento, em b) observa-se o resultado após uma limiarização , em c) o resultado da abertura morfológica da imagem limiarizada, em d) o espaço de Hough da imagem, em e) a

imagem com as linhas detectadas, e finalmente a imagem corrigida e corretamente rotacionada.

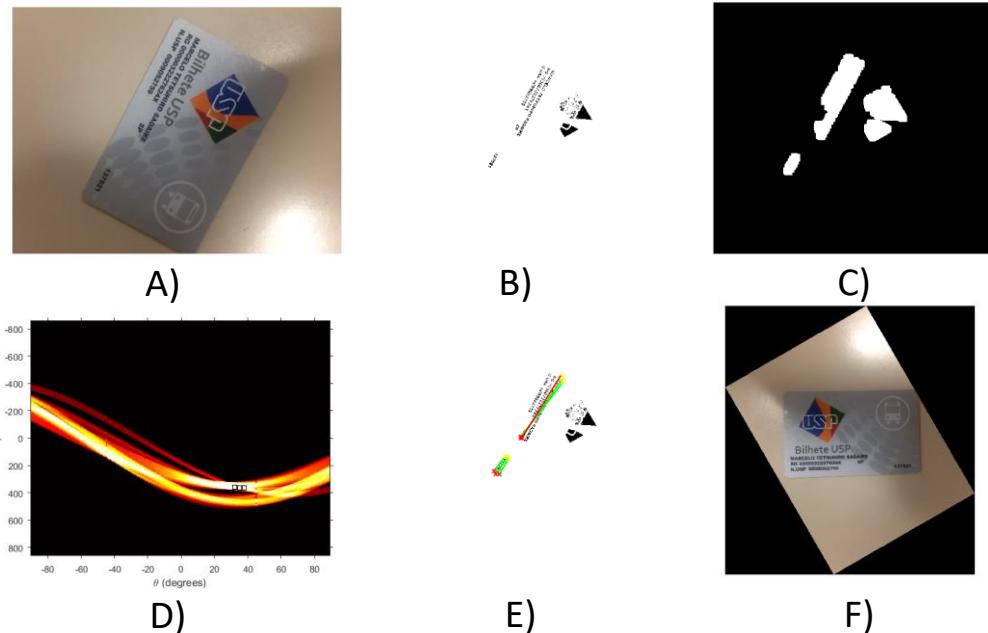


Figura 2-9: Correção de rotação da imagem usando a transformada de Hough.  
Fonte: Autor.

## 2.8 Largura do traçado (Stroke width)

Um traço na imagem é uma banda contínua de largura quase constante. O Stroke Width Transform (SWT) é um operador local que calcula para cada pixel a largura do traço onde ele está contido. Um exemplo de traço é mostrado na Figura 2-10 (a), os pixels do traço neste exemplo são mais escuros do que os pixels de fundo.

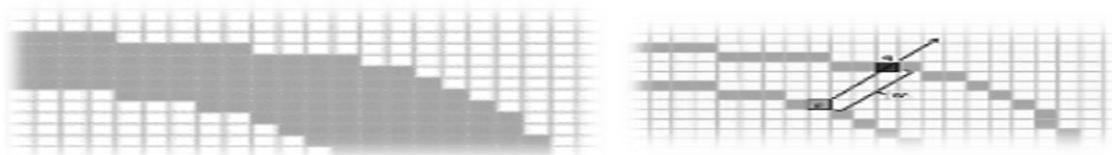


Figura 2-10 a) Exemplo de traço e a sua largura. Fonte: (Epshtain, Ofek et al. 2010).

No cálculo da largura do traçado é usada a transformada da distância. Em processamento de imagem a distância caracteriza a separação entre dois objetos, é uma métrica que deve satisfazer os seguintes requisitos:

Uma métrica em  $M$  é uma função  $d: M \times M \rightarrow \mathbb{R}$  tal que para quaisquer  $x, y, z \in M$  tenhamos:

$$1. \quad d(x, y) \geq 0 \text{ e } d(x, y) = 0 \text{ se e só se } x = y; \quad (\text{Equação 2-26})$$

$$2. \quad d(x, y) = d(y, x); \quad (\text{Equação 2-27})$$

$$3. \quad d(x, z) \leq d(x, y) + d(y, z). \quad (\text{Equação 2-28})$$

Qualquer função que satisfizer estas três propriedades servirá para “medir” a distância entre pontos de um conjunto e tais funções serão chamados métricas. Algumas métricas usuais para  $d(x, y)$  são:

$$\text{City-block: } d_4(x, y) = |x_1 - y_1| + |x_2 - y_2| \quad (\text{Equação 2-29})$$

$$\text{Chessboard: } d_8(x, y) = \max\{|x_1 - y_1|, |x_2 - y_2|\} \quad (\text{Equação 2-30})$$

$$\text{Euclidiana: } d_E(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad (\text{Equação 2-31})$$

Então, a função distância de um pixel  $x$  ao conjunto  $X$  pode ser definida como:  
 $d(x, X) = \min\{d(x, y), y \in X\}$  (Dougherty and Lotufo 2003)

A transformada distância é um Operador que atribui a cada ponto  $x$  de um objeto, a menor distância de  $x$  ao complemento do objeto. A função da transformada distância é definida como:

$$\Psi d(f)(x) = d(x, \{y \in E : f(y) = 0\}) \quad (\text{Dougherty and Lotufo 2003})$$

Onde  $E \subset \mathbb{Z} \times \mathbb{Z}$  é o domínio das imagens

Esta função atribui a cada pixel de um objeto numa imagem binária o valor da distância mínima ao fundo. Em outras palavras, atribui a cada pixel de um objeto a menor distância entre este pixel e um pixel de fundo.

O algoritmo da transformada da distância usado neste trabalho é descrito a seguir:

---

**Algoritmo** Transformada da distância
 

---

Entrada: Imagem binária  $f$ , conectividade  $G$ ,  $p, q \in E \subset \mathbb{Z} \times \mathbb{Z}$

Saída: Imagem níveis de cinza  $g = \Psi d(f)(p)$

---

Percorra  $E$ , em sentido raster,  $\forall p \in E$

se  $f(p) = 1, f(p) = 1 + \min\{f(q) : q \in N_G^-(p)\}$

Percorra  $E$ , em sentido anti-raster  $\forall p \in E$

se  $f(p) \neq 0, f(p) = \min\{f(p), (1 + \min\{f(q) : q \in N_G^+(p)\})\}$

---

$g = f$

---

Usando a transformada da distância e o esqueleto da imagem como definidos em (Dougherty and Lotufo 2003), podemos calcular a largura do traçado para cada componente conexo. O SWT pode detectar caracteres de diferentes idiomas (inglês, hebraico, árabe, etc.), o texto pode ser de tamanhos variados, estilos e cores, pode ser de orientação diferente, perspectiva e rotação, incluindo texto *curvy*, até mesmo a escrita manual pode ser detectada. Essa técnica junto com a informação da variação da largura é usada nesta dissertação para selecionar os candidatos a texto.

## 2.9 MSER (Maximally Stable Extremal Regions)

As regiões extremas maximamente estáveis (MSER) são usadas como um método de detecção de componentes em imagens. Esta técnica foi proposta por Matas (Matas, Chum et al. 2004) inicialmente para encontrar correspondências entre elementos de duas imagens com diferentes pontos de vista, e hoje é amplamente usada na localização de texto em imagens.

O algoritmo MSER extrai de uma imagem um número de regiões covariantes, chamadas MSER: O MSER baseia-se na ideia de tomar regiões que ficam quase iguais através de uma ampla gama de limiares. Esta operação pode ser realizada seguindo as seguintes etapas:

- 1) Considerar uma sequência de limiares (fazer todos os pixels abaixo de um limite branco, os outros pretos) com valores cada vez maiores que varrem de branco para preto, passamos de uma imagem branca para imagens onde regiões pretas aparecem e crescem unindo-se, até a imagem final.
- 2) Extrair os componentes conectados ("Regiões Extremas")

- 3) Encontrar os limites quando uma região extrema é "Maximamente Estável" (a região abaixo / acima coincidem com a região atual em certo grau)
- 4) Manter essas regiões descritas como características, sobre uma ampla gama de limiares essas regiões são estáveis e mostra alguma invariância para diferentes transformações.

O algoritmo MSER definido por (Matas, Chum et al. 2004) e usado nesta dissertação é descrito a seguir:

Seja a imagem  $I$  um mapeamento  $I : D \subset \mathbb{Z}^2 \rightarrow S$ . As regiões extremas estão bem definidas nas imagens se:

- 1)  $S$  está totalmente ordenado
- 2) Uma relação de adjacência (vizinhança)  $A \subset D \times D$  é definida

A região  $Q$  é um subconjunto contíguo de  $D$

Limite de Região  $\partial Q = \{q \in D \setminus Q : \exists p \in Q : qAp\}$  o limite  $\partial Q$  de  $Q$  é o conjunto de pixels que são adjacentes a pelo menos um pixel de  $Q$  mas não pertencem a  $Q$ .

Região Extrema  $Q \subset D$  é uma região que  $\forall p \in Q, q \in \partial Q : I(p) > I(q)$  (região de intensidade máxima) ou  $I(p) < I(q)$  (região de intensidade mínima).

Regiões extremas maximamente estáveis (MSER). Seja  $Q_1, \dots, Q_{i-1}, Q_i, \dots$  uma sequência de regiões extremas aninhadas ( $Q_i \subset Q_{i+1}$ ). Região extrema  $Q_i$  é maximamente estável se e somente se:

$$q(i) = |Q_{i+\Delta} \setminus Q_{i-\Delta}| / |Q_i| \quad (\text{Equação 2-32})$$

Tem um mínimo local em  $i^*$  (onde  $|...|$  denota cardinalidade).  $\Delta \in S$  é um parâmetro do método,  $i \pm \Delta$  a região no  $\Delta$ -ésimo limiar inferior ou superior. Os MSER são identificados pelo mínimo local de  $q$ .

A equação verifica as regiões que permanecem estáveis em um certo número de limiares. Se uma região  $Q_{i+\Delta}$  não é significativamente maior do que a região  $Q_{i-\Delta}$  a região  $Q_i$  é considerada como uma região de máxima estabilidade.

O algoritmo MSER é aplicado na imagem de um documento mostrando-se os resultados das regiões detectadas na Figura 2-11



Figura 2-11: Regiões MSER. Fonte: Autor.

O conceito mais simples pode ser explicado usando limiar. Todos os pixels abaixo de um determinado limite são "pretos" e todos aqueles acima ou iguais são "brancos". Dada uma imagem de origem, se gerarmos uma sequência de imagens de resultado com limiares, onde cada imagem  $I_t$  corresponde a um limiar  $t$ , veríamos primeiro uma imagem branca, então aparecerão manchas "pretas" correspondentes aos mínimos de intensidade local, depois aumentarão. Esses pontos "negros" acabarão por se fundir, até que toda a imagem seja preta. O conjunto de todos os componentes conectados na sequência é o conjunto de todas as regiões extremas. A sequência deste processo na região selecionada da Figura 2-11 (região com quadrado em vermelho) é apresentada na Figura 2-12.

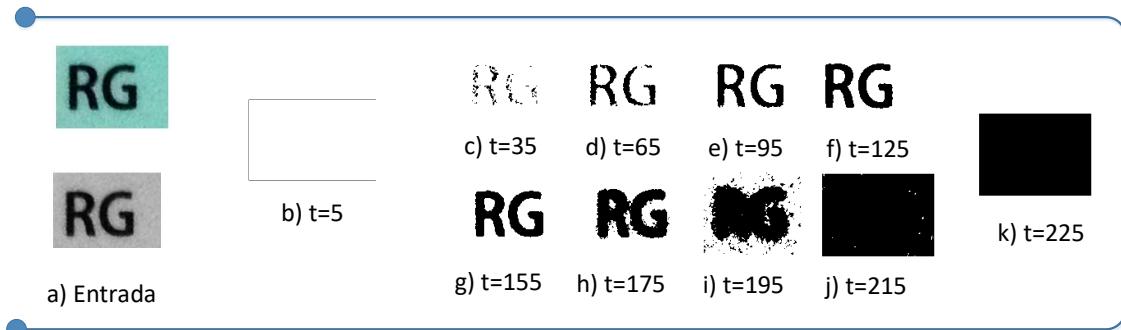


Figura 2-12: Sequencia dos limiares aplicados a uma imagem para calcular as regiões MSER. Fonte: Autor.

Nesse sentido, o conceito de MSER está vinculado ao de uma árvore de componente da imagem. A árvore de componentes fornece uma maneira fácil de implementar o MSER como observa-se na Figura 2-13.

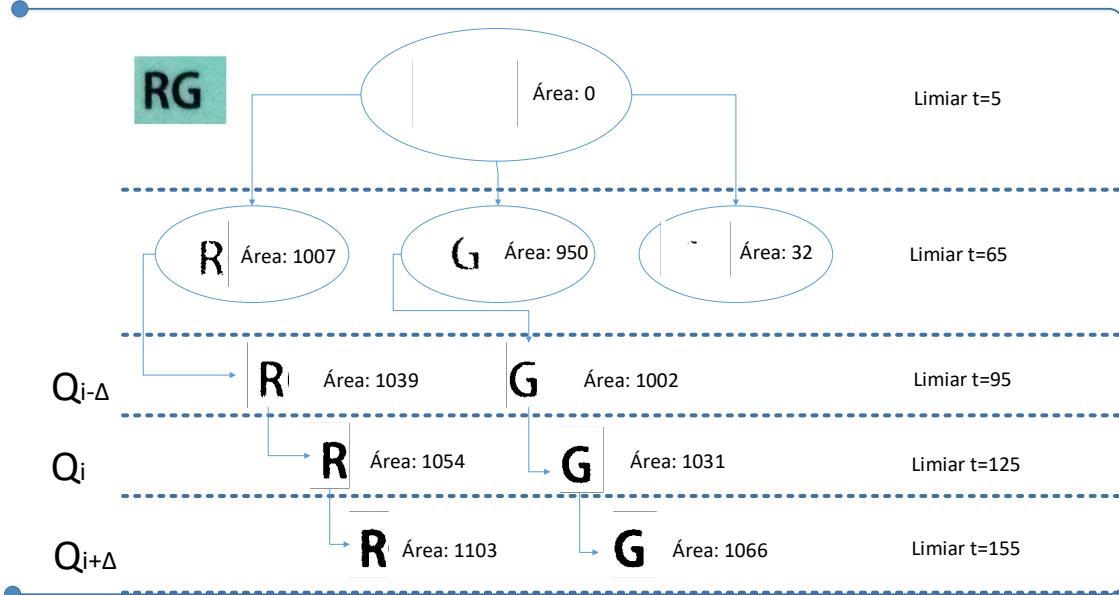


Figura 2-13 : Árvore de componentes da imagem para o algoritmo MSER. Fonte: Autor.

O MSER é capaz de localizar com precisão texto em diferentes tamanhos, estilos e cores, independente de perspectiva e rotação, portanto é amplamente usado na localização de texto. Chen combina o MSER com as bordas de Canny para detecção de texto, as bordas simples são usadas para ajudar a lidar com a fraqueza do MSER para desfocar. O MSER é aplicado pela primeira vez à imagem em questão para determinar as regiões de caracteres. Para melhorar as regiões do MSER, todos os pixels fora dos limites formados pelas bordas de Canny são removidos (Chen, Tsai et al. 2011). Uma utilização alternativa do MSER na detecção de texto é o trabalho de Shi usando um modelo de grafos. Este método aplica novamente o MSER à imagem para gerar regiões preliminares. Estes são usados para construir um modelo de grafos com base na distância da posição e distância de cor entre cada MSER, que é tratado como um nó. Em seguida, os nós são separados em plano principal e plano de fundo usando funções de custo. Uma função de custo usa a distância do nó com o plano principal e o plano de fundo. A outra penaliza os nós por serem significativamente diferente do seu vizinho. Quando estes são minimizados, o gráfico é cortado para separar os nós de texto dos nós que não são de texto (Shi, Wang et al. 2013). Para permitir a detecção de texto em uma cena geral, Neumann usa o algoritmo MSER em uma variedade de projeções. Além da projeção de intensidade em escala de cinza, ele usa os canais de cor vermelho, azul e verde para detectar regiões de texto que

são cores distintas, mas não necessariamente distintas em intensidade de escala de cinza (Neumann and Matas 2010, Lukas Neumann 2015).

A localização do texto pode ser realizada usando operadores morfológicos (Alves and Hashimoto 2010), aprendizado de máquina (Luccheseyz and Mitray 2001, Gonçalves, da Silva et al. 2016) e outros. Recentemente tem sido amplamente usado algoritmos como o MSER, com excelentes resultados (Gonzalez, Bergasa et al. 2012, Neumann, Matas et al. 2012, Yin, Yin et al. 2014). Pela sua demostrada eficiência o MSER é usado nesta dissertação para localizar os caracteres na imagem.

## **2.10 Reconhecimento ótico de caracteres - Tesseract**

Uma vez definidos os algoritmos de processamento de imagem, eles são usados para processar e analisar as imagens antes da etapa de reconhecimento onde são usados sistemas de OCR. A ideia principal do OCR é o reconhecimento de um caractere em uma determinada imagem digitalizada. Para isso, as características desse caractere são comparadas com características de padrões de um determinado alfabeto.

Após o reconhecimento de cada um dos caracteres, deve-se fazer a associação entre eles com a finalidade de se ter uma sequência de caracteres que tenha algum significado, ou seja, eles devem ser agrupados de modo a formar palavras. Além disso, é importante ressaltar que se deve considerar a ocorrência de erros na etapa de classificação. Sendo assim, no pós-processamento pode ser realizado uma atividade de detecção e correção dos erros. Por fim, torna-se necessário que se tenha uma interface de comunicação entre o sistema e o mundo externo a ele com o intuito de expor o resultado. Há sistemas que formatam sua saída em documentos de texto, tabelas, banco de dados, etc. Há também sistemas que expõem suas saídas a outros sistemas, fazendo parte de um sistema automatizado.

## **Capítulo 3**

### **3 Trabalhos relacionados. Sistemas de extração de texto mais relevantes**

Nos últimos anos, vários têm sido os sistemas apresentados com o intuito de resolver o problema da extração automática de texto em imagens de documentos. Exemplos destes sistemas podem ser encontrados em (Messelodi and Modena 1999, da Conceição Palma 2004, Tahim 2010, Minetto 2012, Peanho, Stagni et al. 2012, Rossi 2016). No caso de documentos de identificação duas abordagens são usadas: com modelo, e sem modelo aproveitando os algoritmos existentes para a detecção de texto (Sharma and Fujii , Sonia Bhaskar 2011, de las Heras, Terrades et al. 2015, Ryan and Hanafiah 2015, Simon, Rodner et al. 2015).

Estes sistemas de extração da informação textual geralmente são divididos em quatro subsistemas ou etapas: (i) localização das regiões candidatas a serem texto; (ii) seleção das regiões que possuem realmente caracteres; (iii) extração e correção do texto selecionado; (iv) reconhecimento do texto, como foi apresentado no diagrama de blocos da Figura 1-6. Tendo em conta que os trabalhos mais relevantes apresentam uma arquitetura similar, nas secções seguintes, serão apresentadas as soluções propostas pelos diferentes autores em cada etapa. Finalmente, são apresentados os trabalhos de reconhecimento de texto em documentos de identificação (Sonia Bhaskar 2011) e (Ryan and Hanafiah 2015).

#### **3.1 Etapas fundamentais no processo de reconhecimento de texto.**

Nesta seção apresenta-se uma descrição das etapas dos sistemas de extração e reconhecimento de texto explicando as abordagens usadas pelos diferentes autores. (Jain and Yu 1998, Messelodi and Modena 1999, da Conceição Palma 2004, Epshteyn, Ofek et al. 2010, Tahim 2010, Minetto 2012, Yin, Pei et al. 2015, Zhang, Shen et al. 2015, Zhao, Fang et al. 2015, Zhu, Wang et al. 2015, Zhu, Wang et al. 2015, Huang, He et al. 2016, Jaderberg, Simonyan et al. 2016, Rossi 2016). No apêndice (página 150), listam-se as características textuais comumente utilizadas nos sistemas de extração de texto.

### 3.1.1 Etapa de localização

Existem diversas abordagens para o problema de localização de textos em imagens. Basicamente, elas podem ser divididas em dois grupos: métodos que encontram os caracteres partindo da premissa de que os pixels que o formam possuem características semelhantes, usa-se processamento de imagens (sem modelo) e métodos baseados em busca por padrões conhecidos a priori (com modelo) (Brunelli 2009, Peanho, Stagni et al. 2012, Lukas Neumann 2015, Ryan and Hanafiah 2015, Islam, Mondal et al. 2016, Jaderberg, Simonyan et al. 2016). No apêndice (página 150) é efetuada uma descrição mais detalhada das abordagens usadas na etapa de localização. Os algoritmos de localização baseados em processamento de imagens serão apresentados a seguir, estes utilizam características extraídas da imagem para determinar as possíveis regiões textuais. Dependendo dos objetivos um grande número de técnicas para segmentação automática tem sido proposto na literatura. Podemos classificá-las em técnicas espaciais e técnicas temporais. Como nosso trabalho foca-se em imagens de documentos e não sequências de vídeo, são estudadas e usadas as técnicas de segmentação espacial.

Na segmentação espacial as regiões pretendidas deverão ser homogêneas em termos das suas características espaciais. Vários tipos de segmentação espacial podem ser considerados dependendo da aplicação. Assim, as técnicas de segmentação espacial podem ser divididas nas seguintes classes:

- Baseadas em regiões: Técnicas que detectam regiões homogêneas na imagem, separadas por fronteiras bem definidas.
  - Baseadas em componentes conexos (CCs): Técnicas simples que identificam as várias regiões conexas com base na análise dos histogramas e a conectividade;
  - Baseadas em bordas: Técnicas que detectam primeiramente as fronteiras existentes na imagem e depois processam o resultado de forma a identificar as várias regiões;
- Baseadas na textura: Técnicas que segmentam as regiões com base nas suas características em termos de textura.

A seguir são apresentados os trabalhos relacionados que usam as diferentes técnicas de segmentação espacial, exibindo as suas vantagens e desvantagens (Jain and Yu 1998, Messelodi and Modena 1999, Chen, Bourlard et al. 2001, Yangxing and IKENAGA 2006, Epshtein, Ofek et al. 2010, Yin, Pei et al. 2015, Zhang, Shen et al. 2015, Zhao, Fang et al. 2015, Zhu, Wang et al. 2015, Zhu, Wang et al. 2015, Huang, He et al. 2016, Jaderberg, Simonyan et al. 2016, Zhu and Zanibbi 2016).

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.

Jain e Yu (Jain and Yu 1998) 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. Após a redução do número de cores, uma clusterização hierárquica *single-link* é realizada sobre a imagem com número reduzido de cores. A partir das cores, 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 perceptivelmente diferentes. 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 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.

A técnica proposta por Jain e Yu possui baixa complexidade computacional, 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 consequência, 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. O autor enfatiza que o método é eficiente somente 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 (Messelodi and Modena 1999), 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. O passo seguinte é a criação de duas imagens binárias mediante o uso de dois limiares globais ( $I_1$  e  $I_2$ ) aplicados sobre a imagem de intensidade normalizada. 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 do método proposto por Messelodi e Modena possuir um sistema de localização capaz de identificar caracteres em diferentes orientações, os próprios autores sinalizam que a seleção dos filtros e seus limiares são altamente dependentes da aplicação e dimensões da imagem de entrada.

Os métodos baseados em detecção de fronteiras apresentam como principal vantagem um custo computacional razoável associado aos detectores de fronteira. Como ponto fraco, há que referenciar a sua sensibilidade ao ruído, especialmente quando se usam janelas muito pequenas como máscaras/filtros de detecção. O fato destas técnicas se basearem unicamente na informação espacial leva a que possam produzir um número elevado de pequenas regiões, sobretudo para imagens com muita textura.

Para o caso de texto, diversas características textuais são utilizadas para sua localização nas imagens, no entanto, os métodos baseados em bordas 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 anteriormente visto (Gonzalez and Woods 2008) sobre a imagem. Posteriormente

empregam filtros heurísticos para seleção das bordas textuais e as fundem por meio de operadores morfológicos ou de suavização para a delimitação das regiões textuais.

Chen et al. (Chen, Bourlard et al. 2001) geram inicialmente duas imagens de bordas, uma de bordas verticais e outra horizontais, por meio da detecção Canny. As bordas textuais geralmente possuem coesão espacial, portanto a densidade de bordas na direção vertical e horizontal em regiões de texto é superior à do plano de fundo. Baseando-se nessa característica, a operação morfológica de dilatação 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  $1 \times 5$ , visando fundir as bordas verticais na direção horizontal. A imagem de bordas horizontais é dilatada por meio de um elemento estruturante  $6 \times 3$ , 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. 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. (Yangxing and IKENAGA 2006) baseiam-se em um modelo híbrido, em que a localização é realizada mediante a identificação de bordas da imagem e a seleção 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. Um limiar adaptativo seleciona os contornos mais evidentes. Em seguida, tais contornos são filtrados por meio das suas características estruturais e texturais. Os contornos remanescentes são então caracterizados como textuais e delimitados por BBs.

O método de localização proposto por André Pires (Tahim 2010), é baseado no trabalho de Liu et al., 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. (Yangxing and IKENAGA 2006), a detecção de bordas do método proposto por André Pires, 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 ao método de Liu et al. no processo de detecção de bordas, uma vez que este último utiliza a detecção por campos vetoriais aplicado diretamente à imagem colorida. O método identifica em uma única varredura as possíveis regiões textuais, eliminando o processo de busca exaustiva mediante técnicas de multi-resolução.

A principal limitação desse método 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. O método limita-se a reconhecer caracteres que estão no foco na imagem. Regiões desfocadas da imagem possuem a energia concentrada nos componentes de baixa frequência. 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.

Os métodos baseados em textura apresentam como grande vantagem a sua capacidade para detectar homogeneidades mais sofisticadas, ainda que as regiões da imagem possuam texturas com elevada variedade; esta capacidade não existe associada a outros tipos de técnicas.

Para o caso do texto, a característica mais intuitiva é a sua regularidade. 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 (Sin, Kim et al. 2002) ao considerar que as regiões textuais possuem um certo tipo de textura cujos componentes freqüências 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 de frequências pode ser utilizada para segmentar e identificar regiões de texto em uma imagem, tais como: filtros de Gabor (Jain and Bhattacharjee 1992), *Wavelet* (Saoi, Goto et al. 2005), *Fast Fourier transform* (FFT) (Sin, Kim et al. 2002) e variância espacial (Zhong, Karu et al. 1995).

A maior desvantagem dos métodos baseados em textura é a complexidade computacional envolvida no estágio de classificação textural, que é superior à 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.

### **3.1.1.1 Abordagens para a localização usando correspondência de modelos**

A abordagem que utiliza correspondência de modelos, é custosa computacionalmente, visto que será necessária uma janela deslizante para percorrer toda a imagem. Mais do que isso, caso se queira robustez à rotação e variação de escala, objetivos deste trabalho, é necessário o processamento para diferentes escalas e, dependendo da situação, para diferentes ângulos de rotação, o que torna a abordagem ainda mais custosa em termos computacionais.

Técnicas de reconhecimento de objetos, como por exemplo, SIFT (Lowe 2004) e SURF (Bay, Ess et al. 2008), não possuem esse problema quanto à variação de orientação e escala. Contudo, para a localização de um caractere como sendo um objeto, elas não se mostraram eficientes, visto que um caractere não possui pontos característicos o suficiente para ser localizado em uma imagem do banco criado. As algoritmos SIFT (Lowe 2004) e SURF (Bay, Ess et al. 2008), são explicados no apêndice (página 160).

O uso de modelos para extrair a informação dos IDs garante um processamento rápido, já que a localização desta é conhecida anteriormente. No entanto, a localização da informação contida em um documento de identificação varia em documentos com diferentes formatos. Assim, os sistemas que usam modelos para classificar as informações de interesse com base em suas posições são limitados pelo

número de modelos que eles poderiam reconhecer. Na Figura 3-1 mostra-se um exemplo no qual o modelo não foi criado, e portanto o documento não é reconhecido,

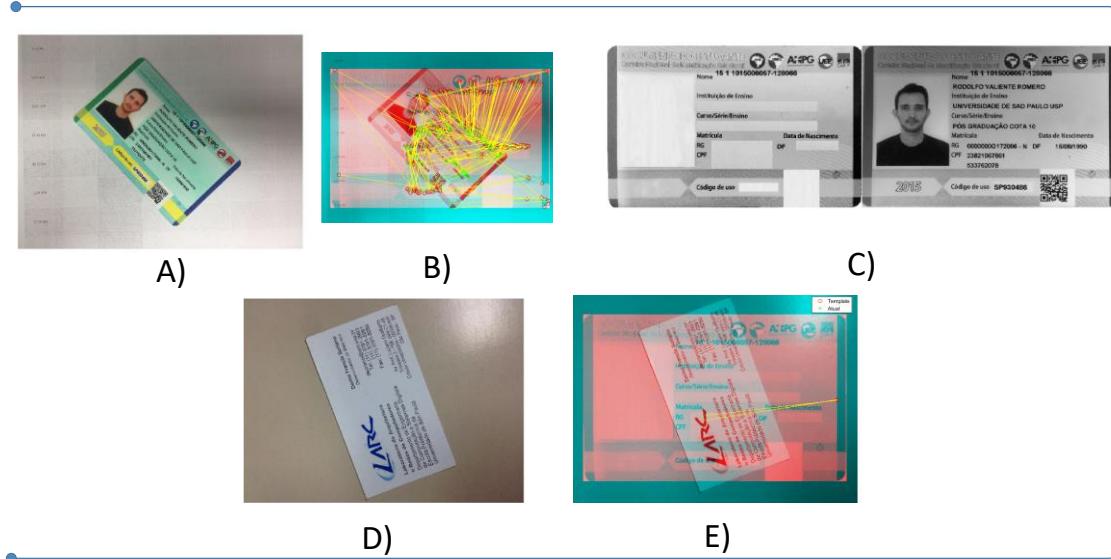


Figura 3-1: Processo de reconhecimento com modelo , a) documento do qual existe o modelo, b) reconhecimento do documento, c) extração do documento segundo o modelo, d) documento do qual não existe modelo, e) o documento não foi reconhecido, não tem pontos em comum com o modelo. Fonte: Autor.

O processo com modelo deve possuir um modelo feito previamente. Esse processo de criação de modelo é feito manualmente, retirando as informações do documento, ou seja, “limpando” o documento (Figura 3-2) e selecionando as regiões de interesse, como mostra a Figura 3-2.



Figura 3-2 Criação de Modelo. Fonte: Autor.



Figura 3-3: Processo de criação de modelo em MATLAB. Fonte: Autor.

Sabendo que os melhores resultados têm sido obtidos com a utilização da abordagem de processamento de imagens, optou-se por sua utilização (Yin, Yin et al. 2014, Sun, Huo et al. 2015). Porém, uma abordagem híbrida é proposta para trabalhos futuros com o objetivo de melhorar o desempenho do sistema.

Com o objetivo de superar as desvantagens anteriormente citadas o método de localização proposto neste trabalho visa localizar as regiões textuais da imagem usando processamento de imagens, a etapa de localização utiliza um método baseado em região e combina o algoritmo MSER com a detecção de borda e a melhora de contraste.

### 3.1.2 Etapa de seleção

Depois da segmentação de uma imagem ou trama em regiões através de métodos como os analisados, as regiões provenientes da segmentação, necessitam de serem representadas de forma eficiente para que possa mais facilmente serem processadas e classificadas pelo computador.

Existem diversas abordagens para o problema de seleção das regiões textuais que basicamente se resumem em selecionar os principais parâmetros que descrevem as regiões e posteriormente classificar cada região selecionada como sendo texto ou não (Chen, Bourlard et al. 2001, da Conceição Palma 2004, Yan, Lu et al. 2011, Gonzalez, Bergasa et al. 2012, Li, Lu et al. 2012, Neumann, Matas et al. 2012, Dong, Loy et al. 2016). A descrição ou representação de uma região pode ser efetuada com base nas suas características internas ou externas. Em qualquer dos casos, as

características escolhidas para descrever uma região devem ser tão insensíveis quanto possível a variações como mudanças de escala, rotações e translações. Uma das descrições mais usadas é a descrição da forma, duas classes principais de descritores de forma são consideradas: descritores de forma baseados no contorno e baseados em regiões. Os principais parâmetros dos descritores de forma usados na literatura podem ser organizados segundo as suas propriedades como: parâmetros geométricos, parâmetros baseados em transformadas, baseados em momentos e baseados em contornos normalizados (da Conceição Palma 2004). No apêndice (página 163) apresenta-se uma descrição destes parâmetros.

### **3.1.2.1 Métodos utilizados na classificação das regiões**

Depois de efetuada a descrição (representação) das várias regiões segmentadas, utilizando para tal um ou mais dos parâmetros/descritores anteriormente apresentados, torna-se necessário classificar cada região segmentada como sendo texto ou não. Para efetuar essa classificação podem ser encontrados na literatura vários tipos de métodos (Chen, Bourlard et al. 2001, Neumann, Matas et al. 2012). Dentre estes métodos, os mais utilizados são:

- Análise geométrica das regiões: Este tipo de método efetua comparações entre os valores dos parâmetros que descrevem as regiões segmentadas na imagem ou trama e determinados valores previamente definidos que caracterizam a presença de texto em termos dos parâmetros em questão. As regiões que não verificarem os critérios de filtragem que caracterizam o texto são classificadas como não texto e posteriormente descartadas. Alguns exemplos da aplicação de filtros baseados em heurísticas podem ser encontrados em (Zhong, Karu et al. 1995, Jain and Yu 1998, Gonzalez, Bergasa et al. 2012). Estes métodos apresentam dificuldades na classificação quando não são usadas as heurísticas com os valores corretos.
- Redes neurais: Este tipo de método utiliza redes neurais para efetuar a classificação de cada região como texto ou não. Os valores dos parâmetros que descrevem cada região segmentada servem de entrada para a rede neural. A resposta da rede é comparada com um limiar pré-definido característico da presença de texto para assim efetuar a classificação de cada região como texto ou não. Alguns exemplos da aplicação de métodos de classificação baseados em redes neurais podem ser encontrados em (Yan, Lu

et al. 2011, Li, Lu et al. 2012, Dong, Loy et al. 2016). A eficácia deste tipo de métodos depende muito da qualidade do treino feito à rede neural. Atendendo a que o texto possui vários tamanhos, fontes, estilos, etc., o treino de um classificador neural genérico torna-se particularmente difícil.

- Outros métodos existem, ainda que menos difundidos, para classificar como texto ou não, por exemplo, métodos baseados em operadores morfológicos , e métodos baseados em SVM (Chen, Bourlard et al. 2001).

A seguir são apresentados trabalhos relacionados que usam as diferentes técnicas anteriormente explicadas.

Em (Messelodi and Modena 1999) cada componente conexo representa uma região caracterizada por um contraste acentuado relativamente às regiões que lhe são vizinhas. Considera-se que o conjunto das regiões contém todos os componentes de texto misturados com aqueles que não são texto. A filtragem das regiões de texto é conseguida através da aplicação de regras heurísticas que permitem classificar cada uma delas como texto ou não texto. Estas regras baseiam-se em características das regiões conexas tais como: área, altura, largura, proximidade, excentricidade, solidez e contraste.

André Pires (Tahim 2010) usa um aprendizado supervisionado, 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 de frequências, etc. Essa etapa é comumente conhecida como extração de atributos. 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. Os atributos selecionados (extraídos do conjunto de treinamento) associados a suas classes correspondentes alimentam o algoritmo de aprendizado. Baseado no conjunto de exemplos e suas

classes correspondentes, o algoritmo de aprendizado cria uma regra de decisão capaz de prever exemplos desconhecidos. A etapa de seleção desenvolveu-se mediante a busca de atributos, extraídos da imagem, capazes de classificar as regiões localizadas. O conjunto de atributos com o melhor desempenho classificatório foi então utilizado para treinar uma SVM, criando-se dessa maneira um classificador binário, texto e não-texto.

Na tese de (Chen 2003) é apresentado um sistema de detecção e reconhecimento de texto em imagens e sequências de vídeo. Ele propõe uma abordagem de localização/verificação em duas etapas. O primeiro passo visa localizar rapidamente linhas de texto candidatas, permitindo a normalização de caracteres em um tamanho único. Na etapa de seleção, uma SVM treinada ou preceptores multicamadas são usados para remover os falsos positivos. Tal abordagem permite obter alto desempenho com um menor custo computacional em comparação com outros métodos.

A principal limitação destes métodos baseados em SVM está na quantidade considerável de regiões textuais identificadas na etapa de localização que são rejeitadas na etapa de seleção 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.

Com a finalidade de diferenciar corretamente as regiões de texto, mantendo um baixo custo computacional o método de seleção proposto neste trabalho usa propriedades heurísticas apropriadas com valores específicos para IDs e a largura do traçado.

### **3.1.3 Etapa de extração**

Após as etapas de localização e seleção das regiões textuais, o sistema de extração de texto 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, 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 (Lukas Neumann 2015, Huang, He et al. 2016, Jaderberg, Simonyan et al. 2016).

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 extração pode ser definida como a decomposição de uma imagem em regiões que são homogêneas de acordo com algum critério. O algoritmo de extração deve adaptar-se ao conteúdo da imagem, separando cada objeto desejado em uma região. As abordagens para o problema de extração de textos em imagens são divididas em técnicas de limiarização e métodos de clusterização de cores (Jung, Kim et al. 2004, Jung, Kim et al. 2004, Chen, Yin et al. 2016, Zhang, Lin et al. 2016).

O trabalho de André Pires (Tahim 2010) 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 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). Após a extração dos caracteres, são agrupadas e normalizadas as dimensões dos caracteres para alimentar o sistema de OCR.

O sistema apresentado não tem em consideração possíveis problemas de orientação ou rotação da imagem, não são usados algoritmos de retificação. Também não são consideradas imagens em baixa resolução que podem dificultar o processo de reconhecimento.

Duarte Manuel (da Conceição Palma 2004), desenvolve, implementa e avalia um mecanismo de extração automática de texto para imagens. Para diminuir a influência de alguns efeitos indesejáveis no desempenho final do processo de extração de texto, é proposta uma técnica para a simplificação das imagens que

preserva as zonas de elevado contraste (normalmente correspondentes a regiões de texto). O mecanismo de extração automática de texto em imagens desenvolvido explora, principalmente, o contraste existente entre o texto e o fundo da imagem, bem como a forma e a distribuição espacial dos caracteres. Foram propostas soluções melhoradas tanto para a segmentação, como para a detecção de caracteres: para ambos estes módulos, partiu-se de técnicas conhecidas, tendo-se introduzido melhorias de modo a alargar a sua gama de aplicação e a melhorar o seu desempenho na detecção de texto. Dentre estas melhorias, fazem parte técnicas para melhorar a precisão das fronteiras das regiões conexas detectadas na segmentação e técnicas que permitem melhorar a eficiência da detecção de caracteres com base na análise do contraste, nomeadamente em imagens pouco contrastadas. Foi também proposto um filtro que combina a detecção de fronteiras com um filtro de mediana de modo a diminuir a influência de alguns efeitos indesejáveis nas imagens ao mesmo tempo que preserva as zonas de elevado contraste (normalmente correspondentes a regiões de texto). Foram ainda propostas técnicas que permitem tanto detectar palavras com inclinações compreendidas entre 0 – 90º, como efetuar a sua rotação para a horizontal e o agrupamento das regiões classificadas como caracteres de texto de modo a formar palavras e linhas, com o objetivo de melhorar o resultado do OCR.

Porém o mecanismo não tem bom desempenho em extrair texto de pequenas dimensões e não são apresentadas técnicas para melhorar a resolução. Também existem dificuldades no reconhecimento de texto com sombra, texto tridimensional e texto com os mais variados formatos numa única palavra. Além disso, precisa-se aperfeiçoar a técnica para efetuar a extração de texto inclinado de modo a diminuir o número de falsas detecções para este tipo de texto.

Nesta dissertação não são consideradas contribuições na etapa de extração, esta etapa é incluída como pré-processamento antes do reconhecimento, propõe-se o uso de limiarização utilizando diferentes abordagens e a imagem é retificada para oferecer melhores resultados na etapa de reconhecimento.

### **3.1.4        Etapa de reconhecimento**

As etapas de localização, seleção 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.

As abordagens para o problema de reconhecimento de textos em imagens podem ser divididas em dois grupos: as que usam OCR convencionais e as que criam um próprio classificador de texto (Peng, Cao et al. 2013, Burie, Chazalon et al. 2015, Urbschat, Meier et al. 2015, Chabchoub, Kessentini et al. 2016, Islam, Mondal et al. 2016, Jaderberg, Simonyan et al. 2016, Neumann and Matas 2016, Sharma and Sharma 2016, Zhu and Zanibbi 2016). Cada etapa de tal transformação pode ser vista na Figura 1-7.

A seguir são apresentados trabalhos relacionados que usam as diferentes técnicas anteriormente explicadas, exibindo as suas vantagens e desvantagens.

Rodrigo Minetto (Minetto 2012), aborda o problema de detecção e reconhecimento de objetos de texto planos em imagens de cenas reais; para reconhecimento de texto desenvolve um novo descritor de imagem baseado em um Histograma de Gradientes Orientados (HOG) especializado para escrita romana, que denomina T-HOG. O objetivo é localizar os objetos físicos de texto na cena; o que não inclui a identificação dos caracteres que compõem aqueles textos (OCR). Explora o uso do HOG para o problema acima descrito. Classificadores baseados em HOG são utilizados para o reconhecimento de pedestres (Dalal and Triggs 2005), de objetos sólidos (Zhang, Zelinsky et al. 2007), e para a detecção de texto (Pan, Hou et al. 2008, Hanif and Prevost 2009, Wang, Huang et al. 2009). (Minetto 2012) descreve um novo classificador de texto, Text HOG (T-HOG) (Minetto, Thome et al. 2011) que acuradamente caracteriza textos de uma linha. O T-HOG é uma melhoria do HOG padrão, otimizado para a tarefa específica de reconhecimento de linhas de texto. Combina um detector de texto “permissivo”, SNOOPERTEXT (Minetto, Thome et al. 2010), com um classificador T-HOG como filtro de saída. Também melhora a eficiência do detector SNOOPERTEXT através de técnicas multiescala que permitem a detecção de caracteres com tamanhos variados. O classificador T-HOG desenvolvido pode ser utilizado em diversas aplicações tais como: a detecção de texto, rastreamento de texto, e reconhecimento por OCR.

Porém estes trabalhos não aproveitam um pré-processamento antes da realização do OCR o que reduz consideravelmente a taxa de acerto em imagens com baixa qualidade ou difíceis de segmentar. Além disso apresentam algumas desvantagens em comparação com o Tesseract, que é considerado o mecanismo de

OCR livre mais preciso existente (Smith 2007, Chandabenmohanbhai, Atulpatel et al. 2012, Heliński, Kmiecik et al. 2012, Mishra, Patvardhan et al. 2012).

Com o objetivo de superar as desvantagens anteriormente citadas, nesta dissertação é aplicado um pré-processamento como etapa previa antes do uso do Tesseract, além disso é criado um algoritmo iterativo para melhorar o resultado do OCR.

### **3.2 Trabalhos selecionados**

Como explicado anteriormente os documentos de identificação são considerados documentos complexos que contêm uma combinação de diferentes tipos de fonte, cores e artefatos de fundo. Como não existe um modelo de identificação internacional, alcançar um sistema capaz de localizar o texto no contexto complexo para qualquer modelo de identificação é um desafio difícil porém, necessário. O uso dos métodos descritos acima para extrair as informações de texto de imagens podem ser adaptados a documentos de identificação como descrito nos trabalhos apresentados a seguir (Sharma and Fujii , Sonia Bhaskar 2011, de las Heras, Terrades et al. 2015, Ryan and Hanafiah 2015, Simon, Rodner et al. 2015).

De las Heras et al. (Heras, Terrades et al. 2015) apresenta uma aplicação real para a classificação do documento de identificação de imagens tiradas de dispositivos móveis. O método proposto baseia-se na estrutura tradicional de *Bag-of-Words* (BoW), no qual um descritor SURF é usado para detectar as características locais e, em seguida, um algoritmo *k-means* agrupa essas características em um vocabulário de palavras representativas. Finalmente, cada imagem é representada como um histograma de características locais quantificadas, que é usado para treinar um classificador SVM para as diferentes classes de documentos. Além disso, para tornar o método proposto mais robusto, os autores treinam com três modificações que introduzem diferentes imagens no conjunto de dados. Para a segmentação eles confiam no conhecimento prévio de que os documentos de identificação contêm texto, e alguns deles incluem uma imagem facial do proprietário. Portanto, ao detectar o texto e a imagem de retrato, o documento é segmentado nos limites aproximados do documento. O algoritmo de localização de texto utilizado é baseado no proposto por Neumann (Neumann and Matas 2016). Os autores avaliam sua proposta com configurações diferentes em três conjuntos de dados contendo mais de 2000 imagens de 129 classes de documentos diferentes. Os conjuntos de dados são divididos em

imagens tiradas por celular, imagens digitalizadas e imagens sintéticas. Os resultados mostram como o desempenho dos métodos é aumentado após a adição de desfocagem ao treinamento. Esse aumento ocorre, uma vez que adicionar imagens desfocadas ao conjunto de treinamento leva o classificador a discriminar melhor entre classes de documentos similares. Além disso, a classificação de um documento segmentado é maior quando o plano de fundo é removido e a imagem resultante é semelhante à usada para treinamento.

Porém, o sistema poder ser usado somente com documentos previamente conhecidos e treinados no sistema, limitando suas aplicações.

Por outro lado Simon et al. (Simon, Rodner et al. 2015) propõem um sistema em que os documentos de identificação de diferentes países são classificados em diferentes classes uma vez que uma imagem de treinamento é usada por classe. Sua proposta é capaz de identificar o país de origem e o tipo de documento (cartões de identificação, passaportes, vistos e licenças de motorista) com uma precisão de 97,7% em um conjunto de dados com 375 imagens classificadas em 74 classes diferentes. Todas as imagens do conjunto de dados já estão recortadas e contêm apenas o documento de interesse. Os autores realizam uma avaliação de diferentes técnicas utilizadas para classificação com a restrição de usar apenas uma imagem de treinamento por caso.

No entanto, não descrevem o uso de algoritmos de localização e segmentação da informação textual o que reduz o desempenho do reconhecimento.

Em (Sonia Bhaskar 2011) é apresentado um algoritmo para o reconhecimento preciso do texto em um cartão de visita em diferentes condições ambientais. Primeiro, uma implementação no MATLAB do algoritmo é descrita na qual o objetivo principal é otimizar a imagem para entrada no mecanismo Tesseract OCR. Em seguida, é discutida uma implementação simplificada de complexidade reduzida para o sistema operacional Android. A implementação de MATLAB é bem-sucedida em uma variedade de condições ambientais adversas, incluindo iluminação variável em todo o cartão, fundo variado em torno do cartão, rotação, perspectiva e fluxo de texto horizontal variável.

As seguintes etapas foram usadas para pré-processamento: limiarização adaptativa e máscara de cartão, verificação de consistência através da direção de

fluxo do texto e transformação de perspectiva. A imagem foi subdividida em uma matriz de  $2 \times 3$  blocos, sobre os quais são calculados os limiares locais. Após o limiar adaptativo, realizaram-se a abertura, o fechamento e a dilatação usando operações morfológicas, utilizando elementos estruturais quadrados. O contorno do cartão foi então obtido e em seguida usada a transformação de Hough para detecção das linhas. O contorno em todos os casos é um quadrilátero, que é definido por quatro linhas. Portanto, os quatro maiores picos "não contíguos" na matriz Hough foram retirados. A partir da saída do comando do MATLAB, foram dados parâmetros para as quatro linhas em termos de  $\rho$  e  $\theta$ . Para determinar os quatro cantos do cartão de visita, as quatro equações para as linhas foram resolvidas para as suas interseções e as interseções relevantes dentro dos limites da imagem foram tomadas. Uma série de comparações foram então feitas para determinar qual canto era o canto "superior esquerdo", o canto "superior direito", e assim por diante. Foi realizada uma transformação de perspectiva baseada nos conjuntos de quatro pontos de canto. Finalmente, a imagem é redimensionada e realizada uma segmentação vertical tendo em conta que o Tesseract geralmente tem problemas para ler linhas de texto que não estão alinhadas horizontalmente.

Ryan et al. (Ryan and Hanafiah 2015) apresenta um sistema de reconhecimento de caracteres de documentos de identificação em quatro etapas: pré-processamento, extração de área de texto, segmentação e reconhecimento (Ver Figura 3-4). O experimento inclui alguns testes de escala de cinza e de algoritmos de segmentação, bem como a combinação deles. Na fase de pré-processamento é usada uma transformação para escala de cinza e binarização. Na transformação de escala de cinza são estudados e testados oito algoritmos: média, luminância, dessaturação, decomposição máxima, decomposição mínima, canal único colorido, canal de cor única verde e canal de cor azul único. A fase final no pré-processamento é a binarização, foram usados os algoritmos de NiBlack, Sauvola, Wolf e Khurshid no experimento.

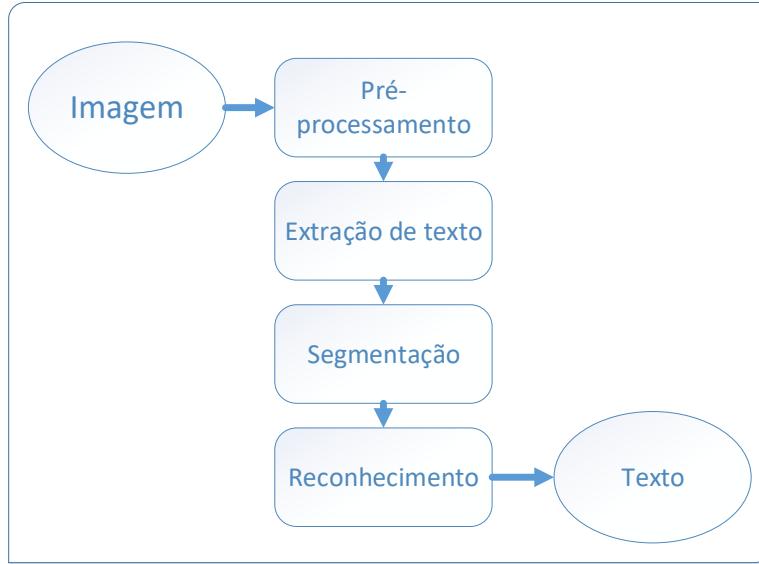


Figura 3-4: Sistema proposto por (Ryan and Hanafiah 2015). Fonte: adaptado de (Ryan and Hanafiah 2015).

Em seguida, a imagem em preto e branco é processada na extração da área de texto. Detectando as aparências de texto na imagem, separando a área em duas partes, região de texto e região de fundo. Um algoritmo de adição é usado na otimização da área de texto, e colocando cada texto em uma linha. A fase de segmentação leva essas linhas a serem segmentadas e produzem o texto segmentado para ser reconhecido pelo sistema de reconhecimento de caracteres. As imagens de teste usadas são apresentadas na Figura 3-5, o resultado após a segmentação é mostrado em Figura 3-6, as palavras extraídas e número de caracteres são apresentados na Tabela 3-1



Figura 3-5: Imagens de teste. Fonte: (Ryan and Hanafiah 2015).

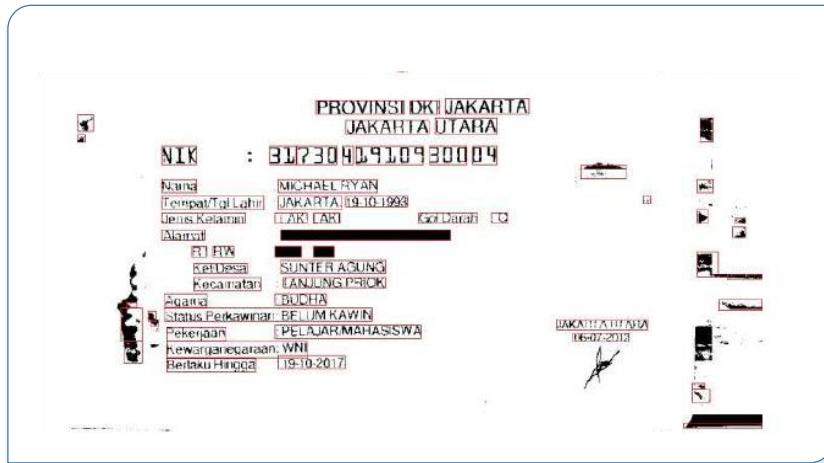


Figura 3-6: Resultados da extração. Fonte: (Ryan and Hanafiah 2015).

Tabela 3-1: Palavras extraídas e número de caracteres. Fonte: (Ryan and Hanafiah 2015).

Palavras extraídas	# de caracteres
PROVINSI DK.I JAKARTA	18
JAKARTA UTARA	12
3 I[X]3041910930004	20
Nama : Michael Ryan	16
Tempat/Tg!Lahir: Jakarta, 19-10-	33
1993 JenisKelamin : LAKI - LAKI	21
Go!. Darah: 0	10
Alamat: [ .... ]	28
RT/RW: [ .... ]	13
Kel/Desa : SUNTER AGUNG	20
Kecamatan: TANJUNG PRIOK	22
Agama : BUDHA	11
Status Perkawinan : BELUM KA WIN	27
Pekerjaan: PELAJAR/MAHASISWA	27
Kewarganegaraan : WN1	19
BerlakuHingga : 19-10-17	23
Total	320

A imagem de saída da fase de segmentação passa para a fase de reconhecimento de caracteres que consistem em várias etapas. No início, a imagem é normalizada alterando a imagem para o tamanho padrão (o mesmo tamanho que o tamanho do modelo). Posteriormente, o afinamento é conduzido aplicando o algoritmo Zhang-suen. Quando o processo de afinamento termina, a imagem está pronta para ser reconhecida. Os algoritmos de correspondência de modelos são usados para completar a fase de reconhecimento e fornecer o texto reconhecido como saída do sistema. Nos testes foram usados os seguintes IDs de Indonésia.

Na etapa de reconhecimento foi usada a *Figura 3-7*



Figura 3-7 Imagem usada na etapa de reconhecimento. Fonte: (Ryan and Hanafiah 2015).

O resultados após o reconhecimento apresenta-se na Figura 3-8

Sentence code	Result	Total letter	%
1	PR0viUSi?KiJ?KART	14	78
2	JAKA?TUT?A	8	67
3	NTKi3TT3D4191D93D001	12	60
4	N?m?i?iO??ELR?N	7	44
5	Tmp?t?giL?hirij???Ti1?41011?9?	16	48
6	JcMi?Kci?miniL??iLA?i	11	52
7	?ciD?Lho	3	30
8	Ai???ti?Q??Li?i?LQ???4?Q??	7	25
9	RTiR?0?6i?10	7	54
10	?ciDo???JINTERAGU??	8	40
11	??c?m?t??TNJ?NCFFi0?	9	41
12	A??????L?A	2	18
13	?t?tU?PcLk??iH?H?ELJ??A?N	10	37
14	?L??j?4??LAJ??i?A????sW	8	30
15	????L??u??????Ui?Ni	2	11
16	??Li?LJ?r?????1?iJ0i?017	4	17
<b>Sum</b>		<b>128</b>	<b>39</b>

Figura 3-8: Resultados após o reconhecimento. Fonte: (Ryan and Hanafiah 2015).

A melhor combinação de algoritmos de escala de cinza e binarização na fase de pré-processamento é alcançada, com o melhor parâmetro na etapa de escala de cinza para dar a melhor entrada para o processo de binarização. De 320 caracteres na imagem do cartão de identificação, o sistema segmentou aproximadamente 296 caracteres corretamente. No estágio de segmentação, aproximadamente 93% de caracteres podem ser cortados corretamente. A extração de área de texto mostrou resultados satisfatórios. No entanto, o processo de reconhecimento ainda precisa ser melhorado. Por estarem disponíveis as imagens de teste e porque no trabalho de Ryan são apresentados resultados em termos de precisão e revocação, este trabalho é usado para comparação com o sistema proposto nesta dissertação.

### 3.3 Comentários finais

Estes últimos sistemas tentam dar solução ao reconhecimento de texto em IDs genéricos e apresentam importantes contribuições, porém precisam melhorias na etapa de localização, não descrevem corretamente uma etapa de seleção e não são usadas técnicas de pré-processamento antes do reconhecimento. Portanto são

propostas nesta dissertação algumas estratégias para a melhoria do desempenho do reconhecimento de texto em IDs.

Nesta dissertação é proposto o uso de MSER combinado com uma melhoria do contraste e aproveitando a informação das bordas dos objetos da imagem, para conseguir extrair caracteres sobrepostos a um plano de fundo contendo variação abrupta de cor, regiões desfocadas e em baixa resolução. É melhorado o processo de seleção usando propriedades heurísticas apropriadas com valores específicos para IDs e a largura do traçado. Adicionalmente é aplicada a correção do texto usando a retificação na imagem. Finalmente, é realizado um pré-processamento como etapa antes do uso do Tesseract, e desenvolvido um algoritmo iterativo para melhorar o resultado do OCR. Desta forma, o método proposto nesta dissertação busca contemplar as estratégias acima identificadas para melhorar os sistemas de extração da informação textual em IDs.

Ao longo deste capítulo fez-se, ainda que sumariamente, a apresentação de vários sistemas e técnicas disponíveis na literatura visando a extração automática de texto em imagem. As técnicas básicas foram apresentadas de acordo com as etapas definidas para uma arquitetura básica de extração de texto, nomeadamente: localização, seleção, extração e reconhecimento.

A localização de imagem é, na maior parte das aplicações que visam a extração de texto em imagens, um passo preliminar e essencial. Embora muitas técnicas existam para esse efeito, a escolha de uma em detrimento de outra está intimamente relacionada com o tipo de características da imagem, bem como com o tipo de processamento que se pretende. Na Tabela 3-2 são resumidas as vantagens e desvantagens das várias técnicas de localização descritas no presente capítulo.

Tabela 3-2: Sumário das vantagens e desvantagens das técnicas de localização apresentadas. Fonte: Autor.

<b>Técnica de localização</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Baseada em CC	Baixo custo computacional; Eficazes para imagens simples. Fáceis de implementar	Número elevado de pequenas regiões que resultam das imagens com muita textura. Elevada sensibilidade ao ruído.
Baseada na Textura	Eficientes na detecção de homogeneidades em imagens com texturas de elevada variedade.	Elevado custo computacional. Requer uma varredura da imagem de entrada em diversas resoluções
Baseada em Bordas	Custo aceitável para o cálculo dos detectores de fronteira. Elevada exatidão na localização das fronteiras.	Elevada sensibilidade ao ruído. Necessidade de um processamento complexo para produzir fronteiras fechadas.

Na fase de seleção, as várias regiões provenientes da localização são classificadas como regiões de texto ou não texto. Para levar a cabo tal tarefa, foram descritas várias técnicas que permitem descrever as regiões para posterior classificação. A descrição das regiões tem como principal objetivo a captura das diferenças essenciais existentes entre as várias regiões e deve ser o mais insensível possível a variações de localização, tamanho ou orientação das regiões. Na Tabela 3-3 são resumidas as vantagens e desvantagens dos métodos majoritariamente utilizados para efetuar a seleção das regiões no âmbito da detecção de texto.

Tabela 3-3: Sumário das vantagens e desvantagens dos métodos de seleção de texto. Fonte: Autor.

<b>Técnica de Classificação</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Análise Geométrica das Regiões	Facilidade de implementação. Não necessitam de treino.	Dificuldades na classificação de texto de vários tamanhos. Dificuldades na classificação de texto quando os caracteres se tocam entre si.
Redes Neurais	Capacidade de aprender.	A sua eficácia depende muito do treino dado à rede. Grande dificuldade em treinar um classificador genérico, uma vez que o texto existente nos vídeos possui vários tamanhos, fontes, estilos, etc.

Os trabalhos utilizados como referências neste domínio descrevem sistemas capazes de reconhecer o texto em IDs. Na Tabela 3-4 são resumidas as vantagens e desvantagens dos abordagens usadas nos sistemas de reconhecimento de IDs.

Tabela 3-4: Resumo das vantagens e desvantagens dos abordagens usadas nos sistemas de reconhecimento de IDs. Fonte: Autor.

<b>Abordagens</b>	<b>Vantagens</b>	<b>Desvantagens</b>
Sistemas que usam modelos	Facilidade de implementação. Processamento rápido.	Limitados pelo número de modelos que eles poderiam reconhecer.
Sistemas sem modelos	Não estão limitados a um modelo. Podem ser usados em outras aplicações.	A sua eficácia depende muito dos algoritmos usados. Implementação complexa.

Observa-se que os sistemas que usam modelos para classificar as informações de interesse com base em suas posições são limitados pelo número de modelos que eles poderiam reconhecer. Portanto uma alternativa é o uso de sistemas sem modelos. A Tabela 3-5 apresenta as vantagens e desvantagens dos sistemas de reconhecimento de texto em IDs sem modelo apresentados.

Tabela 3-5: : Resumo das características dos sistemas de extração de texto sem modelos. Fonte: Autor.

<b>Sistemas de extração de texto apresentados</b>	
Vantagens	Taxa média de reconhecimento aceitável para determinadas aplicações Texto pode ser de cena ou gráfico e pode ainda caracterizar-se por diferentes tamanhos, fonte e estilos.
Desvantagens	Capacidade para identificar texto constituído por linhas orientadas em qualquer direção. Baixa taxa de reconhecimento em determinados cenários.

Uma vez que os sistemas de extração da informação textual em imagens de IDs é um problema complexo para o qual não existe uma técnica adequada para todos os tipos de conteúdo e situações, a sua solução passa muitas vezes pela combinação de várias técnicas, aproveitando as vantagens de cada uma, de forma a obter uma solução adequada às necessidades das várias aplicações. Ainda, precisa-se aperfeiçoar as técnicas já existentes e o desenvolver novas soluções com o objetivo final de superar as dificuldades até agora existentes, ver Tabela 3-5. É neste contexto que se insere o trabalho desenvolvido nesta dissertação e apresentado nos capítulos seguintes.

## Capítulo 4

### 4 Sistema de extração automática de texto proposto

Neste capítulo é apresentada a arquitetura do sistema de extração de texto, melhorando o resultado do OCR usando um método iterativo. O processo é implementado e testado usando o MATLAB e os experimentos realizados demonstraram que a solução proposta melhora significativamente o resultado do OCR. A arquitetura proposta é mostrada na Figura 4-1. Os blocos em verde destacam contribuições específicas do trabalho, fundamentalmente na etapa de seleção e na etapa de reconhecimento. O processo e algoritmos desenvolvidos assim como as contribuições serão explicadas a seguir.

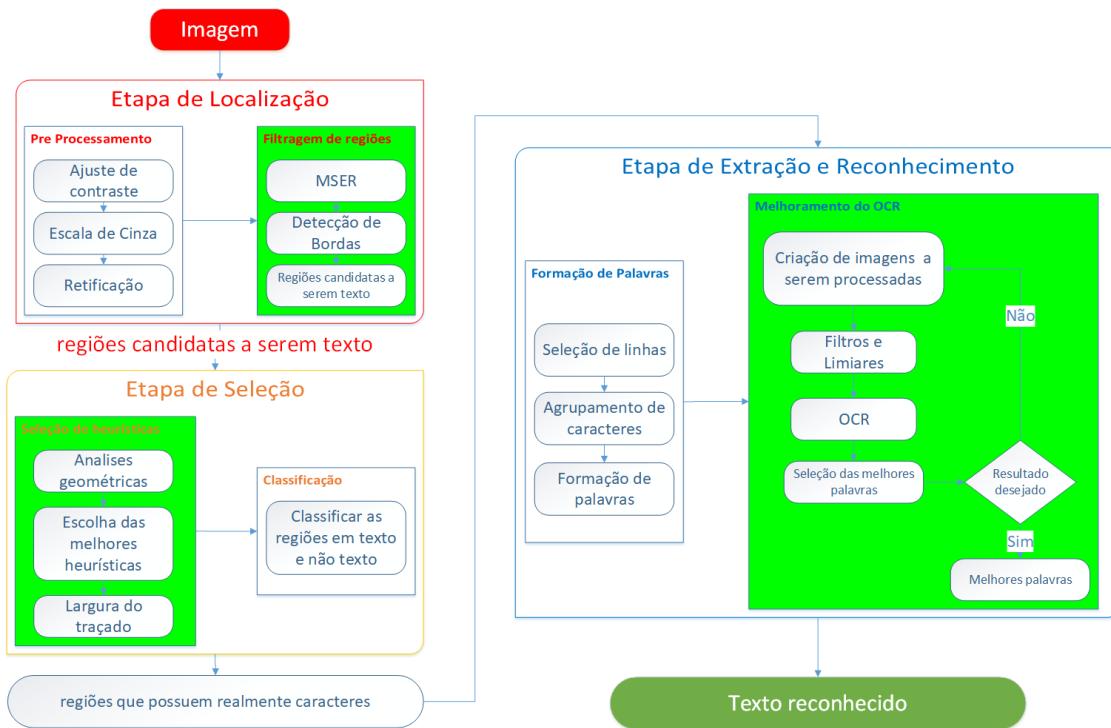


Figura 4-1: Arquitetura do sistema proposto. Fonte: Autor.

#### 4.1 Etapa de localização das regiões candidatas a serem texto

O método proposto visa localizar as regiões textuais da imagem mediante uma abordagem híbrida, a etapa de localização utiliza um método baseado em região. Combina o algoritmo MSER com a detecção de borda. Ver Figura 4-1.

#### 4.1.1 Pré-processamento

Como as imagens são tomadas em diferentes condições de iluminação e distâncias, as imagens precisam ser pré-processadas para melhorar as próximas etapas (Gonzalez, Bergasa et al. 2012) , Ver (Figura 4-1,1).

O pré-processamento reduz o ruído da imagem e aumenta a velocidade de processamento. A imagem é convertida em escala de cinza e, em seguida filtrada. São usados no pré-processamento o ajuste automático de contraste e redução de ruído através de um filtro de mediana. Ver Figura 4-2.

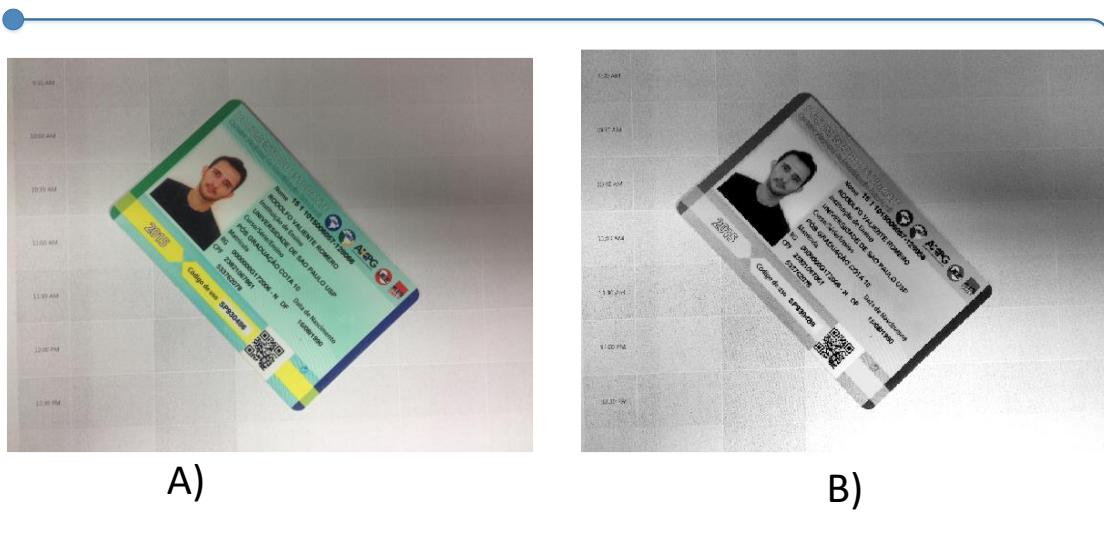


Figura 4-2: a) Imagem original , b) imagem em níveis de cinza e filtrada. Fonte: Autor.

Após a redução de ruído a imagem é retificada, primeiramente a imagem é subdividida em uma matriz de 3x6 blocos, sobre os quais são calculados os limiares locais, Figura 4-3a. Depois da limiarização, realizam-se a dilatação utilizando operações morfológicas com elementos estruturais quadrados, Figura 4-3b. Em seguida a transformada de Hough é calculada para a detecção das linhas de texto. A partir dos parâmetros obtidos das linhas (em termos de  $\rho$  e  $\theta$ ), é selecionada a maior linha e usado o ângulo de inclinação para retificar a imagem, realizando a rotação e extração do documento como observa-se na Figura 4-4.

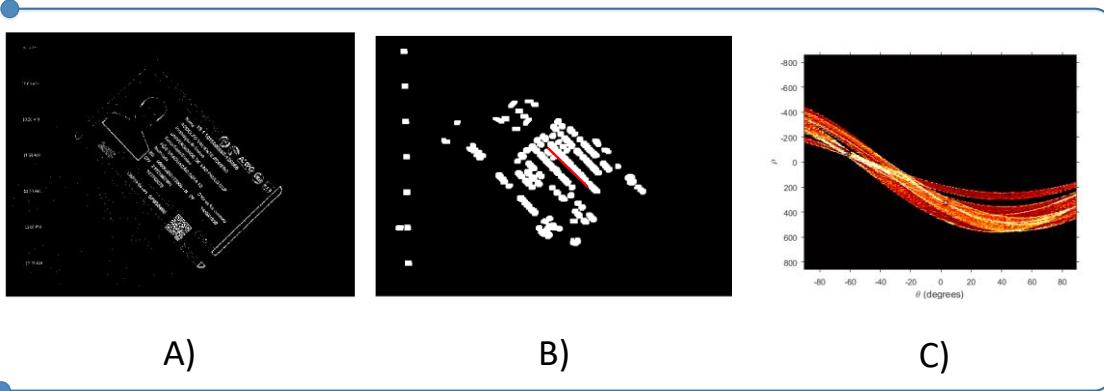


Figura 4-3: Retificação da imagem usando a transformada de Hough. Fonte: Autor.

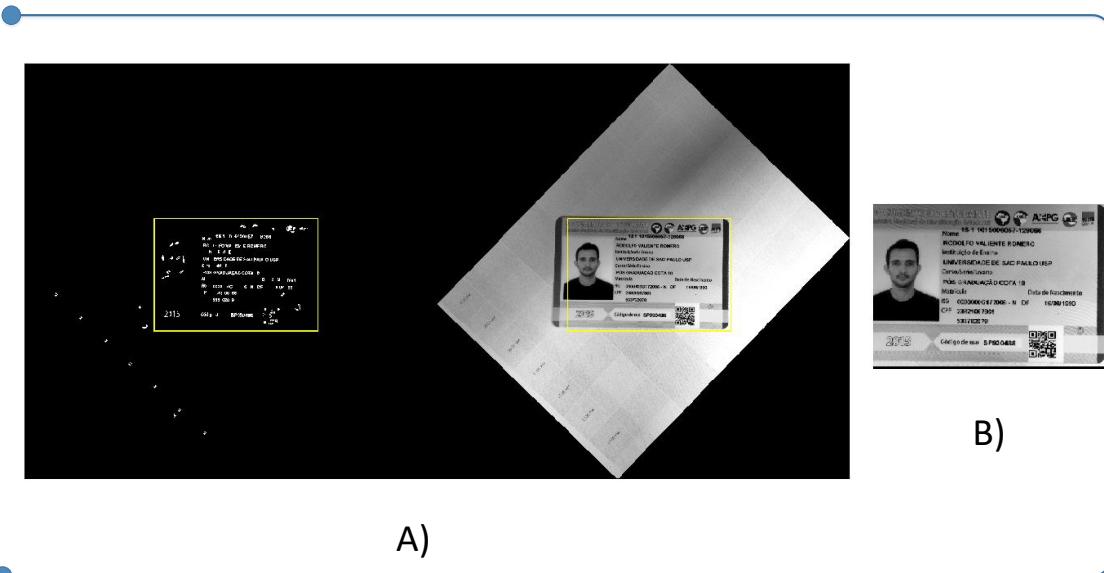


Figura 4-4: Retificação da imagem e seleção da região de interesse na qual está o documento. Fonte: Autor.

#### 4.1.1.1 Detecção de texto usando MSER

Uma vez retificado o documento precisa-se localizar as regiões textuais. Neste trabalho é usado o algoritmo MSER para localizar regiões que possuem texto (Li, Lu et al. 2012, Yin, Yin et al. 2014). O método de localização proposto explora a característica de contraste existente entre os pixels de contorno e o plano de fundo nos caracteres legíveis. É usado o algoritmo MSER junto com uma melhoria do contraste e aproveitando a informação das bordas dos objetos da imagem, para localizar os possíveis caracteres da imagem. O algoritmo MSER é usado de acordo com a seção 2.9, para melhorar o contraste é usado equalização de histograma (seção

2.3) e na detecção de bordas o algoritmo Canny (seção 2.6). O método identifica em uma única varredura as possíveis regiões textuais. Além disso, possui a vantagem de identificar os caracteres individualmente. O resultado do uso do MSER é mostrado na Figura 4-5.

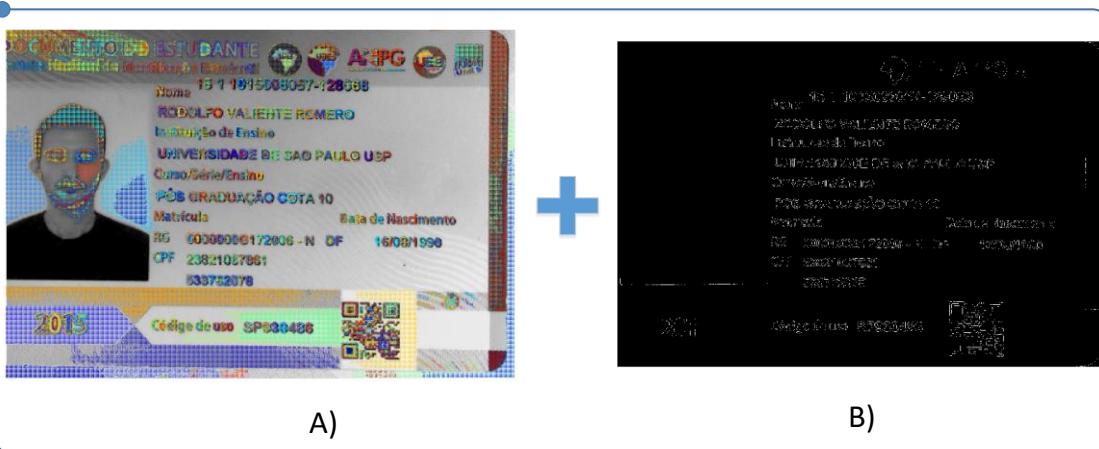


Figura 4-5 : a) Resultado do MSER, b) Resultado do algoritmo Canny. Fonte: Autor.

Embora o algoritmo MSER detecte a maior parte do texto, também detecta muitas outras regiões estáveis sobre a imagem que não são texto, portanto a etapa de seleção de tais áreas é realizada, proporcionando assim, uma maior robustez na localização dos caracteres. O método de localização delimita, na imagem, um conjunto de regiões com propriedades textuais.

#### **4.2 Etapa de seleção das regiões que possuem realmente caracteres**

O objetivo da etapa de seleção é classificar as áreas da imagem delimitadas 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 seleção pode ser vista como uma filtragem refinada das regiões obtidas durante a etapa de localização, aumentando a robustez na determinação das áreas textuais.

A etapa de seleção proposta neste trabalho desenvolveu-se mediante a busca de heurísticas, extraídas da imagem, capazes de classificar as regiões localizadas como textuais e não-textuais (Gonzalez, Bergasa et al. 2012). São propostas e usadas propriedades heurísticas apropriadas com valores específicos para IDs e a largura do

traçado para criar um classificador binário (texto e não-texto). As heurísticas propostas e os valores são explicados a seguir.

#### 4.2.1.1 Filtragem de propriedades geométricas

Existem várias propriedades geométricas que são boas para discriminar entre regiões de texto e de não texto definidas pelas fórmulas de (1) a (9), (Gonzalez, Bergasa et al. 2012, Li, Lu et al. 2012).

$$\text{Área} = \text{número de pixels da componente conexa} \quad (1)$$

$$\text{Proporção} = \frac{\text{largura}}{\text{altura}} \quad (2)$$

$$\text{Solidez} = \frac{\text{área}}{\text{área convexa}} \quad (3)$$

$$\text{Taxa de ocupação} = \frac{\text{área}}{\text{largura} * \text{altura}} \quad (4)$$

$$\text{Circularidade} = 4 * \pi * \frac{\text{área}}{\text{perímetro}^2} \quad (5)$$

$$\text{Taxa de ocupação da área conexa} = \frac{\text{área convexa}}{\text{largura} * \text{altura}} \quad (6)$$

$$\text{Número de Euler} = \# \text{ componentes conexos} - \# \text{ buracos} \quad (7)$$

$$\text{Excentricidade} = \sqrt{1 - \frac{\text{corda menor}^2}{\text{corda maior}^2}} \quad (8)$$

$$\text{Compacidade} = \frac{\text{perímetro}^2}{4 * \pi * \text{área}} \quad (9)$$

Para a seleção das heurísticas e obtenção dos valores adequados foram realizados dois experimentos:

- 1- Obtenção dos valores: Primeiramente foram usadas 3000 imagens, contendo somente regiões textuais, e foram calculados os valores adequados das heurísticas. O procedimento para o cálculo foi o seguinte: foi iniciado o valor da propriedade em 0 e aumentado em intervalos constantes, selecionando os componentes com valor menor que o valor da propriedade, assim até selecionar todos os caracteres. Para a obtenção dos valores foi criada uma função em MATLAB que automaticamente inicializa

a propriedade em 0, conta o número de caracteres detectados e aumenta seu valor até detectar todos os caracteres.

- 2- Seleção das propriedades mais discriminativas: Foram usadas 2000 imagens, contendo somente componentes que não representam caracteres, e foram calculados o número de regiões filtradas (selecionadas como não caracteres) por cada heurística independentemente, usando os valores obtidos no experimento 1, a porcentagem de regiões filtradas para cada propriedade se mostra na Tabela 4-1. Foram selecionadas as quatro propriedades mais discriminativas, essas propriedades usadas uma a seguir da outra (primeiro as regiões são filtradas segundo a área, depois as regiões restantes são filtradas segundo a proporção, depois a solidez e finalmente a excentricidade), filtram o 94% das regiões que não são caracteres. Ver Tabela 4-2.

Tabela 4-1 : Porcentagem de regiões filtradas para cada propriedade usada independentemente. Fonte: Autor.

<b>Propriedades</b>	<b>Número de Regiões Filtradas em %</b>
Área	69
Proporção	57
Solidez	41
Taxa de ocupação	11
Circularidade	19
Taxa de ocupação da área conexa	16
Número de Euler	8
Excentricidade	38
Compacidade	22

Tabela 4-2: Porcentagem de regiões filtradas para as propriedade heurísticas quando usadas uma a seguir da outra. Fonte: Autor.

<b>Propriedades</b>	<b>Número de Regiões</b>
	<b>Filtradas em %</b>
Área	69
Proporção	16
Solidez	5
Excentricidade	5
Total	94

Considerando os experimentos realizados, as melhores propriedades discriminativas obtidas experimentalmente foram: Proporção da imagem, excentricidade, solidez e área.

São descartadas as regiões segundo os seguintes critérios:

- Área: As regiões com uma área inferior a 12 pixels ou maior do que 1/10 da área da imagem.
- Proporção: Regiões com proporção superior a 5 ou inferior a 2.
- Excentricidade: Regiões com uma excentricidade superior a 0.9.
- Solidez: Regiões com uma solidez inferior a 0,3.

Além dessas regras, foi adicionada outra regra que ajudou a eliminar ruído. Esta regra indica que a relação entre a contagem de pixels do componente e a quantidade de pixels na caixa delimitadora (BB) do componente deve estar dentro de um intervalo limitado. Isso rejeita componentes que possuem uma pequena contagem de pixels (relação 1/100 da área do BB) e componentes que cobrem a maior parte de sua caixa delimitadora (relação 2/3 da área do BB).

A Tabela 4-3 mostra as propriedades heurísticas apropriadas obtidas, com os valores específicos para IDs, que foram calculados experimentalmente.

Tabela 4-3: Propriedades heurísticas.  $t$  = número de pixels da imagem. Fonte: Autor.

Propriedades ( $P_G$ )	Valor (v)
Proporção ( $P_c$ )	$0.2 < p_c < 5$
Excentricidade ( $E_c$ )	$e_c < 0.9$
Solidez ( $S_c$ )	$S_c > 0.3$
Área ( $A_c$ )	$12 < a_c < (1/10) * t$

A Figura 4-6 mostra o resultado da aplicação das heurísticas para diferenciar corretamente as regiões de texto, observa-se como muitas regiões não textuais foram eliminadas.



Figura 4-6: Uso das heurísticas para eliminar possíveis regiões não textuais. Fonte: Autor.

#### 4.2.1.2 Largura do traçado

A largura do traçado é outra métrica comum usada para discriminar entre texto e não-texto. A largura do traçado permanece quase a mesma num caractere. No entanto, há uma mudança significativa na largura do traçado em regiões não-texto como resultado da sua irregularidade (Epshtain, Ofek et al. 2010, Li, Lu et al. 2012). Regiões de texto tendem a ter pouca variação de largura, enquanto que as regiões que não são de texto tendem a ter maiores variações. Assim, além do uso das heurísticas anteriormente explicadas as regiões com variações consideráveis na largura do traçado também são removidas.

Depois de encontrados os candidatos a texto na etapa de localização e após a primeira filtragem usando as propriedades geométricas. É calculada a largura do traço para cada componente.

A largura do traço é calculada usando a transformada da distância e o esqueleto da imagem (ver seção 2.8). Na Figura 4-7 observa-se o resultado da aplicação para três componentes (1,2,3) da transformada da distância (b) e o esqueleto da imagem (c). Os pontos de maior intensidade (mais brancos) na transformada da distância são os pontos mais separados das bordas. Usando o esqueleto da imagem e a transformada da distância calculamos a largura do traçado (indicada em vermelho) para cada componente, a largura de traçado varia pouco em letras Figura 4-7 (1,2) e tem maior variação em componentes que não representam letras Figura 4-7 (3).

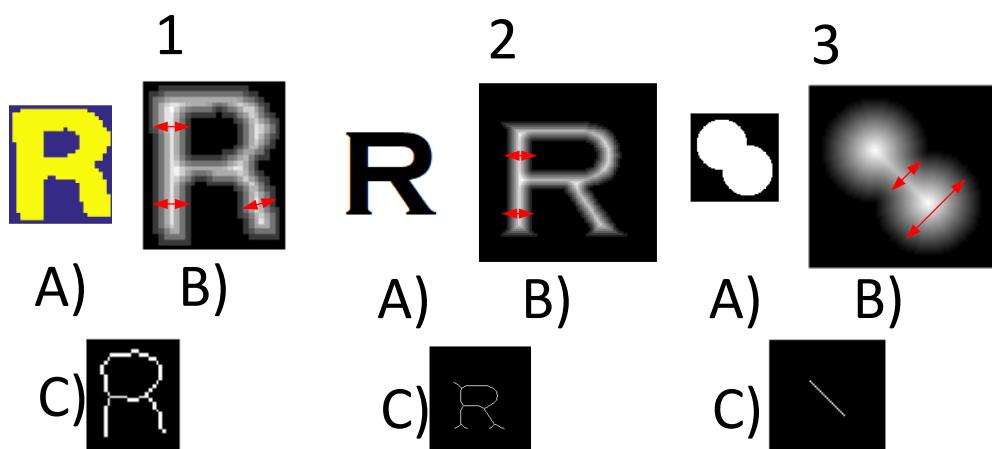


Figura 4-7 Transformada da distância (b) e o esqueleto da imagem (c) para 3 componentes. Fonte: Autor.

Agora temos um mapa das larguras de traçado para cada pixel de cada componente. O próximo passo é determinar os componentes candidatos a serem letras. Isso será feito tendo em conta que a variância da largura do traço dentro de um componente não deve ser muito grande.

Finalmente, o método de seleção proposto neste trabalho usa as heurísticas com valores específicos para IDs anteriormente mencionadas e a variação da largura

do traçado. O algoritmo proposto na etapa de seleção é descrito a seguir. O resultado da aplicação do algoritmo é mostrado na Figura 4-8, em A) está a imagem após a etapa de localização, B) após o uso das propriedades geométricas, C) após o uso da largura do traçado. É importante observar como o “R” na Figura 4-8f por ter uma largura do traço pouco variável, é mantida em Figura 4-8c e corretamente classificada como letra, porém o componente Figura 4-8e é eliminado após a filtragem da imagem usando a largura do traçado Figura 4-8c, pois não é uma letra, há uma mudança significativa na largura do traçado como resultado da sua irregularidade.

---

#### **Algoritmo 1- Algoritmo proposto na etapa de seleção**

---

**Entrada:** Regiões MSER  $Re$ ,  $Re \in f \subset \mathbb{Z} \times \mathbb{Z}$ ;

Propriedades a calcular,  $P_G \{P_c, E_c, S_c, A_c\}$ ;

Valores heurísticos,  $V_H \{p_c, e_c, s_c, a_c\}$ ;

**Saída:** Regiões textuais  $Retext^*$

---

$\forall Re$  calcular as propriedades geométricas,  $P_G \{P_c, E_c, S_c, A_c\}$

Se os valores das propriedades são adequados,

$$Retext = \{ Re : \text{Valores de } P_G \subset V_H \}$$

$\forall Retext$  calcular:

- transformada da distância  $d = \Psi d(Retext)$
- esqueleto da imagem  $e = \Psi e(Retext)$
- largura do traçado  $Sw = d(\Psi e)$
- variação da largura do traçado

$$\Delta Sw = \sigma(Sw)/\mu(Sw)$$

Se os valores da variação da largura do traçado são adequados,

$$Retext^* = \{ Retext : \Delta Sw < 0.5 \}$$


---

$Retext^*$  regiões candidatas a texto selecionadas

---

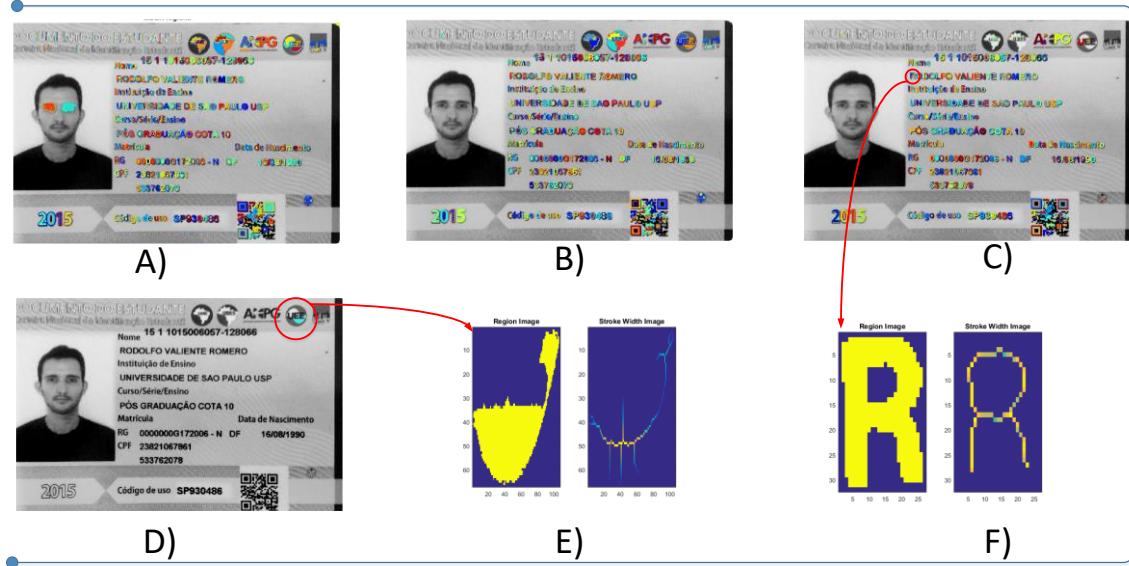


Figura 4-8 Resultado após da etapa de seleção. Fonte: Autor.

O principal objetivo da etapa de seleção 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 e MSER, o método de classificação proposto utiliza métodos estruturais para a certificação de tais regiões como textuais. A utilização de diferentes abordagens promove robustez à identificação de regiões textuais.

#### 4.3 Etapa de reconhecimento

Os métodos de localização e seleção propostos buscam 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.

É apresentado o método proposto para agrupar caracteres de uma mesma palavra e em seguida realizado o pré-processamento antes do uso do Tesseract, finalmente é proposto um algoritmo iterativo para melhorar o resultado do OCR.

### 4.3.1 Mesclar caracteres em palavras

Neste ponto, todos os resultados detectados são constituídos por caracteres de texto individuais. Para usar esses resultados em tarefas de reconhecimento, tais como OCR, os caracteres de texto individuais devem ser fundidos em palavras ou linhas de texto, que carregam informações mais significativas do que apenas os caracteres individuais. Os caracteres são agrupados em palavras com base na distância, orientação e semelhanças entre os caracteres.

O sistema calcula a altura das letras do texto, a distância entre elas e a sobreposição dos BBs. As regiões de texto individuais são fundidas em palavras ou linhas de texto encontrando regiões de texto vizinhos. Isto é conseguido através do aumento das caixas delimitadoras de cada caráter individual calculado anteriormente, até que eles se sobreponham. As caixas delimitadoras sobrepostas podem ser fundidas para formar uma cadeia de caixas delimitadoras simples horizontais em torno de palavras individuais ou linhas de texto.

Antes de mostrar os resultados de detecção finais, as detecções de falsos textos são suprimidas, removendo as caixas delimitadoras com apenas uma região de texto dentro. Este processo remove regiões isoladas com pouca probabilidade de ser uma palavra, tendo em conta que cada texto é geralmente encontrado em grupos (palavras e frases). O resultado da formação das linhas mostra-se na Figura 4-9

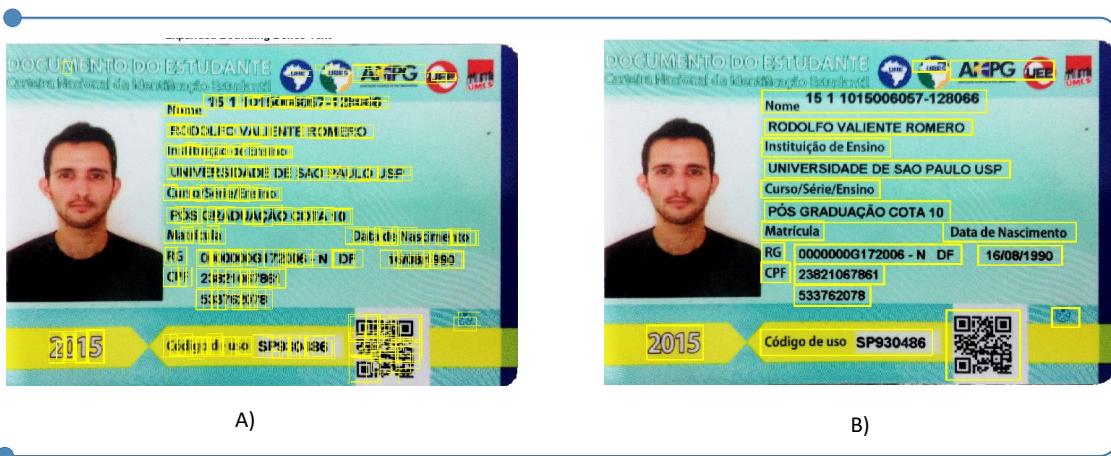


Figura 4-9: União dos BBs e formação das linhas de palavras b). Fonte: Autor.

### 4.3.2 Melhoria do OCR e escolha das melhores palavras

Uma vez formadas a linha é proposto um algoritmo iterativo para melhorar o resultado do OCR. A resolução da imagem é aumentada a 300dpi. O aumento da

resolução da imagem antes de usar OCR é recomendada, pois algoritmos e ferramentas de OCR produzem bons resultados com imagens em alta resolução, e de boa qualidade. Para textos normais (tamanho de fonte 8-10 pontos), recomenda-se usar a resolução de 300 dpi para OCR.

Em seguida, a imagem é submetida a modificações do brilho, contraste, diferentes limiares. Cada mudança, gera uma nova imagem e consequentemente, esta, quando passada para o OCR, gera um novo arquivo-texto. Baseado nestas condições, a seguir descreve-se o processo de criação das imagens e a escolha dos melhores resultados através da seleção das palavras dentre as várias palavras geradas como resultado do OCR nas diferentes imagens.

São usados vários filtros e a em seguida é usado o OCR para cada uma das palavras dentro de cada caixa delimitadora, para todas as imagens criadas. Na seleção das melhores palavras é usado um dicionário e tendo em conta o parâmetro de confiabilidade do OCR, valor de confiança dado pela função de OCR. Se o resultado da confiabilidade para a palavra está dentro do intervalo desejado, é realizado o mesmo processo com a próxima palavra na imagem até obter todas as palavras com a confiabilidade desejada; se não, são modificados os parâmetros dos limiares, geradas novas imagens e realizado o OCR novamente, assim até obter o resultado desejado ou chegar a 5 iterações. O algoritmo proposto é ilustrado na Figura 4-10 e explicado a seguir.

## Melhoramento do OCR

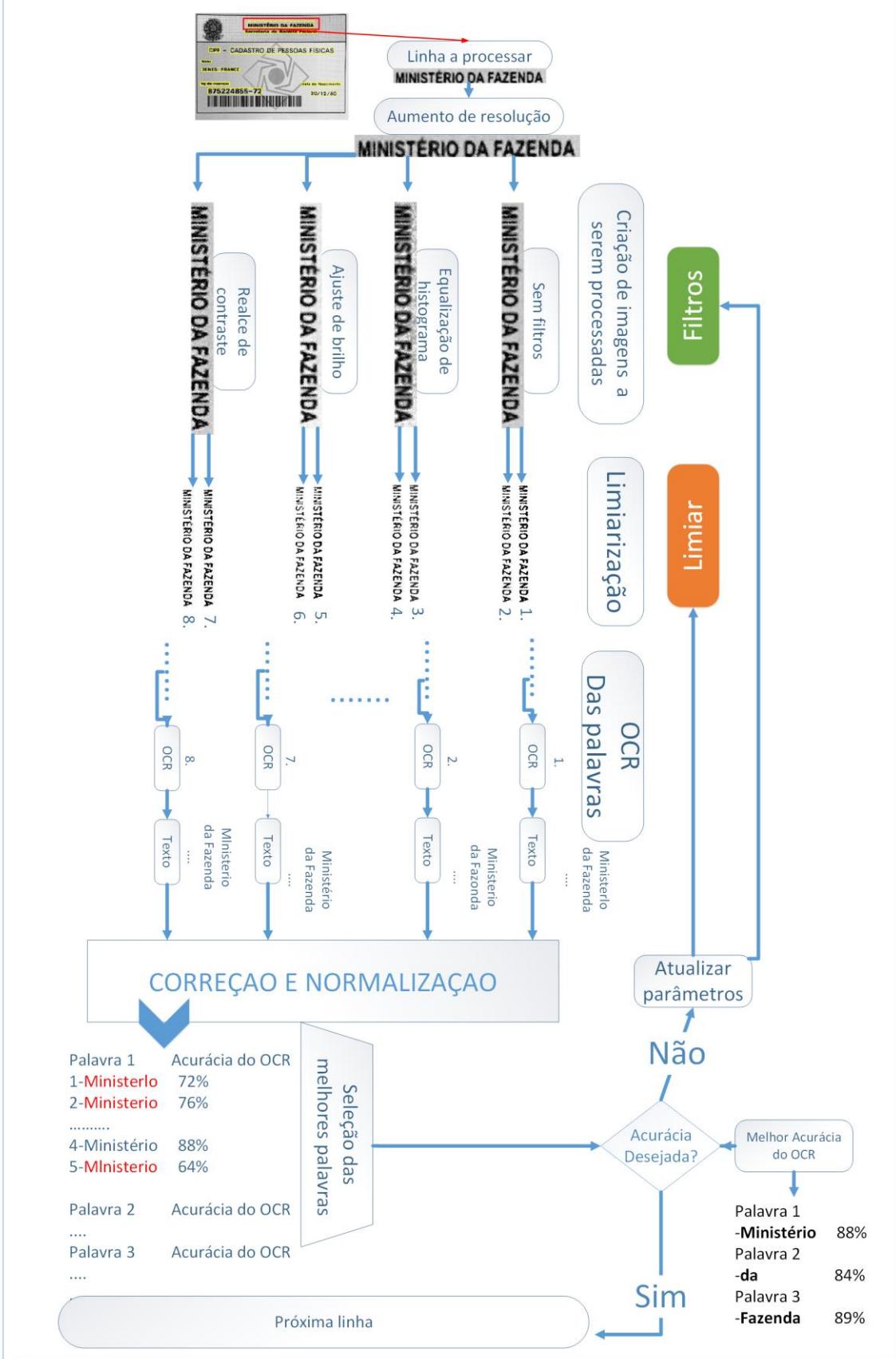


Figura 4-10: Arquitetura para o melhoramento do OCR. Fonte: Autor.

Dada uma região de interesse (ROI) Q de uma imagem I ( $f(x,y)$ ). O processo de melhora do reconhecimento tem a seguinte sequência: utiliza-se a imagem Q para gerar um grupo de 8 imagens. Inicialmente são geradas 4 imagens  $Q_{f1}$ ,  $Q_{f2}$ ,  $Q_{f3}$ ,  $Q_{f4}$ . Sendo  $Q_{f1}$  a imagem Q com a resolução aumentada,  $Q_{f2}$  a imagem Q após uma equalização de histograma,  $Q_{f3}$  a imagem Q após do ajuste de brilho,  $Q_{f4}$  a imagem Q após do realce de contraste. Usando as imagens  $Q_{f1}$ ,  $Q_{f2}$ ,  $Q_{f3}$ ,  $Q_{f4}$  para cada imagem são geradas 2 novas imagens, a primeira é obtida após uma limiarização local usando o método de Otsu  $Q_{fxt1}$ , e a segunda após uma limiarização adaptativa  $Q_{fxt2}$ , como resultado são geradas 8 imagens  $Q_{f1t1}$ ,  $Q_{f2t1}$ ,  $Q_{f3t1}$ ,  $Q_{f4t1}$ ,  $Q_{f1t2}$ ,  $Q_{f2t2}$ ,  $Q_{f3t2}$ ,  $Q_{f4t2}$ .

Os parâmetros para as transformações são definidos a seguir ( $P_{f1}$ ,  $P_{f2}$ ,  $P_{f3}$ ,  $P_{f4}$ ,  $P_{t1}$ ,  $P_{t2}$ ) e foram obtidos experimentalmente:

- $P_{f1}$  é usado para obter  $Q_{f1}$ , aumento de resolução a 300dpi usando método bicúbico.
- $P_{f2}$  é usado para obter  $Q_{f2}$ , equalização de histograma usando a equação descrita na seção 2.3.
- $P_{f3}$  é usado para obter  $Q_{f3}$ , ajuste de brilho, entre 10 % e 90%, evitando imagens muito claras ou escuras.
- $P_{f4}$  é usado para obter  $Q_{f4}$ , realce de contraste usando o kernel apresentado na seção 2.3.
- $P_{t1}$  é usado para obter  $Q_{fxt1}$  limiarização adaptativa usando o método de Otsu, janela 20x20, seção 2.4.
- $P_{t2}$  é usado para obter  $Q_{fxt2}$ , limiarização local usando o método de Niblack, valor 0,6 para K e janela W de 25 × 25, seção 2.4.

Na figura mostra-se o resultado da aplicação da primeira parte do algoritmo na região de interesse mostrada na Figura 4-11b (ROI pertencente à imagem Figura 4-11a.).  $Q_{f1t1}$ ,  $Q_{f2t1}$ ,  $Q_{f3t1}$ ,  $Q_{f4t1}$ ,  $Q_{f1t2}$ ,  $Q_{f2t2}$ ,  $Q_{f3t2}$ ,  $Q_{f4t2}$ , representam as 8 imagens geradas, observa-se como o resultado é diferente para diferentes transformações.

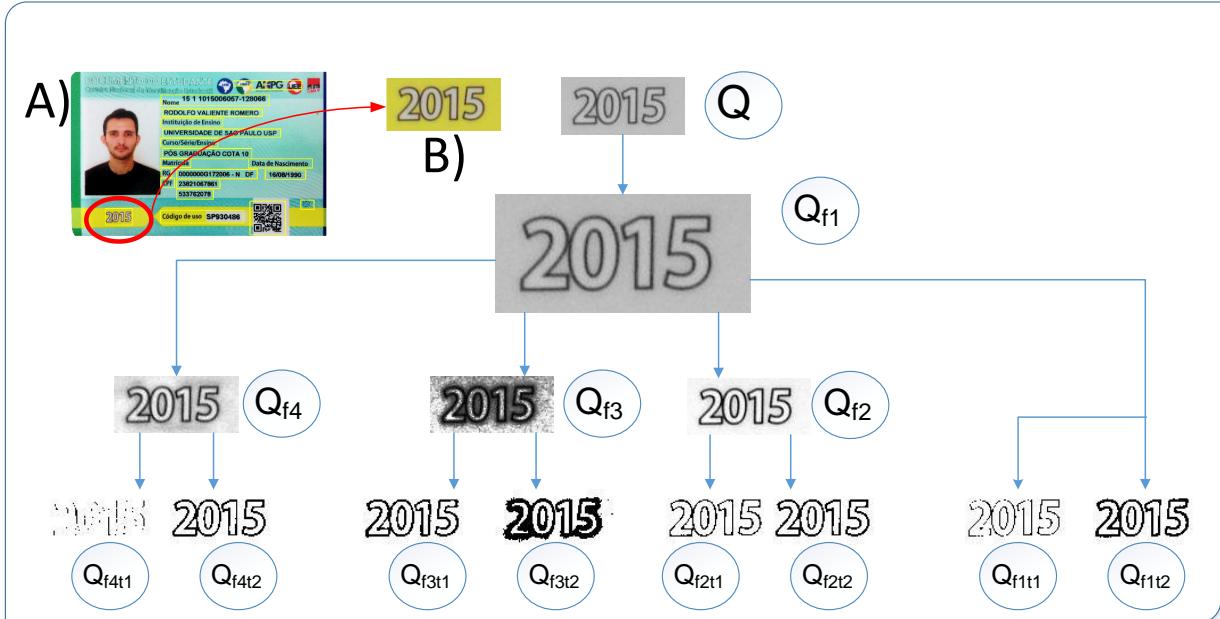


Figura 4-11 Imagens geradas:  $Q_{f1t1}, Q_{f2t1}, Q_{f3t1}, Q_{f4t1}, Q_{f1t2}, Q_{f2t2}, Q_{f3t2}, Q_{f4t2}$

O algoritmo proposto é o seguinte:

---

### **Algoritmo 2- Algoritmo proposto na etapa de reconhecimento**

---

**Entrada:** Regiões de interesse  $Q$ ,  $Q \in f \subset \mathbb{Z} \times \mathbb{Z}$ ;

Parâmetros dos filtros e limiares  $P_i \{P_{f1}, P_{f2}, P_{f3}, P_{f4}, P_{t1}, P_{t2}\}$

Variação dos parâmetros:  $\Delta_p = 10$

Número de iterações:  $n^* = 5$ ; Acurácia do OCR desejado:  $A_c^* = 85\%$

**Saída:** Texto  $Text$

---

$\forall Q$

1-gerar,  $Q_{xx} \{Q_{f1t1}, Q_{f2t1}, Q_{f3t1}, Q_{f4t1}, Q_{f1t2}, Q_{f2t2}, Q_{f3t2}, Q_{f4t2}\}$

$\forall Q_{xx}$  Realizar OCR:

- Separa as palavras reconhecidas segundo o valor de  $A_c$
- Usar dicionário para corrigir as palavras.

Para cada palavra lida pelo OCR, se  $A_c < A_c^*$ , aumentar os parâmetros  $P_{t1}, P_{t2}$  em  $\Delta_p$ , e voltar a 1.

Realizar o processo até  $A_c > A_c^*$  ou  $n > n^*$

:  $Text = \text{palavras lidas}$

---

$Text$  melhores palavras para cada caixa delimitadora em cada imagem

---

A saída do algoritmo 2 é o melhor resultado para cada palavra. Duas considerações são interessantes:

- Para este trabalho o algoritmo somente gera um grupo de 8 imagens com as transformações explicadas (os parâmetros das transformações são configuráveis), porém, outro grupo de imagens e transformações podem serem utilizados. As transformações utilizadas foram selecionadas experimentalmente por oferecer os melhores resultados nos testes realizados.
- Nesta trabalho se na primeira iteração não é obtida a acurácia desejada; então, nas próximas iterações (número configurável), somente são modificados os parâmetros dos limiares  $P_{t1}, P_{t2}$ ; porém é possível modificar os outros parâmetros. Nesta dissertação foram modificados somente  $P_{t1}, P_{t2}$  pois nos experimentos realizados não foram obtidas grandes diferenças variando os outros parâmetros.

#### **4.4 Comentários finais**

Ao longo deste capítulo foram propostos algoritmos que permitem e melhoram a seleção e reconhecimento de texto em imagens de IDs. O texto pode ser constituído por caracteres de vários tamanhos, fontes e cores e estar escrito em qualquer direção. O método proposto começa por efetuar a localização das regiões textuais, posteriormente, filtradas de acordo com várias heurísticas, eliminado regiões que não correspondem a texto. Finalmente, é realizado o reconhecimento

As contribuições do trabalho estão principalmente na etapa de seleção e reconhecimento, bem como do seu aperfeiçoamento e adaptação para texto de IDs, em comparação com técnicas já conhecidas, mas que foram desenvolvidas para outros cenários.

## Capítulo 5

### 5 Validação e resultados

A partir da abordagem proposta neste trabalho na etapa de seleção e reconhecimento de texto, a avaliação dos resultados considerará o desempenho em três situações: no processo de seleção, no processo de reconhecimento e no sistema em geral. O desempenho do sistema é comparado com o método de (Ryan and Hanafiah 2015), anteriormente apresentado em trabalhos relacionados. Adicionalmente, serão apresentados: o desempenho das heurísticas usadas na etapa de seleção, a avaliação dos parâmetros obtidos na etapa de reconhecimento, e os resultados do uso do sistema em um ambiente real. Finalmente, apresentam-se outras aplicações do trabalho e resultados obtidos para outros cenários.

Sendo assim, este capítulo traz, primeiramente, os bancos de imagens usados para os experimentos e a seguir as métricas utilizadas para a avaliação dos resultados. Em seguida, são mostrados os resultados propriamente ditos, divididos de acordo com o que foi detalhado acima. Os algoritmos apresentados neste trabalho são implementados em MATLAB R2016a. As implementações e testes são realizadas em um computador com processador Intel core i3 de 1,8 GHz, 6 Gb de RAM e sistema operacional Windows 7.

#### 5.1 Elaboração do banco de imagens

Para avaliar os algoritmos propostos foram construídos 5 bancos de imagens.

- Banco de imagens 1: 100 imagens de documentos de identificação retificadas, usadas para avaliar o algoritmo de seleção (1321 palavras).
- Banco de imagens 2: 100 imagens de documentos de identificação com as caixas delimitadoras previamente selecionadas, usadas para avaliar o algoritmo de reconhecimento (934 palavras).
- Banco de imagens 3: 2000 imagens de regiões textuais (somente de regiões textuais, sendo um total de 8266 caracteres) tomadas de imagens de documentos de identificação usadas para avaliar as propriedades heurísticas apropriadas e os parâmetros padrões no algoritmo de reconhecimento.

- Banco de imagens 4: 300 imagens de documentos de identificação em diferentes condições para testar o funcionamento do sistema de reconhecimento de texto (3148 palavras).
- Banco de imagens 5: 100 imagens (20 imagens diferentes, com 5 pontos de vista diferentes) de documentos de identificação usadas para testar o sistema em um ambiente real (1100 palavras).

Os banco de imagens estão formados por documentos de identificação brasileiros assim como outros documentos mencionados a seguir: CNH (Carteira Nacional de Habilitação), RG (Carteira de Identidade ou Registro Geral), RNE (Registro Nacional de Estrangeiros), CPF (Cadastro de Pessoa Física), BUSP (Bilhete USP), bilhete único de estudante e carteira de estudante, ver Figura 5-1. Os IDs em estudo são sujeitos a uma variedade de condições adversas, incluindo iluminação variável, fundo, rotação, e inclinação do texto. Dentro do ambiente do sistema desenvolvido neste trabalho as imagens são capturadas por diferentes dispositivos e a aquisição destas é feita em cores. As dimensões das imagens do conjunto de teste variam entre  $196 \times 138$  e  $3000 \times 1912$  pixels, contendo uma ampla variedade de planos de fundo e texto.



Figura 5-1: Amostra do banco de imagens criado. Fonte: Autor.



Figura 5-2: Imagens de regiões textuais tomadas de imagens de IDs. Fonte: Autor.

Para criar o banco de imagens textuais, foi desenvolvido um processo com modelo (ver 3.1.1.1) , no qual as imagens dos documentos foram usadas para criar os modelo e em seguida automaticamente foram extraídas as regiões textuais usadas no banco de imagens. Na Figura 5-3 mostra-se um exemplo para uma carteira de habilitação, o modelo é criado e em seguida para cada documento são extraídas automaticamente as regiões textuais, evitando o processo manual. Na Figura 5-3a tem-se a imagem do ID, na Figura 5-3b observa-se o processo de correspondência de modelos usando SURF, na Figura 5-3c mostra-se a imagem retificada segundo o modelo, na Figura 5-3d observam-se as áreas selecionadas para extração e na Figura 5-3e mostram-se as imagens textuais extraídas. Este processo também foi usado para criar o banco de treinamento do OCR apresentado na seção 5.7.1.

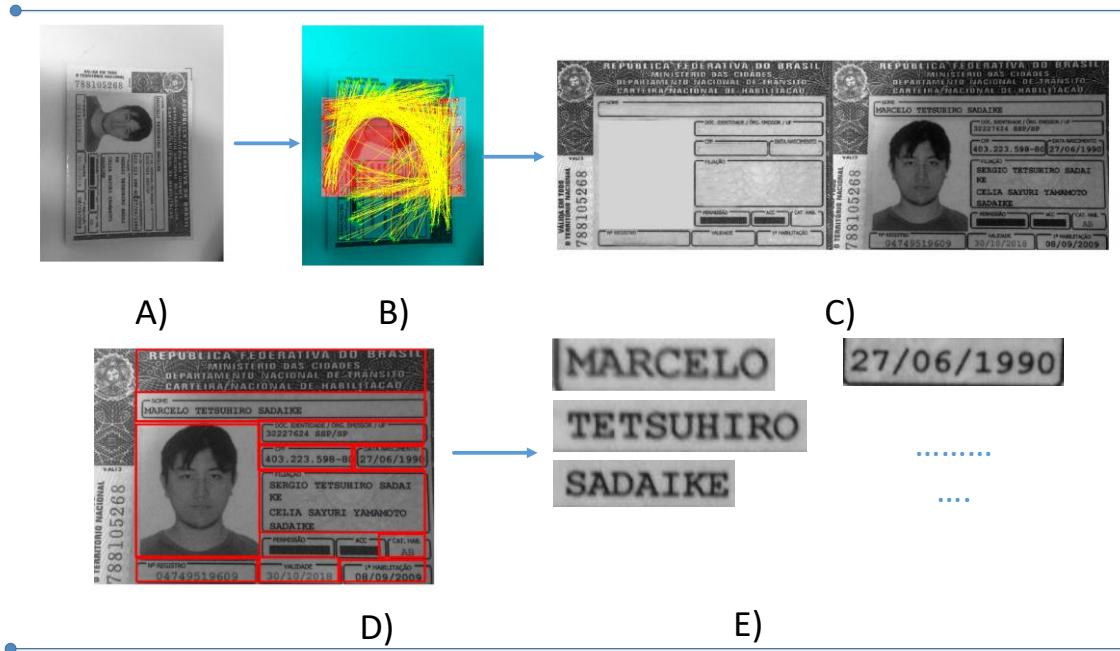


Figura 5-3: Processo automático de extração das imagens textuais usando modelo. Fonte: Autor.

## 5.2 Métodos de avaliação do desempenho do sistema

Para poder avaliar a qualidade do desempenho de nosso sistema, é necessário definir métricas para medir o quanto um texto gerado está próximo de sua transcrição correta. Uma simples aproximação foi proposta na referência (de Mello and Lins 1999), através de um estudo comparativo para analisar ferramentas comerciais de OCR. Atualmente para a avaliação do desempenho dos sistemas de localização e reconhecimento de texto as métricas mais usadas são as propostas na *Robust Reading Competition* publicadas no ICDAR 2015 (2015 2015, Burie, Chazalon et al. 2015).

No ICDAR são usadas as métricas: “revocação” e “precisão”, sendo o coeficiente de precisão igual ao número de verdadeiros positivos (i.e. o número de itens corretamente rotulados como pertencentes aos positivos), dividido pelo número total de elementos identificados como pertencentes ao conjunto positivo (i.e. a soma de verdadeiros positivos e falsos positivos, que são itens incorretamente rotulados como pertencente ao conjunto). Revocação, neste contexto, é definido como o número de verdadeiros positivos, dividido pelo número total de elementos que pertencem aos positivos (i.e. a soma de verdadeiros positivos e falsos negativos, que são itens que

não foram rotulados como pertencentes aos positivos, mas deveriam ter sido). As métricas “revocação” e “precisão” são calculadas como segue:

$$\text{revocação} = \frac{t_p}{(t_p + f_n)} \quad (\text{Equação 5-1})$$

$$\text{precisão} = \frac{t_p}{(t_p + f_p)} \quad (\text{Equação 5-2})$$

Sendo  $t_p$ ,  $f_p$ ,  $f_n$  e  $t_n$  as taxas de verdadeiros positivos (*true positive*), falsos positivos (*false positive*), falsos negativos (*false negative*) e verdadeiros negativos (*true negative*) respectivamente. Os significados de  $t_p$ ,  $f_p$ ,  $f_n$  e  $t_n$  são definidos para cada uma das situações.

Precisão é também chamada de valor preditivo positivo, enquanto revocação é conhecida como sensibilidade. Tanto precisão quanto revocação são, portanto, bases para o estudo e compreensão da medida de relevância. Por exemplo, no contexto de sistema de reconhecimento de texto, suponha que a etapa de seleção de texto identifica 70 caracteres em uma imagem contendo 100 caracteres e alguns componentes que não são caracteres. Se 60 dos 70 caracteres identificados são realmente caracteres, mas 10 são, na verdade, componentes que não são caracteres, a precisão da etapa de seleção é 60/70 enquanto a sua revocação é 60/100. Agora, sena mesma imagem, na etapa de seleção são identificados 120 caracteres, dos quais 90 são realmente caracteres, a precisão da etapa de seleção é 90/120 enquanto a sua revocação é 90/100, neste caso a revocação é maior.

Para o caso da etapa de reconhecimento, suponha que o sistema reconhece 80 caracteres em uma imagem contendo 100 caracteres. Se 50 dos 80 caracteres identificados são corretamente reconhecidos, mas 30 são incorretamente reconhecidos, a precisão do sistema é 50/80 enquanto a sua revocação é 50/100. Precisão, neste caso, é “o quanto os resultados do reconhecimento são úteis”, enquanto revocação é “o quão completos os resultados do reconhecimento estão”.

Em termos simples, um score perfeito de precisão ( razão = 1.0) significa que todos os caracteres selecionados pelo sistema de reconhecimento foram corretamente reconhecidos (mas não diz nada sobre se todos os caracteres do documento foram selecionados), enquanto um score perfeito de revocação ( razão = 1.0 ) significa que todos os caracteres do documento foram selecionados e

corretamente reconhecidos pelo sistema (mas não diz nada sobre como muitos componentes irrelevantes, que não são caracteres, também foram reconhecidos). Observe que o significado e o uso de "precisão" na área de reconhecimento de texto são um pouco diferentes da definição de precisão dentro de outros ramos da ciência.

A precisão também é usada com a revocação. As duas medidas são utilizadas em conjunto no F1 (*F-Score* ou *F-measure*) para fornecer uma única medição para o sistema, fazendo um balanceamento entre os dois conforme a equação.

$$F_1 = 2 * \frac{\text{precisão} * \text{revocação}}{\text{precisão} + \text{revocação}} \quad (\text{Equação 5-3})$$

### 5.2.1 Distância de Levenshtein

Outra métrica usada é a distância de Levenshtein (Soukoreff and MacKenzie 2001), também conhecida como distância de edição. A Distância de Levenshtein entre duas palavras é o menor número de caracteres que devem ser inseridos, apagados ou substituídos em uma das palavras para que a mesma fique igual a outra. A distância de Levenshtein é usada como base para medir a qualidade dos textos gerados pelos OCR's.

Por exemplo, da palavra “tripalium” à “trabalho” a distância de Levenshtein entre elas é:

<code>tripalium =</code> <code>tripaliu</code> <code>trapaliu</code> <code>trabaliu</code> <code>trabalhu</code> <code>trabalho</code>	<ul style="list-style-type: none"> <li>- 1 exclusão “m”</li> <li>- 1 substituição “i” por “a”</li> <li>- 1 substituição “p” por “b”</li> <li>- 1 substituição “i” por “h”</li> <li>- 1 substituição “u” por “o”</li> </ul>
---	--

Portanto, a distância de Levenshtein entre trabalho e tripalium é 5 (1 exclusão, 4 trocas e 0 inserções).

### 5.3 Avaliação da etapa de seleção de texto

Inicialmente foi avaliado o processo de retificação pretendendo avaliar o desempenho do algoritmo na determinação do ângulo de inclinação das linhas de

texto. Para tal, foram selecionadas 321 linhas de texto das 100 imagens. O erro médio obtido para a inclinação das linhas de texto foi de  $0.37^\circ$  e o erro máximo foi de  $2.8^\circ$ , os quais não afetam a etapa de seleção nem reconhecimento. Os maiores erros derivam essencialmente das linhas de texto curtas. Estes resultados demostram um bom desempenho da etapa de retificação.

No caso do processo de seleção os valores de revocação e precisão são definidos como:

$$\text{revocação} = \frac{\#\text{(caracteres corretamente selecionados)}}{\#\text{(total de caracteres existentes)}} \quad (\text{Equação 5-4})$$

$$\text{precisão} = \frac{\#\text{(caracteres corretamente selecionados)}}{\#\text{(caracteres selecionados)}} \quad (\text{Equação 5-5})$$

Na tabela é comparado o resultado do processo sem a etapa de seleção e com a etapa de seleção propostas usando o algoritmo 1 no Banco de imagens 1, avaliando as métricas apresentadas. Observa-se na Tabela 5-1 como a etapa de seleção consegue melhorar consideravelmente a precisão do sistema, selecionando apenas aqueles componentes que são texto. No processo de seleção o valor de revocação varia pouco com o uso do algoritmo 1, pois, o número de caracteres selecionados é aproximadamente igual, porém o F-Score aumenta em 43%, demonstrando um ganho significativo na seleção das regiões textuais.

Tabela 5-1: Avaliação da etapa de seleção de texto. Fonte: Autor.

<b>Imagens</b>	<b>Revocação</b>	<b>Precisão</b>	<b>F-Score</b>	<b>Método</b>
100	0.85	0.31	0.4543	Sem o processo de seleção
100	0.84	0.94	0.8927	Algoritmo 1

Das imagens do banco de imagens 1, formadas por 72% de regiões não textuais e 28 % de regiões textuais. O algoritmo através da filtragem com condições heurísticas classificou 74% de regiões como não textuais e 26% foram classificadas como texto, ver Tabela 5-2.

Tabela 5-2: Regiões detectadas no banco de imagens 1. Fonte: Autor.

Imagens	100	
Regiões detectadas	16124	
Total de regiões textuais	4635	$t_p + f_n$
Regiões textuais detectadas	4192	$t_p + f_p$
Regiões textuais corretamente detectadas	3940	$t_p$

Com o objetivo de avaliar o desempenho de cada uma das propriedades heurísticas, na Tabela 5-3 mostra-se o desempenho do número de regiões filtrados (regiões consideradas não textuais) por cada heurística.

Tabela 5-3: Desempenho em termos da classificação das regiões para as várias condições heurísticas. Fonte: Autor

Heurísticas	Número de regiões filtradas em %
Área	45
Proporção	18
Solidez	5
Excentricidade	2
Stroke Width	4
Total	74

Das regiões classificadas como não textuais (74% do total ou 11932 regiões), o 3,7% são regiões textuais ( $f_n$ ). Das regiões classificadas como textuais (26 % do total ou 4192 regiões), 6% não são regiões textuais ( $f_p$ ), sendo classificadas erradamente apenas o 6%, para um 94% de precisão.

#### 5.4 Avaliação da etapa de reconhecimento de texto

No caso do processo de reconhecimento, revocação e precisão são definidos como:

$$\text{revocação} = \frac{\#(\text{caractere/palavra corretamente reconhecida})}{\#(\text{total de caracteres/palavra existente})} \quad (\text{Equação 5-6})$$

$$\text{precisão} = \frac{\#(\text{caractere/palavra corretamente reconhecida})}{\#(\text{caractere/palavra reconhecida})} \quad (\text{Equação 5-7})$$

Na avaliação dos resultados é usado o Banco de imagens 2. Na Tabela 5-4 são comparados o resultado do processo aplicando o algoritmo 2 sobre os BBs previamente selecionados e o resultado do uso do OCR diretamente sobre os BBs sem o algoritmo 2. Observa-se como o algoritmo 2 melhora o resultado de ambas métricas, obtendo os melhores resultados para cada palavra, aumentando o *F-Score* em 12%. Os valores na Tabela 5-4 apresentam resultados em termos de caracteres e na Tabela 5-5 resultados em termos de palavras.

Tabela 5-4: Avaliação da etapa de reconhecimento de texto para caracteres.

Fonte: Autor.

Imagens	Revocação	Precisão	F-Score	Método
100	0.78	0.81	0.7947	Sem o Algoritmo 2
100	0.88	0.96	0.9182	Algoritmo 2

Tabela 5-5: Avaliação da etapa de reconhecimento de texto para palavras.

Fonte: Autor.

Palavras	Revocação	Precisão	F-Score	Método
934	0.51	0.57	0.5383	Sem o Algoritmo 2
934	0.84	0.87	0.8547	Algoritmo 2

No caso das palavras, no processo sem o algoritmo 2, os resultados por palavra ficaram abaixo dos acertos por caractere devido ao fato de que os erros podem ocorrer em palavras distintas, o universo (total de caracteres e total de palavras) de comparação é diferente, pois há um número menor de palavras. Assim um erro num caractere faz considerar a palavra como incorreta, por exemplo: a palavra consideravelmente tem 17 caracteres sendo 1 errado (caractere p), então a precisão em termos de caracteres é de 0.94, porém em termos de palavra é 0. No caso do uso do algoritmo 2, observou-se que a maioria dos erros foram causados por imagens em baixa qualidade e falta do treinamento do OCR para essa tipografia; estes erros foram

geralmente localizados na mesma palavra. Um treinamento do OCR melhora o resultado como será apresentado neste capítulo.

### **5.5 Avaliação do sistema geral.**

Na avaliação dos resultados do sistema completo é usado o banco de imagens 4. Para calcular as métricas são usados os caracteres reconhecidos no final do processo. Na tabela são comparados o resultado do processo sem usar o algoritmo 1 nem o algoritmo 2, usando somente o algoritmo 1, usando somente o algoritmo 2 e usando os dois algoritmos. Em todos os casos são usadas as etapas de pré-processamento, retificação, localização, seleção de linhas e OCR, com o objetivo de avaliar somente as contribuições dos algoritmos 1 e 2 no processo geral. Observa-se na Tabela 5-6 que sem o uso dos algoritmos o resultado final é o pior, já o algoritmo 1 melhora o resultado de ambas métricas, porém, o uso do algoritmo 2 unicamente, não melhora em alto grau os resultados pois sem a etapa de seleção ainda existe muito ruído que afeta o algoritmo. Na combinação dos dois o resultado melhora consideravelmente obtendo os melhores resultados, aumentando o *F-Score* em 32%.

Tabela 5-6: Avaliação do sistema geral. Fonte: Autor.

Imagens	Revocação	Precisão	F-Score	Método
300	0.62	0.45	0.5214	Sem algoritmos
300	0.73	0.75	0.7398	Algoritmo 1
300	0.66	0.49	0.5624	Algoritmo 2
300	0.81	0.89	0.8481	Algoritmo 1 e 2

Adicionalmente foi avaliada a seleção de linhas, pois uma seleção incorreta acrescentaria os erros no algoritmo 2. De modo a avaliar este desempenho, foram calculadas manualmente todas as linhas de texto das 300 imagens, num total de 944 linhas. Os resultados obtidos para a avaliação da seleção de linhas de texto são apresentados na Tabela 5-7.

Tabela 5-7: Desempenho em termos da seleção de linhas de texto. Fonte: Autor

	<b>Corretamente detectadas</b>	<b>Falhas</b>
Sem o algoritmo 1	745	199
Com o algoritmo 1	906	38

Sem o uso do algoritmo 1, foram detectadas 79% das linhas, já que foram classificadas erradamente como texto regiões que não correspondem a texto, unindo várias linhas, ver Figura 5-4. Este resultado justifica como a etapa de seleção afeta o resultado do reconhecimento, obtendo um baixo acerto no resultado final quando é usado somente o algoritmo 2, como explicado anteriormente. Quando é usado o algoritmo 1 foram detectadas o 96% das linhas de texto o que justifica os melhores resultados obtidos no processo após a etapa de seleção.



Figura 5-4: a,b) formação de linhas usando o processo de seleção; c,d) formação sem o processo de seleção. Fonte: Autor.

A seguir o sistema de reconhecimento de texto proposto é comparado com o trabalho de (Ryan and Hanafiah 2015), discutido nos trabalhos relacionados. (Ryan and Hanafiah 2015) apresenta um sistema de reconhecimento de IDs de Indonésia e para avaliar o sistema usa um ID com 320 caracteres e 16 linhas. Usando o mesmo ID foi realizado o processo de reconhecimento de texto proposto nesta dissertação e foram comparados os resultados. Os resultados da comparação na etapa de seleção mostram-se na Tabela 5-8, e os resultados da etapa de reconhecimento são mostrados na Tabela 5-9 . Em (Ryan and Hanafiah 2015) não são apresentados resultados em termos de precisão nem *F-score*.

Tabela 5-8: Comparação dos resultados da etapa de seleção com (Ryan and Hanafiah 2015). Fonte: Autor.

<b>Caracteres</b>	<b>Selecionados corretamente</b>	<b>Revocação</b>	<b>Método</b>
320	296	0.925	(Ryan and Hanafiah 2015)
320	315	0.984	Algoritmo 1

Tabela 5-9: Comparação dos resultados da etapa de reconhecimento com (Ryan and Hanafiah 2015). Fonte: Autor.

<b>Caracteres</b>	<b>Reconhecidos corretamente</b>	<b>Revocação</b>	<b>Método</b>
320	115	0.359	(Ryan and Hanafiah 2015)
320	274	0.856	Algoritmo 1 e 2

No estágio de segmentação, no trabalho de (Ryan and Hanafiah 2015) aproximadamente 93% de caracteres foram corretamente selecionados. Na Tabela 5-8 observa-se como nosso algoritmo melhora a etapa de seleção em aproximadamente 6%. No processo de reconhecimento o método de (Ryan and Hanafiah 2015) ainda precisa ser melhorado pois foi usado um algoritmo próprio de reconhecimento e não um OCR comercial, obtendo baixos resultados. Com nosso

algoritmo o reconhecimento não foi tão alto quanto a seleção pois a linguagem é indonésio, o qual dificultou o reconhecimento.

## 5.6 Experimentos com as heurísticas e os parâmetros

Usando o banco de imagens 3 foi realizada uma avaliação da classificação de regiões. O banco de imagens está formado somente por regiões textuais (2000 imagens de regiões textuais), sendo um total de 8266 caracteres. Cada propriedade heurística foi testada.

Por exemplo, para a proporção, inicialmente foi iniciado o valor  $p_c$  da propriedade em 0 (não seleciona nenhum caractere) e aumentado em intervalos constantes de 0.1, selecionando os componentes com valor  $< p_c$ , até selecionar todos os caracteres (das 2000 imagens de regiões textuais). O resultado mostra-se na Figura 5-5 , observa-se como para valores menores que 0.2 não são selecionados caracteres, e valores maiores que 5 não incorporam novos caracteres, portanto os caracteres encontram-se entre os valores de  $0.2 < p_c < 5$ . Experimentos similares foram realizados com as outras heurísticas.

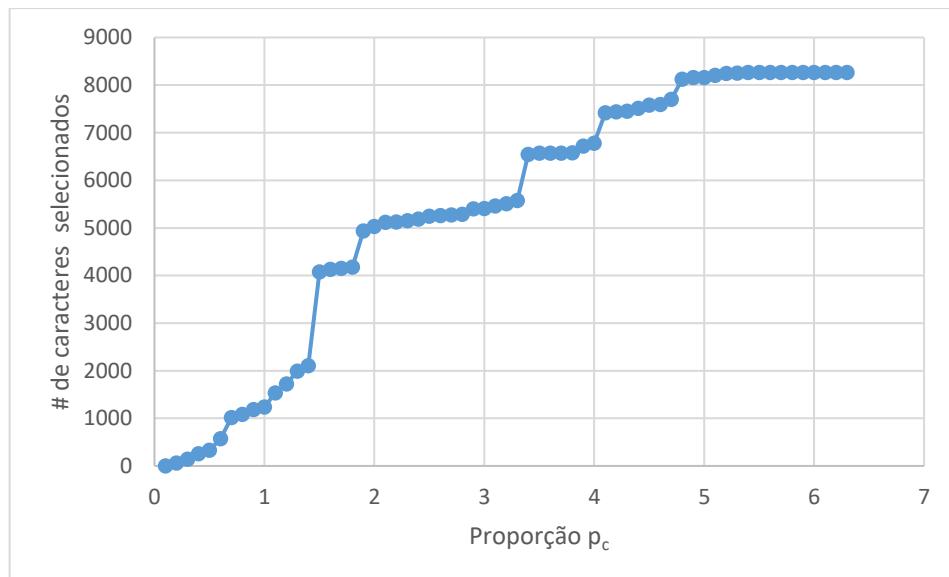


Figura 5-5: Número de caracteres selecionados usando como heurística a proporção. Fonte: Autor.

Nos experimentos de resolução, foram testadas as resoluções: 50 dpi, 100dpi, 200dpi, 300dpi e 600dpi. Na Figura 5-6 percebe-se que para as resoluções de 75 e 100dpi o percentual de acertos foi menor. Observa-se também que com a resolução de 300dpi se obtiveram os melhores resultados. Vale destacar que 300dpi é a

resolução recomendada na literatura. Adicionalmente, observou-se que quanto maior a resolução, maior a quantidade de detalhes, porém para resolução maior do que 300dpi aumenta a presença de ruídos.

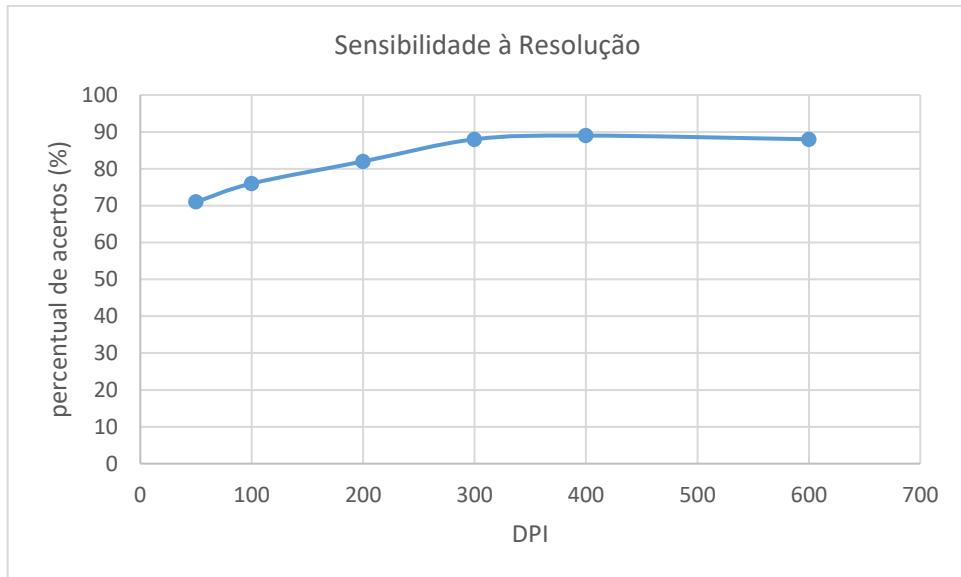


Figura 5-6: Percentual de acertos para diferentes resoluções. Fonte: Autor.

Nos experimentos de brilho, o brilho foi ajustado e normalizado em 6 intervalos e foi medido o percentual de acertos para cada intervalo. Os intervalos foram os seguintes: caso 1, sem ajuste (0 a 100%), caso 2 (normalização entre el 5 a 95% do brilho da imagem), caso 3 (normalização entre el 10 a 90%), caso 4 (normalização entre el 20 a 80%), caso 5 (normalização entre el 30 a 70%) e caso 6 (normalização entre el 40 a 60%). O maior percentual de acerto foi obtido para valores do brilho normalizados entre o 10 e 90 % do brilho da imagem.

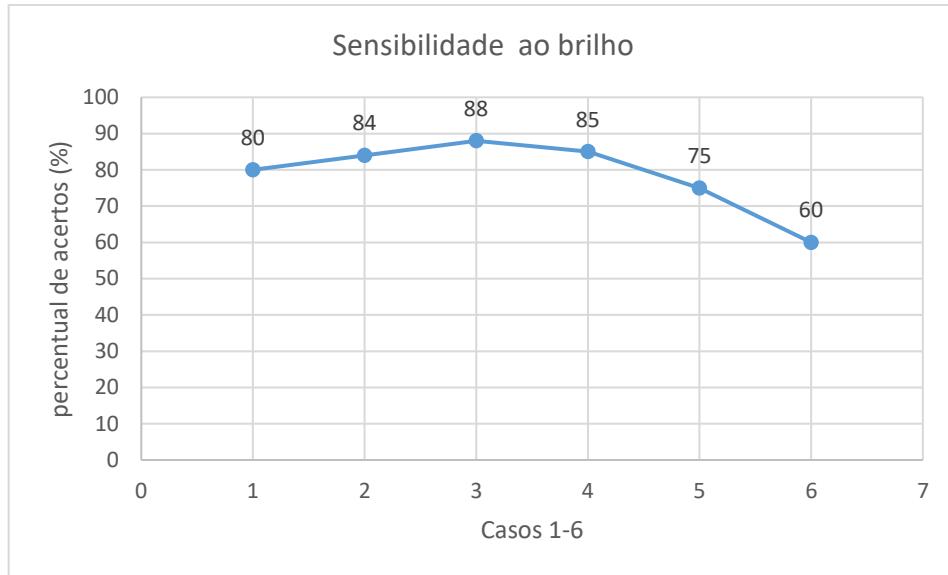


Figura 5-7: Percentual de acertos para variação de brilho. Fonte: Autor.

A seguir são avaliados o número de iterações e o valor da acurácia do OCR usado como padrão no algoritmo 2. Foram realizados testes com 1,5,10,15 e 20 iterações e calculadas as métricas para cada caso. Também foi calculada a média das iterações realizadas até alcançar o valor de acurácia desejado ( $Ac^*$ ). Os resultados mostram-se na Tabela 5-10, observa-se que o desempenho do sistema mantém-se aproximadamente constante após 5 iterações. Na Tabela 5-11 mostra-se o resultado das iterações para  $Ac^*=95\%$ , neste caso o máximo de iterações sempre é alcançado e o número de iterações é usado como critério de parada do algoritmo 2 ( $n > n^*$ ) pois é difícil de alcançar um valor elevado de acurácia ( $Ac^*=95\%$ ), para todas as palavras.

Tabela 5-10: Número de iterações:  $n^*$  para Acurácia do OCR desejado  $Ac^*=85\%$ . Fonte: Autor.

<b><math>n^*</math></b>	<b>Revocação</b>	<b>Precisão</b>	<b>Média das Iterações realizadas até <math>Ac &gt; Ac^* \text{ ou } n &gt; n^*</math></b>	<b>F-Score</b>
1	0.75	0.79	1	0.7694
5	0.81	0.89	5	0.8481
10	0.81	0.90	9	0.8526
15	0.82	0.90	12	0.8581
20	0.82	0.90	16	0.8581

Tabela 5-11: Número de iterações:  $n^*$  para Acurácia do OCR desejado  $Ac^*=95\%$ . Fonte: Autor.

<b><math>n^*</math></b>	<b>Revocação</b>	<b>Precisão</b>	<b>Média das Iterações realizadas ate</b>	<b>F-Score</b>
			<b><math>Ac &gt; Ac^* \text{ ou } n &gt; n^*</math></b>	
1	0.75	0.79	1	0.7694
5	0.81	0.89	5	0.8481
10	0.81	0.90	10	0.8526
15	0.82	0.90	15	0.8581
20	0.82	0.90	20	0.8581

O algoritmo 2 foi usando na Figura 5-8a com os BBs previamente obtidos, foram geradas as imagens  $Q_{f1t1}$ ,  $Q_{f2t1}$ ,  $Q_{f3t1}$ ,  $Q_{f4t1}$ ,  $Q_{f1t2}$ ,  $Q_{f2t2}$ ,  $Q_{f3t2}$ ,  $Q_{f4t2}$ . a partir da Figura 5-8a ( segundo o explicado na seção 4.3.2) o resultado do reconhecimento de texto em cada linha para o exemplo nas imagens  $Q_{f1t1}$ ,  $Q_{f2t1}$ , e  $Q_{f4t2}$  é mostrado na Tabela 5-12 ; é importante observar como as transformações  $Q_{f1t1}$ ,  $Q_{f2t1}$ , e  $Q_{f4t2}$  da Figura 5-8a geram diferentes textos após o OCR. O resultado após a seleção das melhores palavras é mostrado na Tabela 5-13, observando-se as palavras selecionadas para cada linha com os valores de acurácia do OCR para cada uma.



Figura 5-8: Imagem original e a imagem com as linhas de texto selecionadas.  
Fonte: Autor.

Tabela 5-12: Reconhecimento para cada linha nas imagens  $Q_{f1t1}$ ,  $Q_{f2t1}$ , e  $Q_{f4t2}$ .

Fonte: Autor.

<b>Q<sub>f1t1</sub></b>	<b>Q<sub>f2t1</sub></b>	<b>...</b>	<b>Q<sub>f4t2</sub></b>
'Mmasréam DA FAZENDA'	'MINISTÉRID DA FAZENDA'		'Mmasréam DA FAZENDA'
'Secretaria da Receita Federal'	'Secretaria da Receita Federal'		'Secretaria da Receita Federal'
'CADASTRO DE PESSOAS FÍSICAS'	'CADASTRO DE PESSOAS FÍSICAS'		'CADASTRO DE PESSOAS FÍSICAS'
'CPF'	'CPF'		'CPF'
'Nome'	'Nome'		'Nome'
'DENIS FRANCO'	'DENIS FRANCO'		'DENIS FRANCO'
'Nº de Inscrição'	'Nº de Inscrição'		'Nº de Inscrição'
'Data do Nascimento'	'Data do Nascimento'		'Data do Nascimento'
	'.'		'.'
'875224855-72'	'875224855-72'		'875224855-72'
'20/12/80'	'20/12/80'		'20/12/80'

Tabela 5-13: Seleção das melhores palavras. Fonte: Autor.

<b>Melhores palavras</b>	<b>Valores de acurácia do OCR</b>
'MINISTÉRIO DA FAZENDA'	[0.72754967;0.88170958;0.80169231]
'Secretaria da Receita Federal'	[0.78989393;0.86827391;0.81884825;0.79959422]
'CADASTRO DE PESSOAS FÍSICAS'	[0.92267263;0.94256222;0.92011440;0.92165577]
'CPF'	0.93232054
'Nome'	0.85889524
'DENIS FRANCO'	[0.90120322;0.88545120]
'Nº de Inscrição'	[0.92135406;0.84522206;0.85673368]

'Data do Nascimento'	[0.84430969;0.88545072;0.80942792]
'.'	0.77393597
'875224855-72'	0.90553534
'20/12/80'	0.85582471

Do resultado da Tabela 5-13, tem-se o melhor reconhecimento das palavras para cada linha, e os melhores valores de acurácia do OCR, demonstrando a eficiência do algoritmo desenvolvido.

Finalmente, nos experimentos realizados com diferentes valores dos parâmetros, diversos tipos de erros foram encontrados na saída do OCR e são resumidos a seguir.

Para variação do contraste:

- Substituição de um caractere por outro (como “e” por “c”);
- Substituição de um caractere por mais de um (como em “m” por “r n”);
- Substituição de mais de um caractere por apenas um (como em “r n” por “m”);

Para variação do brilho

- Perda de caracteres (supressão);
- Perda completa de linhas de texto;

Para variação do limiar

- Junção de palavras sem perda de caracteres (supressão de espaços em branco);
- Junção de palavras com perda de caracteres;
- Inserção de caracteres;

A substituição de um caractere por outro é o erro mais comum, seguido pela inclusão de caracteres, geralmente, devido a ruídos no documento original. A junção de duas palavras com ou sem perda de algum caractere é também um erro comum encontrado em altos valores de brilho.

## 5.7 Realização de uma prova de conceito do sistema em um ambiente real

Foi realizada uma prova de conceito do sistema em um ambiente. O processo é implementado e testado usando o MATLAB e é aproveitada a computação na nuvem para permitir que dispositivos de baixo poder computacional possam aproveitar a nuvem para o processamento intensivo que requer o sistema proposto. Os documentos de teste usados são documentos de identificação do banco de imagens 5. A Figura 5-9 mostra um diagrama do sistema. Consiste de duas fases: na fase 1, a imagem do ID é capturada com um celular e enviada para o servidor, e na fase 2, um servidor Java processa a imagem para o reconhecimento do texto. Todos os algoritmos de processamento de imagem são implementados e depurados no MATLAB. Usando o compilador SDK do MATLAB, é gerado um pacote java (code.jar) com todas as funções do sistema de reconhecimento de texto. Finalmente, este pacote é executado pelo servidor.

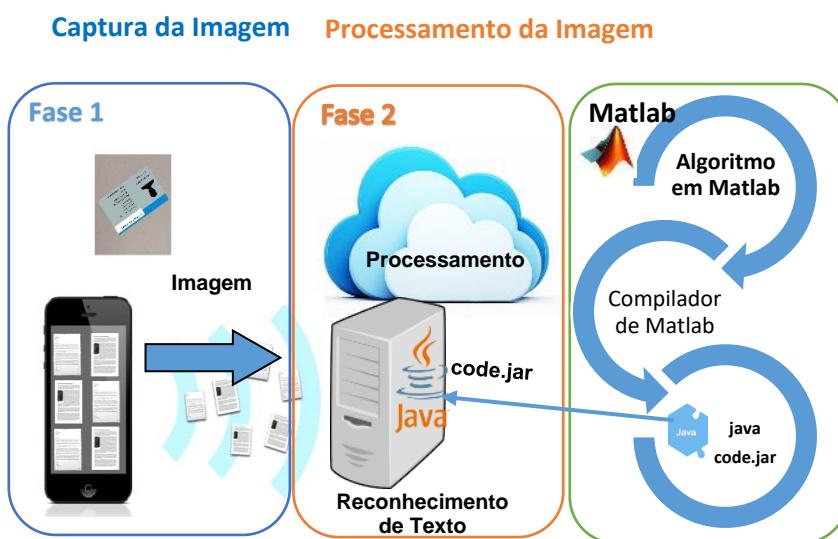


Figura 5-9: Diagrama do sistema de teste. Fonte: Autor.

### 5.7.1 Experimentos e resultados da prova de conceito

A Figura 5-10, mostra os resultados do processo completo numa imagem de cartão de visita. Imagens similares são simples e fáceis de reconhecer. Neste caso são detectadas 100% das caixas delimitadoras das palavras, e 100% de precisão é obtido após a realização do OCR.



Figura 5-10: Resultado do reconhecimento das palavras numa imagem de teste. 12 linhas (100% do documento) com um 100% de precisão do OCR. Fonte: Autor.

O sistema foi avaliado com o banco de imagens 5. São calculadas: a) as palavras totais nos IDs, processo feito manualmente, b) todas as palavras encontradas após a aplicação do OCR, c) Palavras corretamente reconhecida, d) Precisão, e) Revocação f) *F-score*.

Foram calculados os dados em quatro casos diferentes Tabela 5-14):

1. OCR sobre a imagem original, sem retificação
2. OCR sobre a imagem retificada.
3. OCR sobre a imagem com a caixa delimitadora.
4. Sistema usando o algoritmo 1 e 2.

Tabela 5-14: Comparativa dos resultados de saída da função OCR. Fonte: Autor.

Caso de Estudo	1-OCR sobre a imagem original, sem retificação	2-OCR sobre a imagem retificada	3-OCR sobre a imagem com a caixa delimitadora.	4-Sistema usando o algoritmo 1 e 2.
Palavras totais nos IDs	1100	1100	1100	1100
Todas as palavras encontradas após a aplicação do OCR	63	487	1034	1034
Palavras corretamente reconhecida	1	344	871	963
Precisão	0.0158	0.7063	0.8423	0.9313
Revocação	0.0009	0.3127	0.7918	0.8754
<i>F-score</i>	0.0017	0.4335	0.8163	0.9025

No caso 1, a imagem não for retificada; portanto, os resultados obtidos apresentam pouca precisão porque OCR Tesseract funciona com palavras em sentido horizontal. No caso 2, a imagem é corrigida, mas a caixa delimitadora não é extraída, assim, não se conseguem os resultados desejados. Para o caso 3, o OCR é aplicado sobre apenas uma imagem e produz alguns erros. Para o caso 4, o OCR é melhorado usando o Algoritmo 2; e os melhores resultados são obtidos com uma precisão de 93%, e *F-score* de 0.9.

Exemplo da melhora observa-se na Figura 5-11. Antes de usar o algoritmo 2 na primeira palavra, 2015 foi lido, para o segundo foi lido Maincula. Depois da escolha das melhores palavras foram obtidos os resultados desejados (2015 e Matrícula).



Figura 5-11: Palavras corretamente lidas após o uso do algoritmo 2. Fonte: Autor.

Porém ainda apresentam erros no OCR, exemplo de palavras incorretas são mostradas na Figura 5-12.



Figura 5-12: Palavras incorretamente lidas. Fonte: Autor.

O erro está relacionado com a complexidade do fundo e a fonte, além da falta de treinamento do algoritmo OCR para a tipografia específica e a baixa resolução da imagem.

Foi realizado um teste para mostrar a melhoria do processo com treinamento. Dentro do MATLAB, foi utilizada o Toolbox OCR Trainer, para realizar os treinamentos dos documentos. Foram usados 80 documentos para treinamento e 60 documentos para testar os resultados do treinamento (no total 370 palavras e 3520 caracteres). Foram recortados das imagens os campos de texto e usados para fazer o treinamento, como se mostra na Figura 5-13. Os dados para o treinamento foram extraídos dos 80 documentos automaticamente usando o processo com modelo explicado no início do capítulo.

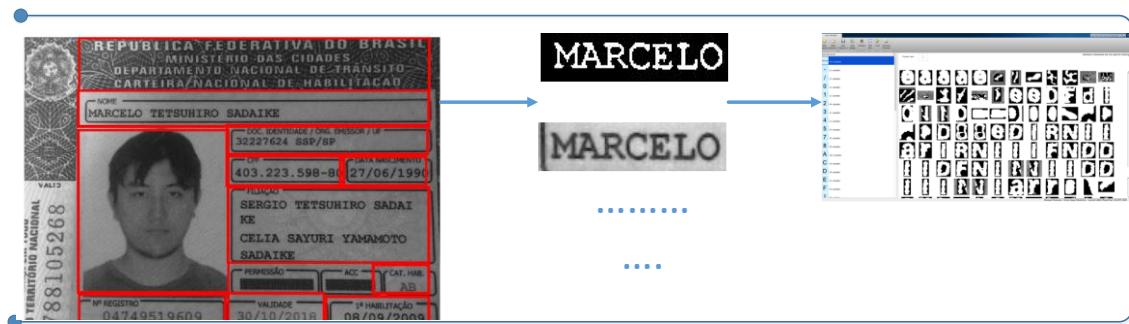


Figura 5-13: Treinamento do OCR. Fonte: Autor.

Finalmente, foram comparados os resultados do sistema vistos anteriormente com os resultados com treinamento. Também foram calculadas e comparadas as distâncias Levenshtein para todas as palavras dos documentos. Ver Tabela 5-15.

Adicionalmente foi realizada a comparação tendo em conta o tipo de dados : texto ou número, os resultados mostram-se na Figura 5-14. Observasse como os resultados para o caso com treinamento foram melhores em todas as comparações.

Tabela 5-15: Distancia Levenshtein do resultado do OCR com e sem treinamento. Fonte: Autor.

	<b>Distancia Levenshtein</b>
Sem treinamento	380
Com treinamento	200

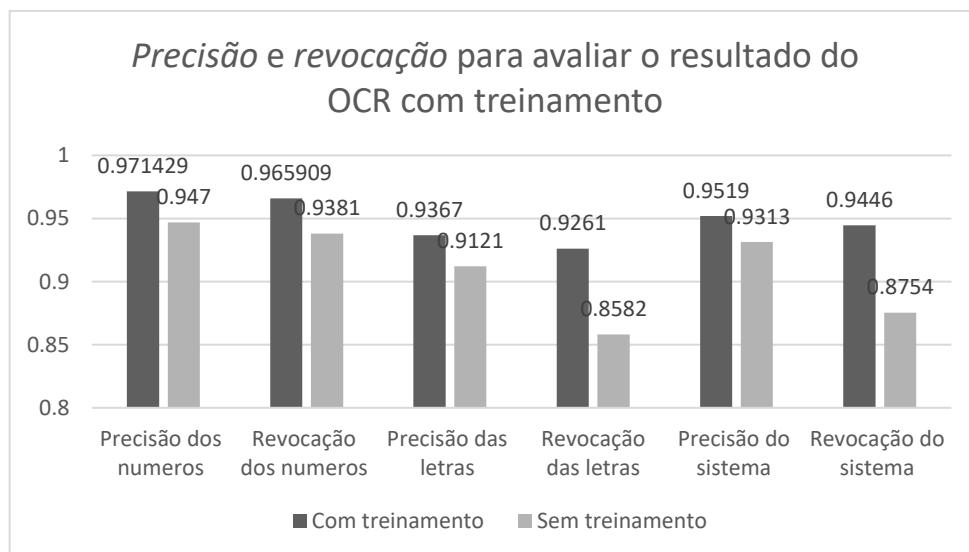


Figura 5-14: Resultados do OCR com e sem treinamento para letras e números.  
Fonte: Autor.

Também foi realizado um estudo de fontes para treinamento, demonstrando que melhora o resultado, no caso da carteira de motorista os melhores resultados foram obtidos com a fontes: Courier Prime e Courier EF Courier B EF. Para o CPF os melhores resultados são com as fontes: ACumin PRo Bold, Ultimte Serial Bold. Visto que o treinamento melhora claramente o desempenho do sistema recomenda-se um estudo mais a fundo das fontes, e um treinamento com um maior número de imagens.

## 5.8 Análise de desempenho

A análise de desempenho está relacionada à quantidade de tempo que os algoritmos 1 e 2 precisam na execução. O número de pixels da imagem é um fator determinante neste tipo de análise, aumentando a complexidade proporcionalmente.

Ver Figura 5-15, observa-se como o aumento do tempo de processamento com o número de pixels é linear e o tempo de execução do algoritmo 2 sempre é maior. Na análise de desempenho foi usado o MATLAB R2016a e um computador com processador Intel core i3 de 1,8 GHz, 6 Gb de RAM e sistema operacional Windows7.

Para o cálculo do tempo foi usada a função timeit do MATLAB, que chama a função especificada várias vezes e retorna a mediana das medições. A função timeit controla a função a ser medida e retorna o tempo de execução típico, em segundos.

Para imagens com 3000x1912, 1439x1054 e 196x138 pixels foi calculado o tempo de execução dos algoritmos 1 e 2 e mostram-se na Tabela 5-16.

Tabela 5-16: Tempo de execução dos algoritmos 1 e 2, em segundos. Fonte: Autor.

Pixels da imagem	3000 × 1912	1439x1054	196 × 138
Tempo de execução do algoritmo 1 (segundos)	20.935446	6.068801	1.549480
Tempo de execução do algoritmo 2 (segundos)	29.342351	10.18003	2.399891

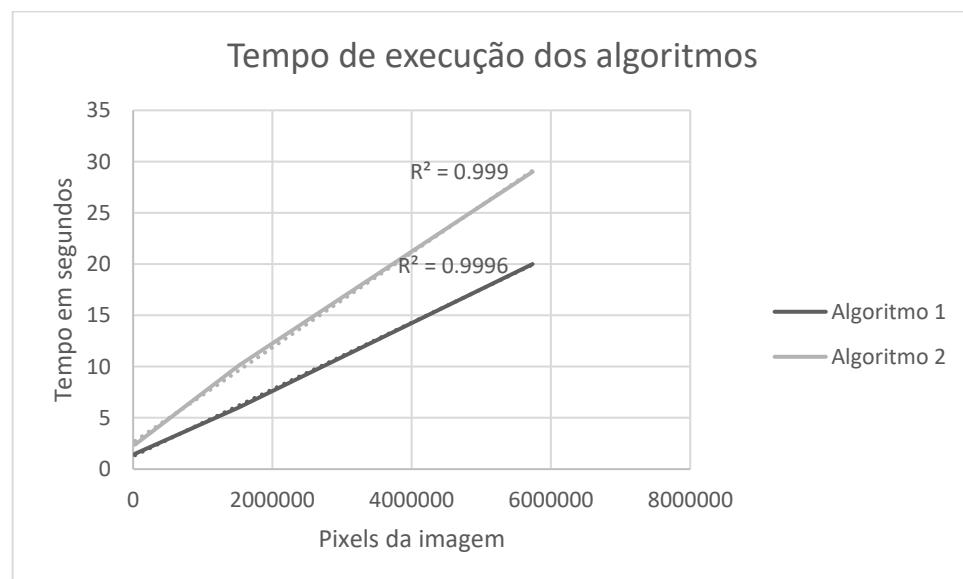


Figura 5-15: Tempo de execução dos algoritmos 1 e 2, em segundos. Fonte: Autor.

Para as imagens do banco de imagens 1, o tempo médio de processamento do algoritmo 1 foi de 17 segundos, nas imagens do banco 2 o tempo médio de processamento do algoritmo 2 foi de 28 segundos e para as imagens do banco 4 o tempo médio de processamento para todo o sistema foi de 43 segundos. Portanto, o sistema proposto tem um tempo médio de 43 segundos para cada documento de identificação.

Tendo em conta que com o aumento da resolução aumenta o tempo de processamento, serão consideradas e estudadas em trabalhos futuros algumas estratégias e critérios práticos para melhorar o desempenho do sistema:

- O sistema orientará ao usuário na captura do documento para obter uma imagem de qualidade e com a orientação certa.
- Será desenvolvido um algoritmo para detectar a qualidade inicial do documento, se a imagem não possui as condições mínimas para ser processada será solicitada uma nova imagem para processar. Por tanto, serão definidos índices que caracterizem a qualidade da imagem.
- Será desenvolvido um processo automático de reconhecimento e criação de modelos, de modo que dada uma imagem de um documento, esse documento será analisado pelo sistema e se houver um modelo do documento, será realizado o processo com modelo (processo mais rápido e eficiente), caso contrário será realizado o processo proposto neste trabalho. Uma vez processado o documento, com a informação obtida será criado um modelo novo para atualizar o banco de modelos, assim quando o sistema receber um documento de esse tipo será realizado o processo com modelo.
- Adicionalmente pretendemos otimizar algumas funções e implementar processamento paralelo com o objetivo de diminuir o tempo de processamento total.

## 5.9 Considerações finais

Ao longo deste capítulo foi testado o sistema proposto utilizando vários tipos de imagens e banco de imagens. Foram efetuados testes da etapa de seleção, reconhecimento e o sistema geral usando as métricas de revocação e precisão.

As imagens de texto usadas apresentam uma grande diversidade de fontes, estilos, tamanhos e orientações. Essas características tornam mais difícil a classificação pelo algoritmo de seleção proposto e o reconhecimento, devido à sua complexidade. Desta forma, os resultados obtidos exibem alguns erros. Contudo os algoritmos propostos apresentam resultados satisfatórios em ambas métricas, quando comparado com os sistemas que foram descritos nos trabalhos relacionados. Os parâmetros e heurísticas propostas foram validados comprovando seu desempenho. Foi ainda realizado um teste do sistema em um ambiente real e realizados experimentos donde o sistema apresentou uma precisão do 93%. Adicionalmente foi mostrado um novo cenário de aplicação dos algoritmos propostos que permanece aberto a trabalhos futuros.

Para concluir pode-se considerar que o algoritmo de seleção proposto é bastante robusto pois seleciona 84% do texto existente nas imagens, 94% do qual é corretamente classificado como texto. O algoritmo de reconhecimento proposto reconhece 88% do texto existente nas imagens, 96% do qual é corretamente reconhecido. Finalmente, o sistema geral apresenta uma precisão do 89 % e um *F-score* de 84% mostrando resultados competitivos.

## 6 Conclusões e trabalhos futuros

Nesta dissertação, foi apresentado um sistema automático de reconhecimento de texto em imagens de IDs, usando uma arquitetura que integra eficientemente os diferentes algoritmos de reconhecimento de imagens e na etapa final um método iterativo para melhorar o resultado do OCR. Foram detalhadas as principais abordagens empregadas no reconhecimento de texto em imagens de documentos. O sistema foi avaliado em diversos conjuntos de imagens e comparado com outros trabalhos analisando os resultados.

O método de localização proposto explora a característica de contraste existente entre os pixels de contorno e o plano de fundo nos caracteres legíveis. Os resultados demonstram 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. A etapa de seleção proposta neste trabalho desenvolveu-se mediante a busca de heurísticas, extraídas da imagem, capazes de classificar as regiões localizadas como textuais e não-textuais e foi desenvolvido também, aproveitando as heurísticas e a informação da largura do traço de cada caractere um classificador binário que melhora o desempenho do sistema.

Em especial discutimos o uso de algumas estratégias de pré-processamento antes da aplicação do OCR para a melhoria da etapa de reconhecimento. Diferente de outros trabalhos foi implementado um método iterativo que facilita um reconhecimento preciso das palavras. A abordagem proposta foi aplicada e comparada com os trabalhos mostrando resultados competitivos, apresentando melhorias na implementação das diferentes etapas do reconhecimento de texto.

O algoritmo de seleção proposto obteve uma revocação de 84% e 94% de precisão. O algoritmo de reconhecimento reconhece 88% do texto existente nas imagens, 96% do qual é corretamente reconhecido. Finalmente, o sistema geral apresenta uma precisão do 89 % e um *F-score* de 84% mostrando resultados competitivos. Adicionalmente, foram apresentados resultados do desempenho das heurísticas usadas na etapa de seleção e do uso do sistema em um ambiente real. Também apresentam-se outras aplicações do trabalho e resultados obtidos para outros cenários, como reconhecimento de texto para imagens da Web.

Finalmente, observamos que localizar e reconhecer texto em imagens de IDs é apenas uma classe de problema de análise de documentos que têm implicações importantes para a recuperação de informações de um cidadão. O grau em que as técnicas existentes podem ser adaptadas e melhoradas usando métodos completamente novos é uma questão aberta.

## 6.1 Próximas etapas

Embora os resultados da extração de texto de imagens de IDs sejam promissores, muito resta, ainda, para ser feito nesta área. Os estágios de localização, seleção e reconhecimento de texto em imagens podem ser melhorados por várias maneiras. Em primeiro lugar, observamos alguns problemas de desempenho no sistema de reconhecimento de palavras quando a etapa de seleção erra na classificação das regiões textuais, e também quando as caixas delimitadoras não foram corretamente estimadas. Uma vez que o sistema de reconhecimento de palavras depende das etapas anteriores, parte da diminuição observada no desempenho no sistema completo é devido a erros no processo de seleção. Outro fator foi o fato de que a etapa de localização do sistema em algumas imagens não localizou corretamente o texto, assim se na etapa de localização não é detectado um caractere, é impossível que o sistema solucione a falha na localização. Assim, melhorar a etapa de localização certamente melhoraria o desempenho do sistema.

Uma possível direção para o trabalho futuro, então, é continuar melhorando o desempenho do sistema de localização e seleção de texto, usando classificadores mais robustos. Por exemplo, criando e treinando um classificador baseado em CNN, ou um classificador neuro-fuzzy, tornando o sistema mais seletivo.

Outra limitação do sistema de texto foi o fato do sistema considerar encontrar linhas de texto numa direção só, existem imagens nas quais o texto pode ser encontrado em várias direções. Embora esta tenha sido uma suposição razoável em muitos casos e tenha funcionado bem na prática, foi, no entanto, ainda uma limitação em nosso sistema. Essa restrição tornou-se particularmente problemática nos casos em que existia texto na direção horizontal e na direção vertical, portanto, foi reconhecido somente a direção na qual existia a maior quantidade de texto. Na outra direção o detector de texto simplesmente não conseguiu localizar os caracteres. Assim, a generalização do sistema para que ele seja capaz de lidar com esses casos é outra direção possível para o trabalho futuro.

Para a seleção do texto, pretendemos incorporar uma abordagem multi-resolução para extrair os caracteres que são muito pequenos ou muito grandes; e usar métodos baseados em aprendizagem de máquina, como SVM, para ajudar a remover as regiões que não sejam de texto. No reconhecimento de texto, nos focaremos em novos métodos de pré-processamento e segmentação de regiões de texto para extrair texto de imagens com fundo complexo, também será implementado um novo algoritmo que integre a arquitetura proposta e métodos específicos de super-resolução para imagens com texto, de forma a tornar o sistema mais ágil e flexível.

O resultado do trabalho está sendo usado em outras aplicações como no reconhecimento de texto em páginas Web, também é usado como entrada para um sistema de análises semânticas de documentos de identificação e de um processo automático de extração de informação de documentos de identificação e criação de modelos.

Para concluir, espero que a descrição acima forneça um senso de direção para futuras pesquisas na área de reconhecimento de texto. Deseja-se que estas sugestões possam futuramente vir a ser implementadas, contribuindo assim para mais um passo no desenvolvimento de sistemas de reconhecimento de texto mais potentes, flexíveis e robustos. Finalmente, espero também que esta pesquisa estimule e inspire novos trabalhos nesta área emocionante.

## Publicações

- Valiente, R., M. T. Sadaike, J. C. Gutiérrez, D. F. Soriano, G. Bressan and W. V. Ruggiero (2016). A process for text recognition of generic identification documents over cloud computing. Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- Gutiérrez, J. C., R. Valiente, M. T. Sadaike, D. F. Soriano, G. Bressan and W. V. Ruggiero (2017). Mechanism for Structuring the Data from a Generic Identity Document Image using Semantic Analysis. Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web. Gramado, RS, Brazil, ACM: 213-216.
- Valiente, R. and G. Bressan (2016). Text recognition improvement using a novel iterative method and super-resolution. V Workshop de Pós-Graduação Engenharia de Computação, São Paulo: PCS/POLI/USP.
- Valiente, R., J. C. Gutiérrez, M. T. Sadaike and G. Bressan (2017). Automatic Text Recognition in Web Images. Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web. Gramado, RS, Brazil, ACM: 241-244.

## Referências

- 2015, I. (2015). "Robus Reading Competition , Task 4. End to End." Retrieved 01/2017, 2017, from <http://rrc.cvc.uab.es/?ch=1&com=evaluation>.
- Alves, W. A. L. and R. F. Hashimoto (2010). Text Regions Extracted from Scene Images by Ultimate Attribute Opening and Decision Tree Classification. 2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images.
- Anagnostopoulos, C.-N. E., I. E. Anagnostopoulos, I. D. Psoroulas, V. Loumos and E. Kayafas (2008). "License plate recognition from still images and video sequences: A survey." IEEE Transactions on intelligent transportation systems **9**(3): 377-391.
- Antonacopoulos, A., D. Karatzas and J. O. Lopez (2001). Accessing textual information embedded in internet images. Proceedings of SPIE Internet Imaging II.
- Arth, C., F. Limberger and H. Bischof (2007). Real-time license plate recognition on an embedded DSP-platform. 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE.
- Bay, H., A. Ess, T. Tuytelaars and L. V. Gool (2008). "Speeded-Up Robust Features (SURF)." Comput. Vis. Image Underst. **110**(3): 346-359.
- Biemann, C. and A. Mehler (2014). Text Mining, Springer.
- Bovik, A. C. (2009). The essential guide to image processing, Academic Press.
- Brunelli, R. (2009). "Template Matching Techniques in Computer Vision."
- Burie, J., J. Chazalon, M. Coustaty, S. Eskenazi, M. M. Luqman, M. Mehri, N. Nayef, J. Ogier, S. Prum, M. Rusi, x00F and ol (2015). ICDAR2015 competition on smartphone document capture and OCR (SmartDoc). Document Analysis and Recognition (ICDAR), 2015 13th International Conference on.
- Canny, J. (1986). "A computational approach to edge detection." IEEE Transactions on pattern analysis and machine intelligence(6): 679-698.
- Chabchoub, F., Y. Kessentini, S. Kanoun, V. Eglin and F. Lebourgeois (2016). SmartATID: A Mobile Captured Arabic Text Images Dataset for Multi-purpose Recognition Tasks. 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), IEEE: 120-125.
- Chandabenmohanbhai, C., Atulpatel, Smt, Chandabenmohanbhai and D. Chandabenmohanbhai (2012). "Optical Character Recognition by Open Source OCR

Tool Tesseract: A Case Study." International Journal of Computer Applications: 975-8887.

Chang, I. W. a. H.-C. (2013). "Signboard Optical Character Recognition."

Chen, D. (2003). TEXT DETECTION AND RECOGNITION IN IMAGES AND VIDEO SEQUENCES. Doctor, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE.

Chen, D., H. Bourlard and J.-P. Thiran (2001). Text identification in complex background using SVM. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, IEEE.

Chen, H. Z., S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, B. Girod and ieee (2011). ROBUST TEXT DETECTION IN NATURAL IMAGES WITH EDGE-ENHANCED MAXIMALLY STABLE EXTREMAL REGIONS. 2011 18th Ieee International Conference on Image Processing. New York, ieee.

Chen, K., F. Yin and C.-L. Liu (2016). Effective Candidate Component Extraction for Text Localization in Born-Digital Images by Combining Text Contours and Stroke Interior Regions. 2016 12th IAPR Workshop on Document Analysis Systems (DAS), IEEE: 352-357.

da Conceição Palma, D. M. (2004). Extracção automática de texto em sequências de vídeo, INSTITUTO SUPERIOR TÉCNICO.

Dalal, N. and B. Triggs (2005). Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), IEEE.

de las Heras, L.-P., O. R. Terrades, J. Llados, D. Fernandez-Mota and C. Canero (2015). Use case visual Bag-of-Words techniques for camera based identity document classification. 2015 13th International Conference on Document Analysis and Recognition (ICDAR), IEEE: 721-725.

de Mello, C. A. and R. D. Lins (1999). A comparative study on ocr tools. Vision Interface, Citeseer.

Dong, C., C. C. Loy, K. He and X. Tang (2016). "Image Super-Resolution Using Deep Convolutional Networks." IEEE Trans. Pattern Anal. Mach. Intell. **38**: 295-307.

Dougherty, E. R. and R. A. Lotufo (2003). Hands-on morphological image processing, SPIE press.

Duda, R. O. and P. E. Hart (1972). "Use of the Hough transformation to detect lines and curves in pictures." Communications of the ACM **15**(1): 11-15.

Epshtain, B., E. Ofek and Y. Wexler (2010). Detecting text in natural scenes with stroke width transform. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE: 2963-2970.

Epshtain, B., E. Ofek, Y. Wexler and Ieee (2010). Detecting Text in Natural Scenes with Stroke Width Transform. 2010 Ieee Conference on Computer Vision and Pattern Recognition. Los Alamitos, Ieee Computer Soc: 2963-2970.

Gonçalves, G. R., S. P. G. da Silva, D. Menotti and W. R. Shwartz (2016). "Benchmark for license plate character segmentation." Journal of Electronic Imaging **25**(5): 053034-053034.

Gonzalez, A. and L. M. Bergasa (2013). "A text reading algorithm for natural images." Image and Vision Computing **31**(3): 255-274.

Gonzalez, A., L. M. Bergasa, J. J. Yebes, S. Bronte and Ieee (2012). Text Location in Complex Images. 2012 21st International Conference on Pattern Recognition. New York, Ieee: 617-620.

Gonzalez, R. C. and R. E. Woods (2008). Digital image processing. Upper Saddle River, NJ, Pearson/Prentice Hall.

Gonzalez, R. C. and R. E. R. E. Woods (2008). "Digital image processing." 954.

Gutiérrez, J. C., R. Valiente, M. T. Sadaike, D. F. Soriano, G. Bressan and W. V. Ruggiero (2017). Mechanism for Structuring the Data from a Generic Identity Document Image using Semantic Analysis. Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web. Gramado, RS, Brazil, ACM: 213-216.

Hanif, S. M. and L. Prevost (2009). Text detection and localization in complex scene images using constrained adaboost algorithm. 2009 10th International Conference on Document Analysis and Recognition, IEEE.

He, J., Q. Do, A. C. Downton and J. Kim (2005). A comparison of binarization methods for historical archive documents. Eighth International Conference on Document Analysis and Recognition (ICDAR'05), IEEE.

Heliński, M., M. Kmiecik and T. Parkoła (2012). "Report on the comparison of Tesseract and ABBYY FineReader OCR engines."

Heras, L. P. d. I., O. R. Terrades, J. Lladós, D. Fernández-Mota and C. Cañero (2015). Use case visual Bag-of-Words techniques for camera based identity document classification. 2015 13th International Conference on Document Analysis and Recognition (ICDAR).

Hough, P. V. (1962). Method and means for recognizing complex patterns.

Huang, W., D. He, X. Yang, Z. Zhou, D. Kifer and C. L. Giles (2016). Detecting Arbitrary Oriented Text in the Wild with a Visual Attention Model. Proceedings of the 2016 ACM on Multimedia Conference - MM '16. New York, New York, USA, ACM Press: 551-555.

Islam, M. R., C. Mondal, M. K. Azam and A. S. M. J. Islam (2016). Text detection and recognition using enhanced MSER detection and a novel OCR technique. 2016 5th International Conference on Informatics, Electronics and Vision (ICIEV), IEEE: 15-20.

Jaderberg, M., K. Simonyan, A. Vedaldi and A. Zisserman (2016). "Reading Text in the Wild with Convolutional Neural Networks." International Journal of Computer Vision **116**(1): 1-20.

Jain, A. K. and S. K. Bhattacharjee (1992). "Address block location on envelopes using Gabor filters." Pattern Recognition **25**(12): 1459-1477.

Jain, A. K. and B. Yu (1998). "Automatic text location in images and video frames." Pattern Recognition **31**(12): 2055-2076.

Jung, K., K. I. Kim and A. K. Jain (2004). "Text information extraction in images and video: a survey." Pattern recognition **37**(5): 977-997.

Jung, K., K. I. Kim and A. K. Jain (2004). "Text information extraction in images and video: a survey." Pattern Recognition **37**: 977-997.

Li, Y., H. C. Lu and Ieee (2012). Scene Text Detection via Stroke Width. 2012 21st International Conference on Pattern Recognition. New York, Ieee: 681-684.

Liu, C., C. Yang, X. Ding and J. Fan (2011). Text extraction from web images. IS&T/SPIE Electronic Imaging, International Society for Optics and Photonics.

Lopresti, D. and J. Zhou (2000). "Locating and recognizing text in WWW images." Information Retrieval **2**(2-3): 177-206.

Lowe, D. G. (2004). "Distinctive image features from scale-invariant keypoints." International journal of computer vision **60**(2): 91-110.

Luccheseyz, L. and S. Mitray (2001). "Color image segmentation: A state-of-the-art survey." Proceedings of the Indian National Science Academy (INSA-A) **67**(2): 207-221.

Lukas Neumann, J. M. (2015). Efficient Scene Text Localization and Recognition with Local Character Refinement. International Conference on Document Analysis and Recognition (ICDAR).

Matas, J., O. Chum, M. Urban and T. Pajdla (2004). "Robust wide-baseline stereo from maximally stable extremal regions." Image and Vision Computing **22**(10): 761-767.

Matas, J., C. Galambos and J. Kittler (2000). "Robust detection of lines using the progressive probabilistic Hough transform." Computer Vision and Image Understanding **78**(1): 119-137.

Messelodi, S. and C. M. Modena (1999). "Automatic identification and skew estimation of text lines in real scene images." Pattern Recognition **32**(5): 791-810.

Minetto, R. (2012). Text Recognition and 2D/3D Object Tracking, UNIVERSIDADE ESTADUAL DE CAMPINAS.

Minetto, R., N. Thome, M. Cord, J. Fabrizio and B. Marcotegui (2010). SnooperText: A multiresolution system for text detection in complex visual scenes. 2010 IEEE International Conference on Image Processing, IEEE.

Minetto, R., N. Thome, M. Cord, J. Stolfi, F. Precioso, J. Guyomard and N. J. Leite (2011). Text detection and recognition in urban scenes. Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE.

Mishra, N., C. Patvardhan, C. Vasantha Lakshmi and S. Singh (2012). "Shirorekha Chopping Integrated Tesseract OCR Engine for Enhanced Hindi Language Recognition." International Journal of Computer Applications **39**: 19-23.

Mohamed, S., T. Mahmoud and M. Ibrahim (2017). "Efficient Edge Detection Technique Based on Hidden Markov Model using Canny Operator." threshold **6**(01).

Neumann, L. and J. Matas (2010). A method for text localization and recognition in real-world images. Asian Conference on Computer Vision, Springer.

Neumann, L. and J. Matas (2016). "Real-Time Lexicon-Free Scene Text Localization and Recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence **38**: 1872-1885.

Neumann, L., J. Matas and Ieee (2012). Real-Time Scene Text Localization and Recognition. 2012 Ieee Conference on Computer Vision and Pattern Recognition. New York, Ieee: 3538-3545.

Otsu, N. (1975). "A threshold selection method from gray-level histograms." Automatica **11**(285-296): 23-27.

Palumbo, P. W., S. N. Srihari, J. Soh, R. Sridhar and V. Demjanenko (1992). "Postal address block location in real time." Computer **25**(7): 34-42.

Pan, Y.-F., X. Hou and C.-L. Liu (2008). A robust system to detect and localize texts in natural scene images. Document Analysis Systems, 2008. DAS'08. The Eighth IAPR International Workshop on, IEEE.

Peanho, C. A., H. Stagni and F. S. C. da Silva (2012). "Semantic information extraction from images of complex documents." Applied Intelligence **37**(4): 543-557.

Peng, X., H. Cao, S. Setlur, V. Govindaraju and P. Natarajan (2013). Multilingual OCR research and applications: an overview. Proceedings of the 4th ACM International Workshop on Multilingual OCR. Washington, D.C., USA, ACM: 1-8.

Perantonis, S. J., B. Gatos, V. Maragos, V. Karkaletsis and G. Petasis (2004). Text area identification in web images. Hellenic Conference on Artificial Intelligence, Springer.

proceedings, I. (2015). "Born-Digital Images (Web and Email)." Retrieved 03/10/2017, 2017, from <http://rrc.cvc.uab.es/?ch=1&com=introduction>.

Risnumawan, A., P. Shivakumara, C. S. Chan and C. L. Tan (2014). "A robust arbitrary text detection system for natural scene images." Expert Systems with Applications **41**(18): 8027-8048.

Rossi, R. G. (2016). Classificação automática de textos por meio de aprendizado de máquina baseado em redes, Universidade de São Paulo.

Ryan, M. and N. Hanafiah (2015). "An Examination of Character Recognition on ID card using Template Matching Approach." International Conference on Computer Science and Computational Intelligence (Iccsci 2015) **59**: 520-529.

Saoi, T., H. Goto and H. Kobayashi (2005). Text detection in color scene images based on unsupervised clustering of multi-channel wavelet features. Eighth International Conference on Document Analysis and Recognition (ICDAR'05), IEEE.

Sharma, P. and K. Fujii "Automatic Contact Importer from Business Cards for Android."

Sharma, P. and S. Sharma (2016). Image processing based degraded camera captured document enhancement for improved OCR accuracy. 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), IEEE: 441-444.

Shi, C., C. Wang, B. Xiao, Y. Zhang and S. Gao (2013). "Scene text detection using graph model built upon maximally stable extremal regions." Pattern recognition letters **34**(2): 107-116.

Simon, M., E. Rodner and J. Denzler (2015). Fine-grained classification of identity document types with only one example. Machine Vision Applications (MVA), 2015 14th IAPR International Conference on, IEEE.

Simon, M., E. Rodner and J. Denzler (2015). Fine-grained classification of identity document types with only one example. 2015 14th IAPR International Conference on Machine Vision Applications (MVA), IEEE: 126-129.

Sin, B.-K., S.-K. Kim and B.-J. Cho (2002). Locating characters in scene images using frequency features. Pattern Recognition, 2002. Proceedings. 16th International Conference on, IEEE.

Smith, R. (2007). An Overview of the Tesseract OCR Engine. Ninth International Conference on Document Analysis and Recognition (ICDAR 2007).

Sonia Bhaskar, N. L., Scott Green (2011). "Implementing Optical Character Recognition on the Android Operating System for Business Cards."

Soukoreff, R. W. and I. S. MacKenzie (2001). Measuring errors in text entry tasks: an application of the Levenshtein string distance statistic. CHI '01 Extended Abstracts on Human Factors in Computing Systems. Seattle, Washington, ACM: 319-320.

Sun, J., Z. Wang, H. Yu, F. Nishino, Y. Katsuyama and S. Naoi (2003). Effective text extraction and recognition for WWW images. Proceedings of the 2003 ACM symposium on Document engineering. Grenoble, France, ACM: 115-117.

Sun, L., Q. Huo, W. Jia and K. Chen (2015). "A robust approach for text detection from natural scene images." Pattern Recognition 48(9): 2906-2920.

Szeliski, R. (2011). Computer vision : algorithms and applications. London ; New York, Springer.

Tahim, A. P. N. (2010). Localização e extração automática de textos em imagens complexas, Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis.

Trier, O. D. and A. K. Jain (1995). "Goal-directed evaluation of binarization methods." IEEE Transactions on Pattern Analysis and Machine Intelligence 17(12): 1191-1201.

Trier, Ø. D., A. K. Jain and T. Taxt (1996). "Feature extraction methods for character recognition-a survey." Pattern recognition 29(4): 641-662.

Turner, V., J. F. Gantz, D. Reinsel and S. Minton (2014). "The digital universe of opportunities: rich data and the increasing value of the internet of things." IDC Analyze the Future.

Urbschat, H., R. Meier, T. Wanschura and J. Hausmann (2015). System and method for increasing the accuracy of optical character recognition (OCR), Google Patents.

Valiente, R. and G. Bressan (2016). Text recognition improvement using a novel iterative method and super-resolution. V Workshop de Pós-Graduação Engenharia de Computação, São Paulo: PCS/POLI/USP.

Valiente, R., J. C. Gutiérrez, M. T. Sadaike and G. Bressan (2017). Automatic Text Recognition in Web Images. Proceedings of the 23rd Brazilian Symposium on Multimedia and the Web. Gramado, RS, Brazil, ACM: 241-244.

Valiente, R., M. T. Sadaike, J. C. Gutiérrez, D. F. Soriano, G. Bressan and W. V. Ruggiero (2016). A process for text recognition of generic identification documents over cloud computing. Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Walha, R., F. Drira, F. Lebourgeois, A. M. Alimi and C. Garcia (2016) "Resolution enhancement of textual images: a survey of single image-based methods." IET Image Processing **10**, 325-337.

Walha, R., F. Drira, F. Lebourgeois, C. Garcia and A. M. Alimi (2015). "Resolution enhancement of textual images via multiple coupled dictionaries and adaptive sparse representation selection." International Journal on Document Analysis and Recognition (IJDAR) **18**(1): 87-107.

Wang, X., L. Huang and C. Liu (2009). A new block partitioned text feature for text verification. 2009 10th International Conference on Document Analysis and Recognition, IEEE.

Yan, Z. Y., Y. Lu and J. W. Li (2011). "Super Resolution of Text Image by Pruning Outlier." Neural Information Processing, Pt Iii **7064**: 649-656.

Yangxing, L. and T. IKENAGA (2006). "A contour-based robust algorithm for text detection in color images." IEICE transactions on information and systems **89**(3): 1221-1230.

Yao, C., X. Bai, W. Y. Liu, Y. Ma, Z. W. Tu and Ieee (2012). Detecting Texts of Arbitrary Orientations in Natural Images. 2012 Ieee Conference on Computer Vision and Pattern Recognition. New York, Ieee: 1083-1090.

Yin, F., D. Chen and R. Wu (2011). "A distortion correction approach on natural scene text image." 1058-1061.

Yin, X. C., W. Y. Pei, J. Zhang and H. W. Hao (2015). "Multi-Orientation Scene Text Detection with Adaptive Clustering." Ieee Transactions on Pattern Analysis and Machine Intelligence **37**(9): 1930-1937.

Yin, X. C., X. W. Yin, K. Z. Huang and H. W. Hao (2014). "Robust Text Detection in Natural Scene Images." Ieee Transactions on Pattern Analysis and Machine Intelligence **36**(5): 970-983.

Yonemoto, S. (2014). "A Method for Text Detection and Rectification in Real-World Images." 374-377.

Yu-peng Gao, Y.-m. L. a. Z.-y. H. (2011). "Skewed Text Correction Based on the Improved Hough Tranform."

Zhang, S., M. Lin, T. Chen, L. Jin and L. Lin (2016). Character proposal network for robust text extraction. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE: 2633-2637.

Zhang, W., G. Zelinsky and D. Samaras (2007). Real-time accurate object detection using multiple resolutions. 2007 IEEE 11th International Conference on Computer Vision, IEEE.

Zhang, Z., W. Shen, C. Yao and X. Bai (2015). Symmetry-Based Text Line Detection in Natural Scenes. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 2558-2567.

Zhao, Z. Y., C. Fang, Z. C. Lin and Y. Wu (2015). "A robust hybrid method for text detection in natural scenes by learning-based partial differential equations." Neurocomputing **168**: 23-34.

Zhong, Y., K. Karu and A. K. Jain (1995). Locating text in complex color images. Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on, IEEE.

Zhu, A. N., G. Y. Wang and Y. B. Dong (2015). "Detecting natural scenes text via auto image partition, two-stage grouping and two-layer classification." Pattern Recognition Letters **67**: 153-162.

Zhu, A. N., G. Y. Wang, Y. B. Dong and B. K. Iwana (2015). "Detecting text in natural scene images with conditional clustering and convolution neural network." Journal of Electronic Imaging **24**(5): 10.

Zhu, S. and R. Zanibbi (2016). A Text Detection System for Natural Scenes with Convolutional Feature Learning and Cascaded Classification. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE: 625-632.

## 7 Apêndice A - Características textuais

Listam-se algumas características textuais comumente utilizadas para identificar as regiões da imagem que possuem texto.

### 1. Geometria

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

b. 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.

c. Coesão espacial. Texto é naturalmente um agregado de caracteres que apresentam-se em uma determinada orientação, exibindo alturas e espaçamento similares (veja Figura 7-1). Tal característica é comumente denominada coesão espacial.



Figura 7-1: Similaridades geométricas dos caracteres: alinhamento (marcador em azul), altura (marcadores em vermelho) e espaçamento (marcadores em verde).  
Fonte: (Tahim 2010).

### 2. Contraste (cor e intensidade)

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

b. O corpo (traço) dos caracteres geralmente possui uma cor (imagem colorida) percentualmente uniforme em toda a sua extensão. No entanto, os caracteres geralmente contêm 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 percentualmente 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.

### 3. Bordas

a. 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. 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. 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.

b. 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 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. Os contornos textuais podem caracterizar completamente o caractere. Além disso, tais contornos podem ser completamente extraídos caso possuam contraste com o plano de fundo.

#### 4. Textura

a. 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. 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 frequências distintas dos outros objetos da imagem.

## 8 Apêndice B - Etapa de localização

### Segmentação espacial

Na segmentação espacial, cada imagem é considerada isoladamente. A partição da imagem, ou seja, o conjunto de regiões disjuntas que compõem (completamente) a imagem é criada unicamente com base nas características espaciais de cada imagem.

A abordagem mais básica consiste na segmentação da imagem com base num limiar para o valor da luminância. Esta técnica, quando aplicada a imagens simples, depois de cuidadosamente selecionado o valor de limiar, resulta numa boa separação dos objetos com um brilho elevado e uniforme em relação ao fundo. A Figura 8-1 ilustra um exemplo da segmentação baseada na amplitude de uma imagem em tons de cinzento, utilizando um único limiar.

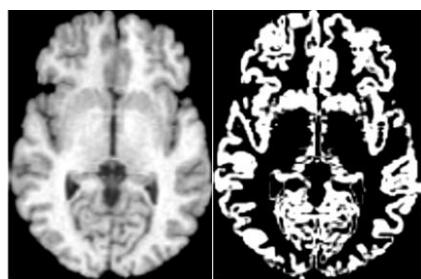


Figura 8-1: Exemplo de segmentação baseada na amplitude: (a) imagem original; (b) segmentação com um único limiar da imagem em (a). Fonte: (da Conceição Palma 2004)

O conceito de segmentação por limiar pode ser generalizado para vários limiares, visando a detecção de regiões que se diferenciam através dos seus níveis de luminância. Este tipo de segmentação usando múltiplos limiares apresenta um maior grau de dificuldade de implementação do que as técnicas de um único limiar. A razão desta dificuldade prende-se com a necessidade de estabelecer vários (bons) limiares para detectar as regiões pretendidas.

### Métodos Baseados em Região

Na segmentação baseada em regiões parte-se do pressuposto que pixels adjacentes e pertencentes a uma mesma região têm características visuais semelhantes, por exemplo, em termos de níveis de cinzento, cor ou textura. Dependendo do tipo de aplicação que a segmentação irá ter, da imagem a segmentar

e dos resultados pretendidos, a seleção dos critérios de homogeneidade pode ser mais ou menos sofisticada.

No caso dos textos legíveis, estes 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) percentualmente distintas do plano de fundo 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) 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.

As técnicas utilizadas na segmentação baseada em regiões apresentam, como principal vantagem, a sua eficiência na identificação de regiões homogêneas em termos das características espaciais selecionadas, bem como a sua exatidão na localização das fronteiras. Como maior desvantagem, deve considerar-se o elevado número de regiões que tipicamente surgem como resultado da segmentação.

### **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, regularidade na altura e espaçamento, alinhamento, etc.

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

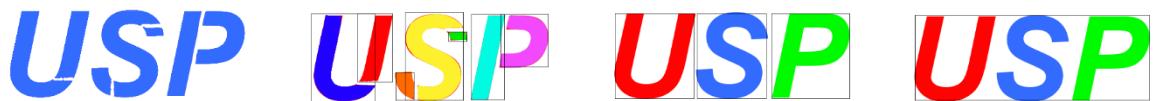


Figura 8-2: Métodos baseados em CCs - geração de CCs. (a) Imagem original com caracteres de baixa densidade e artefatos incluídos durante o processo de compressão. (b) Extração dos CCs da imagem original, em que cada CC está representado por uma cor e delimitado por um BB em preto. (c) Geração correta do CCs da imagem. (d) Agrupamento dos CCs. Fonte: Autor.

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 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 Figura 8-2 apresenta a geração de CCs de uma imagem com caracteres de baixa densidade contendo artefatos Figura 8-2 (a). Nota-se na Figura 8-2 (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 Figura 8-2 (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 Figura 8-2 (c) tratam-se de uma única palavra e devem ser agrupados Figura 8-2 (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. Então, 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.

Os métodos baseados em CCs geralmente utilizam diversos limiares heurísticos para determinar quais conjuntos de CCs representam texto ou não-texto, o que ajuda na redução do custo computacional e funciona bem em aplicações específicas, porém pode ser limitado em aplicações mais gerais.

### **Métodos Baseados em Bordas**

Pode definir-se ‘fronteira’ como sendo uma zona onde ocorre uma ou mais variações nas características da imagem. As técnicas de segmentação baseadas na detecção de fronteiras assumem que o valor de pelo menos uma propriedade dos pixels varia rapidamente na fronteira entre duas regiões. Assim, estas técnicas procuram localizar variações abruptas nos valores de alguma propriedade dos pixels, tais como: o nível de cinzento, cor, contraste ou alguma outra medida local que permita identificar uma fronteira entre duas regiões.

O processo de segmentação baseada na detecção de fronteiras pode ser dividido em três etapas principais:

#### **1) Detecção das fronteiras**

Esta etapa consiste tipicamente na aplicação de operadores para detecção de fronteiras. Estes operadores podem basear-se em uma de duas aproximações:

Detecção das diferenças espaciais na imagem: Neste tipo de abordagem, a imagem é processada de forma a acentuar as variações espaciais de amplitude, i.e. as zonas onde se localizam as fronteiras. Para tal, são normalmente empregues dois tipos de técnicas:

- Cálculo da derivada de primeira ordem: Nestas técnicas, as zonas da imagem sobre as quais o cálculo da derivada de primeira ordem

produz valores elevados correspondem a descontinuidades, i.e. fronteiras. A primeira derivada pode ser estimada através do cálculo dos gradientes (na vertical e horizontal) na vizinhança de cada pixel.

- Cálculo da derivada de segunda ordem: Nestas técnicas, a passagem por zero na segunda derivada indica a presença de fronteiras na imagem, uma vez que este zero corresponde ao ponto central de uma transição na imagem.

Este tipo de detecção é usualmente implementado com base nos designados detectores de fronteira, tais como: operadores de Sobel, Roberts, Prewitt, Laplacian e Canny. Na Figura 8-3 ilustram-se exemplos da detecção de fronteiras utilizando vários operadores.

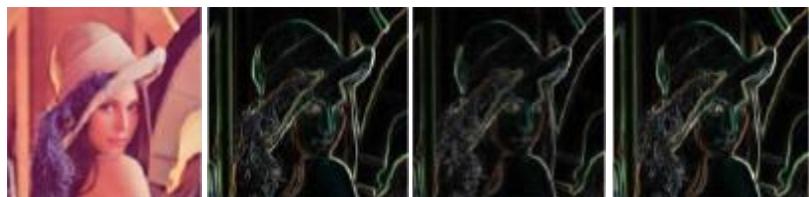


Figura 8-3: Exemplos da detecção de fronteiras: (a) imagem original; (b), (c) e (d) resultado da detecção de fronteiras utilizando os operadores de Prewitt, Roberts e Robison, respectivamente, para imagem em (a). Fonte: Adaptação de(Gonzalez and Woods 2008).

- Adaptação a um dado modelo de fronteira – Neste tipo de abordagem, os valores dos pixels correspondentes a uma determinada zona da imagem são comparados com um modelo de fronteiras. Esta técnica pressupõe o conhecimento a priori do tipo de fronteiras esperado para a imagem. Por exemplo, as fronteiras podem ser detectadas através do seguimento de modelos paramétricos, tais como linhas retas, círculos ou elipses. Neste caso, técnicas especiais como a transformada de Hough têm de ser aplicadas para fazer a identificação das fronteiras.

As fronteiras resultantes da aplicação destes operadores são normalmente descontínuas. Para além disso, nas imagens onde as fronteiras não são muito contrastadas podem surgir falsas detecções, localizadas onde não existem realmente

limites de regiões, ou então, as fronteiras podem ser omitidas onde os limites das regiões realmente existem.

## 2) Seleção das fronteiras

A segunda etapa do processo de segmentação baseado na detecção de fronteiras consiste na seleção dos segmentos de fronteiras mais relevantes detectados no passo anterior. Esta seleção pode ser feita com base nas seguintes técnicas:

- Limiar de fronteira: Neste caso, a seleção de fronteiras na imagem faz-se através da utilização de um valor limiar que permita remover as fronteiras detectadas com um valor de gradiente inferior a esse limiar;
- Relaxação de fronteira: Neste caso, uma medida da qualidade de fronteira é calculada para cada fronteira, decidindo-se assim quais as fronteiras que devem, ou não, ser descartadas. Para tal, é analisada a magnitude da fronteira, i.e. o valor do seu gradiente, bem como o contexto onde a fronteira existe de modo a avaliar a qualidade de cada fronteira. O critério mais usado para descartar uma fronteira é baseado num valor de limiar determinado em função dos valores de qualidade pretendidos.

## 3) Identificação das regiões

Na terceira e última etapa, as fronteiras selecionadas na etapa anterior são combinadas em cadeias de forma a definirem os limites das várias regiões. Após a conclusão desta etapa, os pixels que não estiverem separados por uma fronteira são considerados como fazendo parte da mesma região.

A identificação dos limites das regiões pode ser efetuada utilizando as seguintes técnicas:

- Ligação de fronteiras: As fronteiras podem ser ligadas entre si, se estiverem próximas umas das outras. Assim, se uma fronteira estiver próxima de outra e se o ângulo entre as suas tangentes for relativamente pequeno, estas podem ser ligadas;
- Transformada de Hough: Se os limites das regiões procuradas seguirem um modelo paramétrico conhecido, por exemplo, se a forma do objeto

for conhecida, pode utilizar-se a transformada de Hough para localizar esses limites a partir das fronteiras anteriormente detectadas na imagem;

- Um grafo onde os limites das regiões correspondem a caminhos nesse grafo. Como informação inicial, apenas são necessários os pontos de início e fim do limite da região. Desta forma, uma cadeia de fronteiras representativas do caminho óptimo, para esse limite, pode ser determinada usando uma função de avaliação de caminhos;
- Programação dinâmica: Utiliza-se o princípio de optimização de Bellman's que diz o seguinte “o caminho óptimo entre dois pontos é igualmente óptimo entre quaisquer dois pontos situados no mesmo caminho”. Este princípio pode ser aplicado ao problema da determinação das fronteiras das regiões, se for definida uma noção de ‘boa fronteira’. Esta técnica pode ser utilizada para selecionar a melhor fronteira dentre as várias cadeias de fronteiras existentes entre um ponto de início e um de fim.

### **Métodos Baseados em Textura**

As técnicas de segmentação baseadas na textura detectam regiões com características homogêneas em termos de textura, sendo de salientar a sua eficácia na detecção de regiões com uma diversidade de texturas elevada ainda que tenham a mesma luminância e crominâncias médias.

A noção de textura, apesar de poder ser identificada em praticamente todos os tipos de imagens e, em particular, em imagens naturais, não tem uma definição precisa, universalmente aceite pela comunidade científica. A dificuldade em elaborar uma definição de textura suficientemente genérica resulta, em parte, do elevado número de atributos que seria necessário incluir numa definição desse tipo.

Uma definição de textura foi proposta por Gagalowicz e Ma “Se se mover uma janela sobre uma textura e se efetuarem medidas texturais nessa janela, os resultados dessas medidas devem ser invariantes”. Esta definição remete para uma outra questão importante, ou seja, a noção de resolução da textura que pode ser definida como o tamanho mínimo da janela que permite obter medidas texturais (conjunto de estatísticas locais ou outras propriedades locais) invariantes. Exemplos de texturas e da aplicação da segmentação de textura são ilustrados na Figura 8-4.

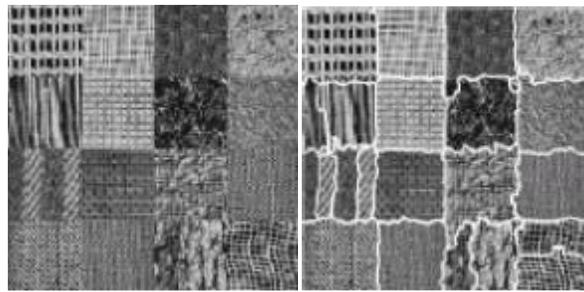


Figura 8-4: Exemplo de segmentação espacial baseada na textura: (a) imagem original constituída por vários tipos de textura; (b) regiões correspondentes à segmentação da imagem em (a). Fonte: (Jain and Yu 1998, da Conceição Palma 2004).

Dois tipos principais de texturas podem ser consideradas

- Texturas aleatórias: Texturas típicas de algumas imagens de superfícies naturais. De um modo geral, não apresentam descontinuidades bem definidas. Na Figura 8-5(a), pode observar-se um exemplo de uma textura aleatória.
- Texturas determinísticas: Texturas que se caracterizam por uma estrutura onde é possível identificar padrões elementares que se repetem no espaço da imagem em várias direções, de um modo mais ou menos regular. Um exemplo de uma textura determinística pode ser observado na Figura 8-5 (b).

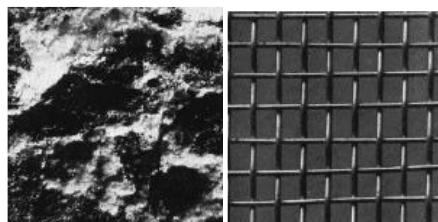


Figura 8-5: Exemplos de texturas: (a) textura aleatória; (b) textura determinística. Fonte: (Jain and Yu 1998, da Conceição Palma 2004).

## Correspondência de modelos

### SIFT (Scale Invariant Feature Transform)

O algoritmo SIFT proposto por (Lowe 2004) consegue identificar e descrever pontos chave em imagens, o que é feito através de um mapeamento com diferentes

vistas de um objeto ou cena, resultando em um vetor com 128 valores que descrevem cada ponto chave da imagem. O algoritmo consiste nas seguintes etapas:

Detecção de extremos no espaço-escala: os pontos chave são detectados aplicando um filtro em cascata que identifica os candidatos, que são invariantes à escala, usando uma função que procura por descritores estáveis ao longo de diferentes escalas. O espaço-escala é definido com a função  $L(x,y,\sigma)$  , com uma imagem de entrada  $I(x,y)$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$$

onde \* é a convolução com a Gaussiana  $G(x,y,\sigma)$  .

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma} e^{-(x^2 + y^2)/2\sigma^2}$$

Para detectar localização de pontos chave estáveis no espaço-escala, (Lowe 2004) Lowe propôs o uso da função de diferença Gaussiana (DoG) no espaço-escala convoluída com a imagem  $I(x,y)$ , resultando em  $D(x,y,\sigma)$ , a qual pode ser calculada a partir de duas escalas próximas separadas por um fator multiplicativo constante  $k$ .

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

Detecção de extremos locais: a partir de  $D(x,y,\sigma)$ , Lowe (Lowe 2004) sugere que os máximos e mínimos locais devem ser detectados pela comparação de cada pixel com os seus oito vizinhos na imagem corrente e nove vizinhos nas escalas superior e inferior (26 vizinhos).

Atribuição de orientação: a escala do ponto chave é usada para selecionar a imagem suavizada pela Gaussiana  $L$ , com a escala mais próxima, de modo que toda a computação seja realizada de modo invariante à escala. O gradiente de magnitude  $m(x,y)$  é obtido com a Equação.

$$m(x, y) = \sqrt{\Delta x^2 + \Delta y^2}$$

onde  $\Delta x = L(x + 1, y) - L(x - 1, y)$  e  $\Delta y = L(x, y + 1) - L(x, y - 1)$ . A orientação  $\theta(x,y)$  é calculada pela Equação:

$$\theta(x, y) = \arctan(\Delta y / \Delta x)$$

Descrição dos pontos chave: calcula um descritor para cada região da imagem local, que é distinta e invariante a variações adicionais, tais como mudanças na iluminação ou ponto de vista 3D.

### **SURF (Speeded Up Robust Features Algorithm)**

SURF (Bay, Ess et al. 2008) é um detector e descritor de pontos chave invariante a rotação e a escala, que é computacionalmente muito rápido. O detector de descritores SURF é baseado na matriz Hessiana. O determinante da matriz Hessiana é usado para determinar a localização e escala do descritor. Dado um ponto  $p = (x,y)$  na imagem  $I$ , a matriz Hessiana  $H(x,\sigma)$  em  $p$  na escala  $\sigma$  é definida como segue:

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix}$$

Onde  $L_{xx}(x, \sigma)$  é a convolução da derivada de segunda ordem da Gaussiana  $\frac{\partial^2}{\partial x^2} g(\sigma)$  com a imagem  $I$  no ponto  $p$ , e similarmente para  $L_{xy}(x, \sigma)$  e  $L_{yy}(x, \sigma)$ . A matriz de determinantes Hessianos é escrita como:

$$\det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2$$

Para localizar pontos de interesse sobre escalas, é aplicada uma supressão não máxima em uma vizinhança  $3 \times 3 \times 3$ .

O descritor SURF é extraído em duas etapas: a primeira etapa é a atribuição de uma orientação com base nas informações de uma região circular em torno dos pontos de interesse detectados. A orientação é computada usando respostas Haar-Wavelet, nas direções x e y, que são pesadas com uma Gaussiana ( $\sigma = 3.3s$ ) centrada no ponto de interesse a fim de aumentar a robustez às deformações geométricas, e respostas Wavelet em direções dx horizontal e vertical dy são adicionadas em cada sub-região. Os valores absolutos  $|dx|$  e  $|dy|$  são somados a fim de obter informação sobre a polaridade das alterações da intensidade da imagem. Portanto, cada sub-região tem um vetor  $v$  de descritor de quatro dimensões.

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$$

Isso resulta em um vetor de descritores para todas as sub-regiões 4x4 de tamanho 64.

### Classes principais de descritores de forma

**Descritores de forma baseados no contorno :** Os descritores baseados no contorno descrevem uma região conexa tendo em conta os seus shapels mais exteriores, ou seja, o contorno fechado da mesma. A Figura 8-6 mostra um exemplo de um contorno fechado a ser descrito por este tipo de parâmetros.

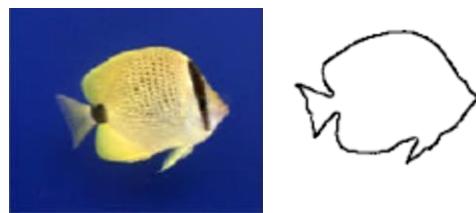


Figura 8-6: (a) Imagem com um objeto; (b) contorno do objeto em (a). Fonte: (Jain and Yu 1998, da Conceição Palma 2004).

Os principais parâmetros de forma baseados no contorno disponíveis na literatura podem ser organizados segundo as suas propriedades do seguinte modo(da Conceição Palma 2004):

- **Parâmetros geométricos :** Parâmetros que representam a forma de um objeto simples (ou seja com uma única região) usando propriedades geométricas do seu contorno tais como: o perímetro, a corda máxima, a circularidade, a convexidade e a excentricidade.
- **Parâmetros baseados em transformadas:** Parâmetros que representam a forma de um objeto simples utilizando coeficientes calculados a partir de uma dada transformada; exemplos são a transformada de Fourier e as Wavelets.
- **Parâmetros baseados em momentos:** Parâmetros que representam a forma de um objeto simples utilizando um conjunto de valores estatísticos; exemplos são os momentos geométricos, também denominados como momentos invariantes.
- **Parâmetros baseados em contornos normalizados:** Parâmetros que representam a forma de um objeto simples utilizando o seu contorno

normalizado; o contorno normalizado é insensível a transformações geométricas e ao número de pontos que o definem.

**Descritores de forma baseados em regiões :** Os descritores baseados em regiões descrevem uma região tendo em conta todos os seus shapes. Os parâmetros de forma baseados em regiões descrevem formas simples, mas também formas mais complexas, por exemplo, a forma de um objeto formado por várias regiões não conexas. A Figura 8-7 apresenta alguns objetos que poderão ser descritos por parâmetros baseados em regiões.



Figura 8-7: Exemplos de objetos simples e complexos, com as respectivas regiões e buracos. Fonte: (Jain and Yu 1998, da Conceição Palma 2004).

Os principais parâmetros de forma baseados em regiões disponíveis na literatura podem ser organizados segundo as suas propriedades do seguinte modo(Gonzalez, Bergasa et al. 2012):

- Parâmetros geométricos: Parâmetros que representam a forma de um objeto simples ou complexo usando as propriedades geométricas da região ou regiões que lhe correspondem; exemplos são a *bounding box*, área, centroide, projeções: altura e largura, diâmetro circular equivalente, solidez e compactação.
- Parâmetros baseados em transformadas: Parâmetros que representam a forma de um objeto simples ou complexo utilizando coeficientes calculados a partir de uma dada transformada; exemplos são a Transformada Angular-Radial (*Angular-Radial Transform*, ART) e a transformada de Fourier.
- Parâmetros baseados em momentos: Parâmetros que representam a forma de um objeto simples ou complexo utilizando um conjunto de valores estatísticos associados a um dado tipo de momento; exemplos são os momentos geométricos, os momentos de Legendre, os momentos de Zernike, os momentos rotacionais e os momentos complexos.

## 9 Apêndice C – Teste do sistema com imagens da Web

### Imagens da Web

Nesta seção, o sistema proposto é adaptado e avaliado para mostrar sua eficiência na detecção e reconhecimento de texto em um novo cenário, imagens com texto da Web, permitindo o reconhecimento do texto das imagens da Web para posterior análise e processamento do conteúdo. O resultado do uso do sistema proposto numa imagem da Web é mostrado na Figura 9-1.

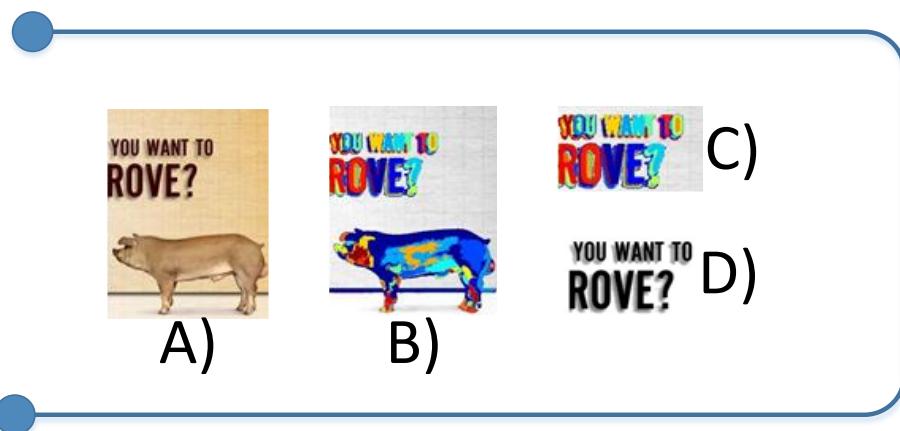


Figura 9-1: a) Imagem Original, b) após a etapa de Localização, c) após a etapa de Seleção, d) após a etapa de Extração. Fonte: Autor.

### Avaliação da etapa de seleção de texto em imagens da Web

Nesta fase, avaliamos a abordagem de seleção de texto em um conjunto de imagens coletado de páginas da Web. Este conjunto de dados contém 1000 imagens coloridas da Web contendo texto, que foram selecionados aleatoriamente baseadas no banco de imagens da competição ICDAR (proceedings 2015), o banco de imagens está distribuído nos seguintes formatos: 21 imagens .gif, 174 imagens .jpg e 805 imagens .png. Os tamanhos de imagens variam de  $44 \times 55$  a  $1625 \times 313$  pixels. Foi selecionada uma amostra representativa de páginas da Web de diferentes categorias (notícias, pessoal, comercial, social, governo, etc.) e e-mails de diferentes tipos (spam, boletins informativos, etc.) em proporções que refletem seu uso no mundo real, ver Figura 9-2. As palavras selecionadas estão em inglês. Neste conjunto de dados, foram rotuladas manualmente as localizações do texto usando BBs e comparadas com os resultados obtidos no processo automático.



Figura 9-2: Amostra do banco de imagens usado. Fonte: Autor.

Usando as imagens selecionadas e tendo em conta as métricas anteriormente explicadas, o desempenho geral do nosso método após da etapa de seleção é mostrado e comparado com outros trabalhos na Tabela 9-1.

Tabela 9-1: Resultados experimentais etapa de seleção. Fonte: Autor.

Autor	Imagens	Revocação%/ Precisão%	Método
(Lopresti and Zhou 2000)	482	78% / -	Cluster de cores e CCA
(Perantonis, Gatos et al. 2004)	1100	80%/ 64%	Localização de áreas de texto
(Sun, Wang et al. 2003)	43	89%/ 62%	Cluster de cores e CCA
(Liu, Yang et al. 2011)	1134	80%/ 74%	Algoritmo de extração de linhas de texto
Proposta	1000	91%/ 81%	Abordagem múltipla

Em (Lopresti and Zhou 2000), são usadas 482 imagens gif coletadas de páginas Web reais. O método de extração é fundamentado em agrupamento em espaço de cores seguido por uma análise de componentes conexos. Por outro lado em (Perantonis, Gatos et al. 2004) são usada estratégias de localização de áreas de texto. Porém nestes trabalhos não é aproveitada uma etapa de seleção (filtragem) das regiões que realmente apresentam caracteres o que reduz a taxa de detecção final.

Em (Sun, Wang et al. 2003) e (Liu, Yang et al. 2011) é realizada uma filtragem das regiões que não representam caracteres, no caso de (Sun, Wang et al. 2003) é

usado o valor da largura do traço para remover regiões não textuais. Os métodos existentes não usam retificação do texto, apresentando dificuldades em textos inclinados. Tendo em conta o experimento e as métricas descritas, o sistema proposto apresenta os melhores resultados, demonstrando o excelente desempenho de nosso algoritmo em outras aplicações.

### **Avaliação da etapa de reconhecimento de texto em imagens da Web**

Nesta fase, avaliamos a abordagem de reconhecimento de texto após a etapa de seleção e usando do algoritmo 2. Para a avaliação de resultados de reconhecimento de palavras, usamos a métrica de distância de edição padrão, ou Distância de Levenshtein. A comparação é insensível a maiúsculas e minúsculas. o desempenho geral do nosso método após da etapa de reconhecimento é mostrado e comparado com outros trabalhos na Tabela 9-2

Tabela 9-2: Resultados experimentais Etapa de reconhecimento. Fonte: Autor.

<b>Autor</b>	<b>Imagens</b>	<b>Distancia de edição%/ Reconhecimento%</b>	<b>Método</b>
(Lopresti and Zhou 2000)	50	-/ 92%	n-tuple classificadores
(Perantonis, Gatos et al. 2004)	1100	-/70%	-
(Sun, Wang et al. 2003)	43	-/70.04%	Simples-pré- processamentos
Proposta	1000	235/ 79%	Abordagem múltipla

Para o reconhecimento de caracteres, (Lopresti and Zhou 2000) usa técnicas de correções de polinômio de superfície e n-tuple classificadores, tentando compensar a baixa resolução das imagens (tipicamente 72 ppi). O método tem uma alta taxa de reconhecimento, porém a avaliação só foi realizada em 50 imagens Web previamente treinadas. Em (Perantonis, Gatos et al. 2004) e (Sun, Wang et al. 2003) não tem-se em conta a baixa resolução das imagens da Web. Nenhum dos trabalhos apresenta resultados em termos de distinção de edição.

O sistema desenvolvido apresenta resultados competitivos, porém existem limitações para textos não homogêneos, nos quais as taxas de detecção foram baixas; para textos extremamente pequenos (2-4 pixels de altura), que resultam em detecções perdidas; e para textos em imagens da Web de cenas reais, com problemas de iluminação; nas quais o sistema apresenta limitações.