Pós-Graduação em Ciência da Computação

"SEGMENTAÇÃO DE SENTENÇAS MANUSCRITAS ATRAVÉS DE REDES NEURAIS ARTIFICIAIS"

Por

César Augusto Mendonça de Carvalho

Dissertação de Mestrado

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE, AGOSTO/2008

UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CÉSAR AUGUSTO MENDONÇA DE CARVALHO

"SEGMENTAÇÃO DE SENTENÇAS MANUSCRITAS ATRAVÉS DE REDES NEURAIS ARTIFICIAIS"

ESTE TRABALHO FOI APRESENTADO À PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO DO CENTRO DE INFORMÁTICA DA UNIVERSIDADE FEDERAL DE PERNAMBUCO COMO REQUISITO PARCIAL PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIA DA COMPUTAÇÃO.

ORIENTADOR(A): PROF. GEORGE DARMITON DA CUNHA CAVALCANTI

RECIFE, AGOSTO/2008

Carvalho, César Augusto Mendonça de
Segmentação de sentenças manuscritas através de redes neurais artificiais / César Augusto Mendonça de Carvalho - Recife : O Autor, 2008.
xvii, 72 folhas : il., fig., tab.

Dissertação (mestrado) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, 2008.

Inclui bibliografia.

1. Inteligência artificial. 2. Inteligência computacional. I. Título.

006.3 CDD (22. ed.) MEI2009-038

Dissertação de Mestrado apresentada por **César Augusto Mendonça de Carvalho** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título **"Segmentação de Sentenças Manuscritas Através de Redes Neurais Artificiais"**, orientada pelo **Prof. George Darmiton da Cunha Cavalcanti** e aprovada pela Banca Examinadora formada pelos professores:

Prof. Tsang Ing Ren
Centro de Informática / UFPE

Prof. Carlos Alexandre Barros de Mello
Departamento de Sistemas Computacionais / UPE

Prof. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Visto e permitida a impressão.
Recife, 27 de agosto de 2008.

Prof. FRANCISCO DE ASSIS TENÓRIO DE CARVALHO
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Agradecimentos

Primeiramente a Deus, que me sustenta no ser e no poder difundir, pelos mais necessitados, um pouco do saber que Ele me permitiu adquirir;

Aos meus pais e familiares, que me apóiam e incentivam na superação dos obstáculos que aparecem na minha vida;

Ao Prof. George Darmiton da Cunha Cavalcanti, que foi o meu ilustre orientador e me apoiou fortemente durante o desenvolvimento desta dissertação, dando sugestões e opiniões a respeito de todo o seu conteúdo;

Aos demais Professores do Centro de Informática da Universidade Federal de Pernambuco, que compartilharam conhecimentos e experiências com esmero;

E, finalmente, a todos os alunos que, comigo, trilharam o Curso de Mestrado da Computação com dedicação e incentivo mútuo.

Resumo

O reconhecimento automático de textos manuscritos vem a cada dia ganhando importância tanto no meio científico quanto no comercial. Como exemplos de aplicações, têm-se sistemas bancários onde os campos de valor dos cheques são validados, aplicativos presentes nos correios para leitura de endereço e código postal, e sistemas de indexação de documentos históricos.

A segmentação automática do texto em palavras ou caracteres é um dos primeiros passos realizados pelos sistemas de reconhecimento dos textos manuscritos. Portanto, é essencial que seja alcançado um bom desempenho de segmentação para que as etapas posteriores produzam boas taxas de reconhecimento do texto manuscrito.

O presente trabalho trata do problema de segmentação de sentenças manuscritas em palavras através de duas abordagens: (i) método baseado na métrica de distância *Convex Hull* com modificações que objetivam melhorar o desempenho de segmentação; (ii) um novo método baseado em Redes Neurais Artificiais que visa superar problemas existentes em outras técnicas de segmentação, tais como: o uso de heurísticas e limitação de vocabulário.

O desempenho dos métodos de segmentação foi avaliado utilizando-se de uma base de dados pública de texto manuscrito. Os resultados experimentais mostram que houve melhora de desempenho das abordagens quando comparadas à abordagem tradicional baseada em distância *Convex Hull*.

Palavras-chave: Segmentação automática de texto manuscrito, *Convex Hull*, Redes Neurais Artificiais.

Abstract

Automatic recognition of handwritten texts is an important and challenging task. Many applications require an efficient and low-cost recognition system, such as bank system processing, mail system for address and postal code reading and systems for historical document indexation. Therefore, a massive effort has been done in the academic environment to improve the accuracy rate and the performance, in terms of time, for systems like the ones described above.

Automatic text segmentation is one of the initial steps leading to the complete recognition of handwritten sentences in systems that appraise words separately. Thus, a good segmentation performance is essential to achieve good recognition accuracy rates and to avoid manual intervention, which is much more expensive.

This work addresses the problem of unconstrained sentence segmentation based on two different approaches: (i) an algorithm based on the Convex Hull Gap Metric; (ii) a classification strategy to perform segmentation based on an Artificial Neural Network. These two approaches aim to overcome problems found in the traditional approaches, such as heuristics and vocabulary limitation.

The performance of the segmentation methods was evaluated using handwritten text lines extracted from a public database. The experimental results showed that the developed techniques achieved better accuracy rates in comparison with the traditional Convex Hull approach.

Keywords: Automatic handwritten text line sentence segmentation, Convex Hull, Artificial Neural Network.

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1
Introdução

1.1 Motivação

Com a evolução tecnológica, ficou mais fácil a comunicação entre pessoas de diversas partes do mundo. Para isso, são utilizadas representações digitais do texto desejado que podem ser transmitidas através de sistemas computacionais com alta eficiência. Além de facilitar a comunicação, a tecnologia construiu um meio mais barato e durável de conservação das informações textuais quando comparada ao papel, pois reduziu-se a quantidade de espaço físico necessário para armazenamento das informações, além de reduzir significativamente o desgaste provocado pelo tempo.

Existem diversos trabalhos tanto no meio científico quanto no comercial para transformar as informações manuscritas em representações digitais. Além das vantagens mencionadas anteriormente, o uso da representação digital das informações textuais possibilita automatização de diversas tarefas cotidianas. Como exemplos de aplicações que utilizam-se de procedimentos de reconhecimento de texto manuscrito, têm-se sistemas bancários onde os campos dos cheques são validados automaticamente, aplicativos presentes nos correios para leitura de endereço e código postal. Ainda dentro desse contexto, existem os sistemas de indexação de textos contidos em documentos históricos. Criando-se uma versão digital do documento, é possível utilizar mecanismos automáticos de busca por conteúdo dos textos para agilizar a procura das informações desejadas e facilitar a sua manipulação. Outros benefícios estão no fato dos documentos históricos não estarem mais sujeitos a danos pela ação do tempo e a redução do custo associado à manutenção do acervo histórico.

1.2 Contexto

O foco deste trabalho está na segmentação automática de texto manuscrito em palavras. Ou seja, dada uma linha de texto deseja-se encontrar e separar de maneira correta cada uma das palavras que constituem essa linha. Esse procedimento é um dos passos iniciais dos sistemas de

reconhecimento dos textos manuscritos, pois esses precisam avaliar cada palavra ou caractere separadamente. Portanto, um bom desempenho em termos de taxa de acerto de segmentação é fundamental para que procedimentos posteriores resultem em boas taxas de reconhecimento. Com uma alta confiabilidade, os sistemas de segmentação dispensam intervenções manuais, que são procedimentos mais lentos e mais custosos que os automáticos.

Uma das maneiras mais comumente utilizada de realizar segmentação da linha em palavras é dada pelos seguintes passos:

- Encontrar os componentes conectados da imagem: dentro do contexto deste trabalho, componentes conectados são os elementos da imagem do documento manuscrito, que representam palavras inteiras, letras isoladas ou até mesmo segmentos de letras;
- Calcular a distância entre eles;
- Escolher um limiar que determine quais das distâncias calculadas são de componentes de uma mesma palavra e quais não são.

Seguindo este método, pode-se inferir que a tarefa de segmentar as palavras de um texto impresso é mais simples que em textos manuscritos, pois o espaçamento entre os caracteres é uniforme no primeiro caso, podendo ser facilmente calculado.

Conforme pode ser visto no exemplo de texto impresso mostrado na Figura 1.1, a distância entre caracteres (letras de uma mesma palavra) e distâncias entre palavras são constantes, sendo a segunda maior que a primeira. Dessa forma, as palavras podem ser facilmente separadas por meio da distância horizontal entre os componentes.

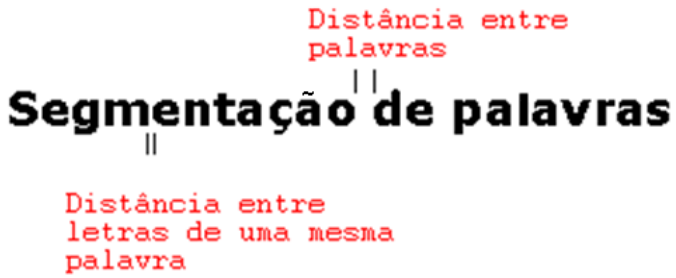


Figura 1.1 Espaçamento regular entre os componentes de um texto impresso

De forma contrária, nos textos manuscritos não existe uniformidade de espaçamento entre os caracteres, transformando a tarefa de segmentação num procedimento mais difícil e elaborado. Existem casos, por exemplo, que o espaçamento entre letras de uma mesma palavra é maior que o espaçamento entre palavras distintas.

Dentre outros problemas existentes em textos manuscritos estão: variação de tamanho dos caracteres, inclinação na escrita, ruídos na imagem, influência da escrita no verso do documento e borramento. Uma outra dificuldade na segmentação deste tipo de texto é a ocorrência de palavras vizinhas conectadas, ou seja, a não existência de espaçamento entre elas. Todos os problemas citados tornam o trabalho de modelagem de sistemas de segmentação de palavras manuscritas correspondente às habilidades humanas bastante desafiador.

Na Figura 1.2, é mostrado um exemplo de texto manuscrito que exemplifica alguns dos problemas na segmentação de palavras. Por exemplo, pode-se observar que o espaçamento entre a segunda letra "a" do texto e a letra "m" é maior que o espaço entre as palavras "Alameda" e "Ave". Outros pontos que podem ser observados estão relacionados à inclinação da escrita e à variação do tamanho dos caracteres.

Figura 1.2 Alguns dos problemas existentes em textos manuscritos como inclinação e espaçamento entre palavras menor que entre letras da palavra "Alameda". Retirada de [SC94]

A maioria das técnicas de segmentação parte do princípio de que os espaços entre palavras são maiores que aqueles entre caracteres. Seni e Cohen [SC94] apresentam oito diferentes técnicas para cálculo de distâncias entre componentes. Dentre as técnicas estão as distâncias *Bounding Box*, Euclidiana, *Run-length* e outras que utilizam-se de heurísticas. A melhor taxa de acerto alcançada foi de 90,30% usando a distância *Run-length* com heurísticas. Posteriormente, Mahadevan e Nagabushnam [MN95] propuseram uma técnica que utiliza as distâncias entre as cascas convexas (*Convex Hull* - CH) dos componentes conectados da imagem para a detecção de distâncias entre caracteres e palavras. O método CH alcançou resultados melhores (93,30% de acerto) que os métodos apresentados por Seni e Cohen. Ambos os experimentos foram realizados em uma mesma base de dados composta por linhas de ruas, cidades, estados, códigos postais e nomes de pessoas extraídas de imagens dos correios dos Estados Unidos da América [MN95].

Mais recentemente, Marti e Bunke [MB01b] e Manmatha e Rothfeder [MR05] testaram o método *Convex Hull* em páginas de textos manuscritos extraídas da base de dados pública IAM[1][MB99]. Os experimentos alcançaram 95,56% e 94,40% de taxas de acerto, respectivamente. Outros métodos utilizam classificadores, como *Hidden Markov Model* (HMM) e Redes Neurais Artificiais (RNA) através do método apresentado em [MSBS04], para realizar a

[1]Disponível em: <http://www.iam.unibe.ch/fki/databases/iam-handwriting-database>

segmentação de sentenças manuscritas através de um procedimento iterativo de segmentação e reconhecimento. Nesse tipo de método, a imagem contendo as palavras manuscritas é dividida em partes menores que são submetidas ao módulo de reconhecimento, que indica se a sub-imagem em questão contém uma palavra conhecida. Esse processo iterativo é repetido até que algum critério de parada seja alcançado. Contudo, essa abordagem apresenta a desvantagem de ser limitada por um conjunto reduzido de palavras do módulo de reconhecimento.

1.3 Objetivo

O principal objetivo deste trabalho é desenvolver métodos de segmentação de sentenças manuscritas em palavras. A seguir, são apresentados os métodos de segmentação desenvolvidos e alguns dos problemas a serem superados:

- O primeiro método desenvolvido baseia-se na técnica *Convex Hull*. Nesse, são propostas quatro heurísticas que foram desenvolvidas objetivando superar alguns problemas existentes na abordagem descrita por Marti e Bunke [MB01b]. Os resultados desta abordagem são testados sobre a base de dados IAM. Um ponto importante a ser mencionado é que no trabalho citado anteriormente, a taxa de acerto foi estimada através de inspeção visual. Neste trabalho, é apresentado o procedimento automático de análise dos experimentos desenvolvido, que tem como finalidade aumentar a confiabilidade de avaliação. Os estudos experimentais são apresentados, mostrando que resultados promissores foram alcançados atingindo menores taxas de erro que a técnica *Convex Hull* tradicional;
- O segundo e principal método de segmentação desenvolvido neste trabalho baseia-se em Redes Neurais Artificiais e procura superar as seguintes dificuldades: i) Sistemas de segmentação baseados em métricas de distâncias (*Gap Metrics*) necessitam de heurísticas para melhorar o desempenho e adaptar-se a diferentes tarefas de segmentação [LVB07]; ii) Algumas abordagens de segmentação, como a apresentada por Morita et al. [MSBS04] que visa o reconhecimento de datas manuscritas, possuem limitação de vocabulário (dias e meses do ano) que reduz o domínio de aplicação do sistema.

1.4 Estrutura do trabalho

O presente trabalho é dividido em seis capítulos: Neste capítulo é apresentada a motivação, contexto e objetivo desta dissertação. No Capítulo 2, é mostrada uma visão geral da arquite-

tura dos sistemas de reconhecimento de texto manuscrito, mostrando as principais etapas que devem ser realizadas para alcançar o resultado desejado. No Capítulo 3, as técnicas de segmentação baseadas em distâncias (*Gap Metrics*) são mostradas em detalhes, enfatizando a distância *Convex Hull*, a partir da qual foram desenvolvidas heurísticas para melhorar o desempenho de segmentação. O método de segmentação baseado em Redes Neurais Artificiais é apresentado em detalhes no Capítulo 4. Todos os experimentos realizados e resultados alcançados nos testes das duas técnicas de segmentação desenvolvidas neste trabalho estão detalhadas no Capítulo 5 e, por fim, o Capítulo 6 expõe as considerações finais e trabalhos futuros.

CAPÍTULO 2
Arquitetura de sistemas de reconhecimento de texto manuscrito

2.1 Introdução

Os sistemas automáticos de reconhecimento de texto manuscrito são divididos em dois grandes grupos: *on-line* e *off-line*. Essa divisão refere-se à forma de representação dos dados, que variam de acordo com a maneira na qual o texto manuscrito foi adquirido. No texto manuscrito *on-line*, os dados são armazenados através de coordenadas bi-dimensionais dos pontos da escrita em função do tempo (representação espaço-temporal) [PS00]. A partir dessas, é possível se obter outras informações, por exemplo, a velocidade média de escrita do usuário. Pode-se armazenar também a pressão exercida pelo escritor entre a caneta e o dispositivo coletor das informações manuscritas no momento da escrita.

Os sistemas de reconhecimento *off-line* trabalham a partir da representação completa do texto manuscrito através de uma imagem obtida como uso de um digitalizador. Com isso a representação *off-line* fornece informações espaciais e de luminosidade do texto manuscrito. A ausência de informações temporais e de pressão, torna o reconhecimento dos textos manuscritos *off-line* uma tarefa mais difícil que a *on-line* devido à menor quantidade de informações. Por outro lado, os sistemas *off-line* são mais viáveis economicamente além de serem a única opção em alguns casos como reconhecimento do texto de um acervo histórico.

Neste capítulo, é dada uma visão geral da arquitetura dos sistemas de reconhecimento de texto manuscrito *off-line*. As principais atividades realizadas no procedimento de reconhecimento do texto manuscrito são apresentadas, dentre as quais está a segmentação de sentenças manuscritas em palavras, que é a atividade foco deste trabalho.

2.2 Módulos do sistema de reconhecimento

Apesar de existirem diversas abordagens para reconhecer textos manuscritos, os sistemas seguem basicamente a arquitetura apresentada na Figura 2.1.

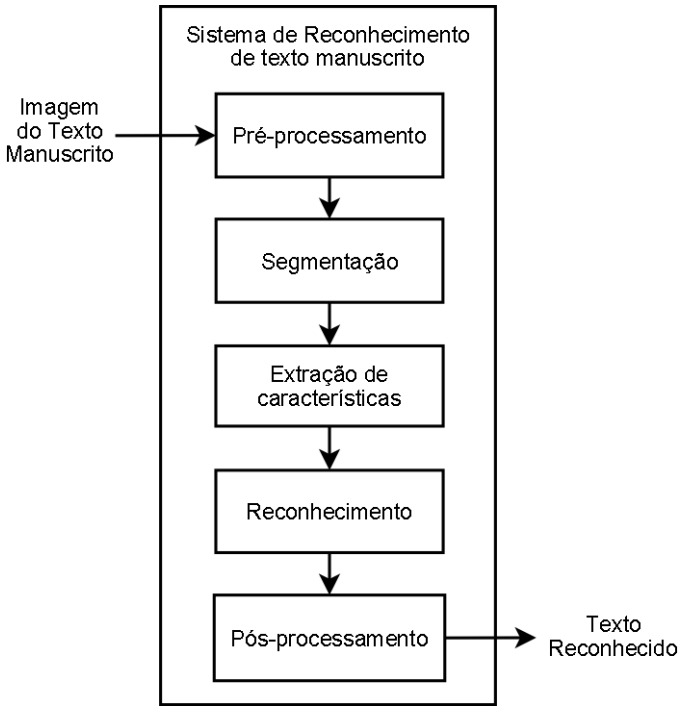


Figura 2.1 Módulos básicos existentes nos sistemas de reconhecimento de texto manuscrito

Inicialmente, tem-se a fase de pré-processamento cujo objetivo principal é realizar correções e ajustes na imagem contendo o texto manuscrito para melhorar o desempenho das fases posteriores. A segmentação visa subdividir a imagem em componentes menores contendo linhas, palavras ou caracteres isolados, que serão utilizados como entrada da etapa seguinte. O módulo de Extração de características é responsável pela criação da representação correspondente à imagem em análise. Na etapa de Reconhecimento, as informações extraídas na fase anterior são analisadas dando como resultado o conteúdo textual associado à imagem de entrada. Por fim, tem-se o módulo de pós-processamento cujo objetivo é maximizar as taxas de acerto de reconhecimento através da análise e modificação do resultado inicial dado pelo sistema.

2.2.1 Pré-processamento

É comum existirem problemas nas imagens de texto manuscrito devido a falhas no processo de digitalização, características inerentes ao documento ou estilo de escrita do usuário. Portanto, são necessários alguns ajustes que são realizados pela fase de pré-processamento objetivando eliminar deformações que comprometam processamentos posteriores [Oli99a].

No pré-processamento, podem-se destacar as seguintes etapas: binarização, redução de

ruídos, esqueletização, correção de inclinação, dentre outras. A seguir, serão descritas as etapas de Binarização, Redução de Ruídos e Esqueletização.

Binarização

O principal objetivo dessa tarefa é separar as informações que são inerentes à tinta (*foreground*) das inerentes ao papel (*background*) [PS00]. As imagens digitalizadas e armazenadas em tons de cinza geralmente apresentam no seu histograma dois picos, conforme exemplificado na Figura 2.2: um menor referente às informações da escrita (região de interesse) e o outro maior relacionado ao papel.

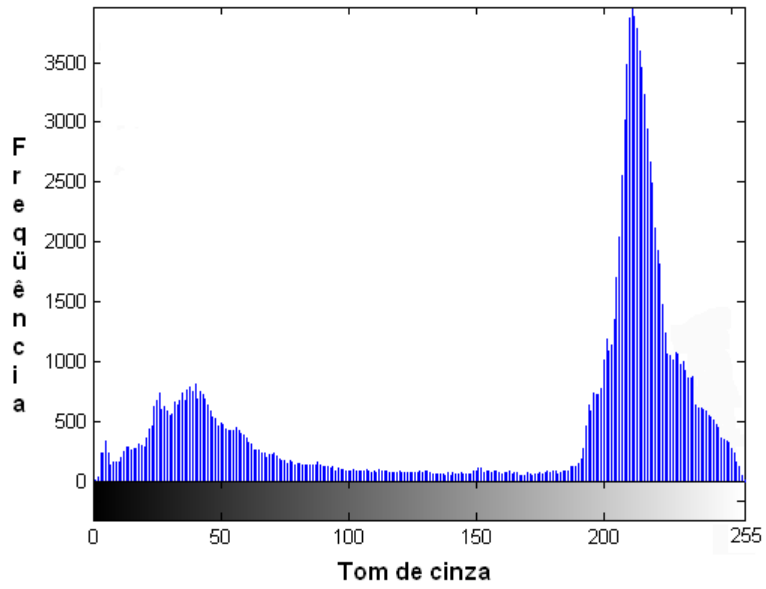


Figura 2.2 Dois picos comuns nos histogramas de imagens de documentos manuscritos

A binarização, portanto, consiste basicamente em definir um valor limiar para transformar as informações em tons de cinza em duas cores (preto e branco). Os tons que estiverem abaixo do limiar, que correspondem às regiões de tinta, serão transformados na cor preta. E os tons que estiverem acima do limiar serão transformados na cor branca.

No procedimento de binarização global é calculado apenas um único limiar para toda a imagem, porém existem técnicas de binarização adaptativas, que ajustam o limiar para diferentes regiões da imagem. Essa segunda abordagem comporta-se melhor que a primeira em documentos que apresentam variação de iluminação na imagem. Entre os diversos métodos de binarização de imagens existentes está o método de Otsu [Ots79], que é um dos mais referenciados da atualidade [SS04]. Como exemplos de métodos adaptativos locais pode-se citar:

Niblack [Nib86] e Sauvola e Pietaksinen [SP00].

Redução de ruídos

Os ruídos existentes nas imagens dos documentos de texto manuscritos surgem principalmente no procedimento de digitalização, por exemplo, o ruído sal-e-pimenta, ou através da mídia de transmissão. Por não se tratarem de elementos textuais, os ruídos devem ser removidos/suavidados para não interferirem nos procedimentos seguintes a fim de melhorar o desempenho do reconhecimento do texto manuscrito.

Para a tarefa de redução de ruídos são utilizados algoritmos de suavização. Estes, além de remover pixels destoantes da imagem, abrandam os contornos das palavras. Existem diversos tipos de algoritmos de suavização, dentre eles estão aqueles baseados em operações morfológicas, operações celulares e filtros espaciais [O'G92].

O algoritmo kFill é um dos métodos que pode ser usado para redução de ruídos maiores que um pixel [O'G92]. A Figura 2.3 apresenta o resultado do uso de um algoritmo proposto por Krisana et al. [CRT98] que baseia-se no kFill. Na parte superior da Figura encontra-se a imagem com ruídos e abaixo desta pode-se observar a imagem resultante do uso do algoritmo de redução de ruídos.

Figura 2.3 Resultado do uso do algoritmo de redução de ruídos baseado no kFill. Retirada de [CRT98]

Esqueletização

O objetivo principal dos algoritmos de esqueletização é diminuir a quantidade de informações dos componentes mantendo a estrutura do texto. Neste procedimento, os traços, que

compõem as letras ou componentes da imagem, são reduzidos a linhas centrais, também conhecidas como esqueleto. Dessa forma, apenas as informações essenciais que descrevem os componentes da imagem continuam na imagem. Dentre as técnicas de esqueletização, estão os algoritmos apresentados por Carrasco e Forcada [CF95], Zhang e Suen [ZS84] e Nishida [NM94].

Na Figura 2.4, é exemplificada a esqueletização do número 5371. Na esquerda é mostrada a imagem original e na direita o resultado do procedimento de esqueletização que é o esqueleto da imagem.

(a) Original (b) Esqueletizada

Figura 2.4 Resultado da esqueletização proposta por Nishida [NM94]. Retirada de [Oli99a]

A idéia principal da esqueletização é retirar a influência da espessura do traço nas etapas seguintes. Pois independente do instrumento utilizado para se escrever várias vezes uma determinada palavra, existindo conseqüentemente variação na espessura do traço de escrita, a informação estrutural é a mesma [OK97].

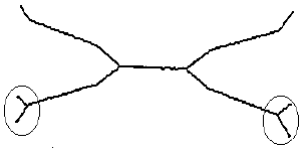
Com a esqueletização, as estruturas da imagem são simplificadas facilitando a execução das etapas subseqüentes, porém os algoritmos utilizados podem introduzir problemas, tais como: eliminação de parte da imagem, quebra de conectividade entre pixels e criação de ramificações indesejadas comprometendo o processo de reconhecimento (problema exemplificado na Figura 2.5).

2.3 Segmentação

Tendo em mãos a imagem do documento manuscrito, deve-se determinar as regiões de interesse da mesma. Segundo O'Gorman [OK97] segmentação divide-se em segmentação de documento e segmentação de texto. A primeira consiste em identificar regiões de texto e gráfico do documento em análise. A segunda, trabalha nas regiões de texto segmentando-as em trechos de interesse (linhas, palavras ou caracteres).

A imagem do documento manuscrito pode ser segmentada em diversos níveis, dependendo

(a) Imagem Original



(b) Imagem esqueletizada com ramificações indesejadas circuladas

Figura 2.5 Falha gerada no procedimento de esqueletização proposto Jang [JC94]. Retirada de [Oli99b]

da abordagem de reconhecimento adotada pelo sistema. Os possíveis tipos de segmentação, do mais macro para o de menor magnitude, subdividem a imagem em: documento; texto; blocos de linhas, que formam as colunas da imagem; parágrafos compostos por uma ou mais linhas de texto; linhas de texto isoladamente; palavras; caracteres ou componentes dos caracteres.

A seguir, são descritos os seguintes cenários: segmentação da imagem em linhas de texto; e segmentação de linhas de texto em palavras ou caracteres.

2.3.1 Segmentação da imagem em linhas de texto

Uma linha de texto é composta por grupos de caracteres, símbolos e palavras relativamente perto entre si, onde se pode traçar uma linha reta através deles [OK97].

A forma mais intuitiva de se segmentar a imagem em linhas é através de projeção vertical ou horizontal [OK97]. Tais projeções mostram a freqüência de pixels de texto presentes em cada linha ou coluna da imagem (Figura 2.6).

Essa abordagem de segmentação baseada em projeção funciona bem quando a inclinação das linhas presentes na imagem é pequena ou quase nula. Pode-se utilizar, portanto, algoritmos de correção de inclinação do texto (***skew correction***), que através de rotações nos componentes da imagem, deixam as palavras paralelas ao eixo horizontal. A transformada de Hough [DH72] é um dos métodos utilizados para a detecção do ângulo de inclinação das linhas de textos.

Existem comumente, nas projeções de documentos manuscritos, picos (região de alta freqüência) intercalados por região com freqüência muito baixa ou nula. Na Figura 2.6, estão representadas as projeções verticais e horizontais de um documento com duas colunas de texto

e três linhas. Cada pico presente na projeção representa uma região de texto e os espaçamentos entre os picos correspondem às regiões de ausência de texto. Portanto, separando as regiões da imagem que apresentam alta freqüência nas projeções são obtidas as linhas e colunas de texto quando usada a projeção horizontal e vertical, respectivamente.

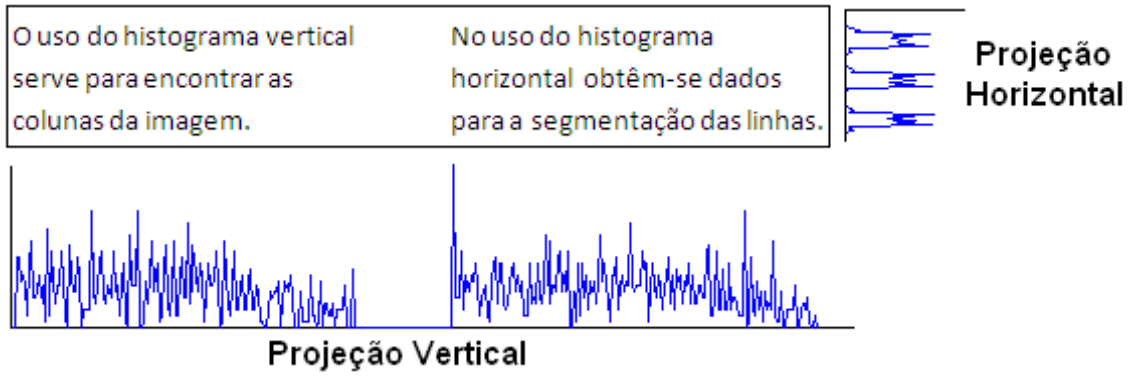


Figura 2.6 Projeções verticais e horizontais de parte de um texto

2.3.2 Segmentação de linhas de texto em palavras ou caracteres

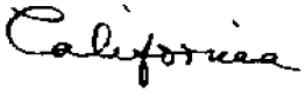
Após a segmentação da imagem do documento manuscrito em linhas, pode-se realizar um procedimento para se encontrar as palavras existentes em cada imagem. O objetivo desse tipo de segmentação é reduzir a complexidade de trabalho do módulo de reconhecimento.

Poucos trabalhos lidam com a segmentação de palavras [PS00]. Os trabalhos iniciais desta área ([SC94],[MN95]) baseiam-se na idéia que o espaçamento entre palavras é maior que a distância entre os caracteres de uma mesma palavra. O método apresentado por Kim e Govindaraju [KG98] usa redes neurais para segmentar as palavras. Este trabalho baseia-se em habilidades humanas de detecção de quebra de palavras, que usam não apenas o espaço entre componentes, mas também: pontuação, letras maiúsculas seguidas de minúsculas, transição de letras para dígito, além do reconhecimento de palavras. Existem outros métodos de segmentação baseados em um processo interativo com um módulo de reconhecimento, por exemplo o apresentado por Morita et al. [MSBS04]. Comparando-o aos outros, tem-se que este último tipo de abordagem é lento devido às constantes chamadas ao módulo de reconhecimento.

Uma vez que as palavras estejam segmentadas é possível utilizar operações de normalização para facilitar o processamento das etapas posteriores, tais como: ***Slant Correction*** que consiste na correção de inclinação dos traços que compõem os caracteres para que os mesmos fiquem na vertical [SRI99] (a Figura 2.7 mostra o resultado do uso de técnica de correção de inclinação)

e ***Scaling*** que consiste na padronização de tamanho dos caracteres para evitar que caracteres maiores tenham mais ou menos influência sobre a etapa de extração de características quando comparados aos caracteres menores.

(a) Imagem Original



(b) Imagem com inclinação corrigida

Figura 2.7 Correção de inclinação. Retirada de [Oli99a]

Dependo do que é esperado pelo módulo de reconhecimento (palavras ou caracteres), é necessário mais um passo de segmentação para se isolar os caracteres das palavras. Com isso simplifica-se mais ainda a tarefa de reconhecimento, pois o universo de letras é muito menor que o universo das palavras.

O procedimento de segmentação de palavras em caracteres pode ser dividido nas seguintes etapas: determinação dos pontos potenciais de fronteira, que são encontrados através de características como ligações e concavidades nos componentes da imagem; e fusão e controle desses pontos para se determinar a localização de cada caractere [Oli99b].

Esta dissertação tem como domínio de atuação a segmentação de linhas de texto manuscritos em palavras. Os passos anteriores e posteriores a essa etapa devem ser tratados através de técnicas já existentes ou trabalhos futuros. Na Seção 2.7, são apresentados alguns trabalhos relevantes da área de segmentação de sentenças manuscritas em palavras.

2.4 Extração de características

Nessa fase, o sistema de reconhecimento de texto manuscrito já realizou o pré-processamento da imagem original e subseqüente segmentação em regiões de interesse. A próxima tarefa é realizar a classificação dos elementos existentes nas imagens (palavras ou caracteres). Para tal, é necessário que seja realizada a extração de características para que as mesmas sejam utilizadas para a identificação da classe do componente em questão. Analisar a imagem da

forma como ela é exige muito esforço computacional devido à sua dimensionalidade, portanto, deve-se extrair características representativas das mesmas para facilitar o desenvolvimento e melhorar o desempenho dos algoritmos de classificação.

Existem dois tipos de características: as estatísticas, que são obtidas através de cálculos numéricos ou de valores booleanos sobre a imagem; e as estruturais, que fazem uma análise comportamental dos elementos presentes na imagem [Oli99b]. Entre as características estatísticas pode-se utilizar os momentos invariantes de Hu [Hu62]. Dentre as diversas características estruturais, pode-se citar: concavidades, contornos, distâncias, informações do esqueleto, dentre outras.

A extração de característica é uma fase bastante importante no procedimento de reconhecimento de texto manuscrito. O desenvolvimento desta etapa deve satisfazer ítens como: bom desempenho de execução para onerar de forma mínima o tempo de processamento do sistema e as características extraídas devem ter boa representatividade dos dados para descrever bem os diferentes estilos de escrita.

2.5 Reconhecimento

Diferentemente dos textos impressos, que apresentam uniformidade no espaçamento entre os caracteres, o manuscrito apresenta muita variabilidade de estilo e distância entre os seus componentes (conforme pode ser visto na Figura 5.2 do Capítulo 5), tornando o processo de reconhecimento mais desafiador.

Dentre as abordagens existentes para o reconhecimento de texto manuscrito existe uma que faz uso de *Hidden Markov Models* (HMM) [MSBS04]. Essa técnica transforma as características observadas das palavras em uma estrutura probabilística capaz de representá-la. Assim, é possível realizar uma comparação entre as características da imagem que contém o texto manuscrito com as existentes no dicionário do sistema, podendo assim classificá-la.

Outros métodos de reconhecimento utilizam-se de Redes Neurais Artificiais para o reconhecimento do elementos presentes na imagem [MGS05]. Essas recebem como entrada características inerentes à imagem e produzem como saída a classe correspondente.

2.6 Pós-processamento

A fase de pós-processamento tem como objetivo realizar modificações nas classificações dadas pelo módulo de reconhecimento objetivando melhorar a taxa de acerto do sistema. Tais modifi-

cações ocorrem, principalmente, quando existe resultado dúbio na classificação de uma palavra ou caractere, isto é, existe mais de uma classificação com probabilidades próximas de ocorrência para o elemento em análise. Portanto, cabe ao módulo de pós-processamento analisar de forma contextualizada qual das opções de classificação é a melhor.

Na Figura 2.8, a frase "*Friday Boston*" exemplifica a importância da análise contextual realizada pelo módulo de pós-processamento. É possível verificar que a letra "B" desta imagem se assemelha à escrita do número "13", no entanto cabe ao módulo identificar que a informação presente na imagem é literária e não numérica.

Figura 2.8 Importância da análise contextual. A letra "B" se analisada separadamente, pode ser reconhecida como o número 13. Retirada de [Oli99a]

A análise contextual está ligada de forma intrínseca ao tipo de aplicação de reconhecimento. Por exemplo, no caso de sistemas que manipulam cheques bancários pode-se utilizar informações de caracteres como "." ou "," para a classificação da parte inteira e os centavos dessa moeda. Outra possibilidade, ainda dentro de sistemas de reconhecimento de cheques, é fazer a comparação das informações numéricas com aquelas escritas por extenso para melhorar a precisão do sistema.

2.7 Estado da arte da área de segmentação de linhas de texto em palavras

Nesta seção, algumas técnicas relevantes presentes na literatura para a segmentação de sentenças manuscritas em palavras são apresentadas. Os métodos em questão tratam da separação das palavras de textos *off-line*, onde as informações do documento manuscrito são descritas através de imagem.

Pode-se dividir as diversas abordagens de segmentação nas seguintes categorias presentes na Tabela 2.1: métricas de distâncias, características gerais, segmentação baseada em reconhecimento e outras.

A primeira categoria utiliza-se de métricas de distâncias (*Gap Metric*) entre os diversos componentes conectados (elementos isolados da imagem que representam palavra, letra(s) ou parte de uma letra) existentes na imagem. Para a segmentação das palavras, é determinado

um limiar que é usado para agrupar os componentes pertencentes a cada palavra presente na imagem. Ou seja, todos os componentes da imagem cuja distância entre si é inferior ao limiar são considerados como pertencentes a uma mesma palavra.

Por se tratar da forma mais intuitiva de segmentação, os métodos baseados em distâncias foram os primeiros métodos desenvolvidos (inicialmente por Seni e Cohen [SC94] e Mahadevan e Nagabushnam [MN95]) e ainda apresentam resultados satisfatórios na segmentação de palavras manuscritas. Mais detalhes sobre *Gap Metrics*, além de um estudo visando o aprimoramento de uma das técnicas, são apresentados no Capítulo 3.

Tabela 2.1 Categorização das técnicas de segmentação de sentenças manuscritas

Segmentação de Sentenças	Métrica de Distâncias	Bounding Box
		Euclidiana
		Run-Lenght
		Run-Lenght com heurística *Bounding Box* (RLBB)
		Run-Lenght com heurística Euclidiana (RLE)
		Convex Hull
	Características Gerais	Histogramas Verticais
		Ascendentes
		Descendentes
		Letras Maiúsculas Seguidas de Minúsculas
	Segmentação Baseada em	*Hidden Markov Model*
		Redes Neurais Artificiais
	Outras	Sistemas Híbridos
		Sistemas Heurísticos

A segunda categoria é formada por técnicas que atuam sobre características gerais dos componentes presentes na imagem. Como exemplo de características, tem-se: histogramas de projeções verticais (contagem dos pixels pretos em cada coluna da imagem), larguras potenciais de fronteira de palavras, ascendentes e descendentes, caracteres maiúsculos seguidos de minúsculos, dentre outras [SC94][PGS99][KG98].

Na terceira categoria, encontram-se os algoritmos de segmentação baseados em métodos de reconhecimento. Nessa, são realizadas divisões da imagem original em sub-imagens que, em seguida, são submetidas para um módulo de reconhecimento, que indica se a mesma é ou não uma palavra conhecida. No entanto, essa abordagem possui um custo computacional maior que as anteriores [MSBS04]. *Hidden Markov Model* (HMM), por exemplo, são classificadores que podem ser utilizados para o reconhecimento de caracteres, palavras ou sentenças.

Na última categoria estão as abordagens que utilizam-se de mais de uma estratégia e/ou

heurísticas, muitas vezes baseadas na cognição humana, para realizar a segmentação de palavras [KG98] [MR05] [NK05].

A seguir, são detalhadas algumas das estratégias de segmentação presentes na literatura na área de segmentação de palavras manuscritas *off-line*. Conforme citado anteriormente, outros métodos de segmentação estão descritos no Capítulo 3.

2.7.1 Algoritmo de Morita *et al.*

Hidden Markov Model (HMM) [Rab89] é um processo probabilístico de eficiência provada como sendo uma das ferramentas mais poderosas de modelagem de voz além de uma variedade de outras aplicações que envolvam sinais do mundo real.

Diversas propriedades deste modelo motivam a sua aplicação em sistemas de modelagem de caracteres, palavras ou sentenças. Uma das principais, é a possibilidade de uso de ferramentas automáticas e eficientes capazes de treinar o modelo sem a necessidade de rotulação de dados previamente segmentados [MSBS04].

Morita *et al*. [MSBS04] apresentaram uma abordagem híbrida de HMM-MLP (*Multilayer Perceptron*) para segmentação e reconhecimento de datas manuscritas. Nesse trabalho, para o processo de segmentação foi usado o HMM. Na etapa de reconhecimento, as duas técnicas (MLP e HMM) foram utilizadas, sendo a primeira utilizada para reconhecimento de dígitos e a segunda para reconhecimento dos meses do ano. As imagens de texto foram extraídas de datas manuscritas de cheques brasileiros. A segmentação foi feita através da transformação das imagens em seqüências de observações através de aplicações sucessivas de pré-processamento, segmentação e extração de características. O pré-processamento consiste em correções de inclinação. Na etapa seguinte, são produzidas várias sub-imagens a partir das quais são extraídas características. Por fim, a segmentação é derivada de um procedimento de reconhecimento alcançado através do melhor caminho produzido pelo algoritmo de Viterbi [Jr73], que representa a melhor forma de segmentação entre as possibilidades de escrita de data existentes nas imagens estudadas.

2.7.2 Heurísticas

Nesta seção são mostradas algumas abordagens de segmentação baseadas em heurísticas, nas quais, algumas técnicas de processamento de imagens são aplicadas no desenvolvimento delas.

Abordagem baseada em escala de espaçamento

A técnica apresentada em [MR05] aplica filtros Laplacianos nas imagens que contêm as sentenças a serem segmentadas. São utilizadas várias escalas do filtro e, como resultado, são geradas regiões de bolhas que correspondem aos componentes conectados da imagem, sejam esses caracteres ou palavras. Assim, o objetivo do algoritmo de segmentação proposto passa a ser a determinação da melhor escala, que faz com que as bolhas correspondam a palavras.

Os autores propõem que a melhor escala do filtro é dada por aquela que produz as maiores áreas das bolhas para cada imagem. Após a escolha da melhor escala, as palavras podem então ser segmentadas através da determinação dos *Bounding Boxes* das bolhas.

Os experimentos realizados em [MR05] objetivaram a comparação dessa abordagem com a existente em [MB01b] que baseia-se na métrica de *Convex Hull* (CH). Duas bases de dados foram utilizadas. Na base de dados IAM [MB99], o algoritmo de segmentação do tipo CH obteve melhor desempenho que o proposto (5,6% contra 9,5%), porém na base de dados de documentos históricos de George Washington (presentes na biblioteca do congresso americano[1]) o resultado foi o inverso, sendo o proposto (17% de taxa de erro) muito melhor que o anteriormente existente na literatura (31,9% de taxa de erro). O que leva a crer que o algoritmo proposto alcança melhores resultados sobre bases de imagens de documentos históricos. Tais documentos, geralmente, apresentam clareamento da tinta usada para a escrita, influência do verso do papel e outros ruídos.

Abordagem através de primitivas verticais

Nwogu e Kim [NK05] apresentaram uma abordagem heurística, baseada em primitivas verticais das imagens, para detectar quebras de palavras. A base de dados utilizada neste artigo é composta por formulários de atendimento hospitalar (*Pre-Hospital Care Reports* - PCRs), que são usados para colher informações dos pacientes atendidos nos centros de emergência médicos.

O método desenvolvido por eles não trabalha diretamente com os pixels das images dos caracteres, e sim com uma representação desses através de primitivas de linhas verticais. As heurísticas apresentadas nesse artigo baseiam-se nas características das imagens que os humanos utilizam para a detecção de quebra de palavras. Conforme apresentado em [KG98], estas características são basicamente: espaços entre componentes, pontuação, letras maiúsculas seguidas de minúsculas, transição de letras para dígitos, além do reconhecimento de palavras.

Primitivas verticais dos pontos convexos dominantes dos contornos superiores foram extraídas para o exame do espaçamento global intra e inter-palavras [NK05]. As métricas globais

[1] Acessível através do endereço: <http://memory.loc.gov/ammem/gwhtml/gwseries.html>

são constantemente atualizadas com os espaçamentos locais. A altura das primitivas verticais também foi considerada objetivando a detecção de pontuação. São feitas diversas considerações heurísticas, como: em uma linha de texto existem mais de uma palavra, a distância entre as primitivas verticais de uma mesma palavra são, em geral, menor que a distância inter-palavra média global.

A heurística apresenta um sistema de pontuação para os espaçamentos encontrados, baseado em informações dos componentes vizinhos e globais. Assim, no final do algoritmo um limiar é determinado e conseqüentemente todas as quebras de palavras.

Os experimentos foram realizados através do uso do algoritmo proposto por Nwogu e Kim [NK05] versus o proposto em [KG98], que utiliza-se de uma abordagem com Redes Neurais Artificiais (RNA). Foram avaliados o número de palavras segmentadas corretamente, as que foram super-segmentadas (dividida em partes menores) e as que foram sub-segmentadas (mas de uma palavra segmentada como uma só). Nwogu e Kim mostraram que o algoritmo proposto por eles obteve melhor resultado já que o número de palavras corretamente segmentadas foi significativamente maior na sua implementação (69% de taxa de acerto versus 55% da abordagem proposta em [KG98]).

CAPÍTULO 3
Segmentação baseada em métricas de distância

3.1 Introdução

Este capítulo tem como finalidade apresentar as técnicas de segmentação de palavras baseadas nas distâncias entre os componentes presentes na imagem das sentenças manuscritas.

A maioria das técnicas de segmentação parte do princípio de que os espaços entre palavras são maiores que aqueles entre caracteres. Seni e Cohen [SC94] apresentam oito diferentes técnicas para cálculo de distâncias entre componentes. Detalhes de cada métrica de distância apresentada por Seni e Cohen são exibidos nas Seções 3.2 à 3.7.

Posteriormente, Mahadevan e Nagabushnam [MN95] apresentaram mais uma métrica entre componentes. Essa técnica, chamada de *Convex Hull* (explicada em detalhes na Seção 3.8), foi reproduzida nos nossos estudos e modificada, com o uso de heurísticas apresentadas na Seção 3.9, para que a mesma pudesse alcançar melhores resultados de segmentação. Os valores apresentados em tais heurísticas foram obtidos sobre a base de dados IAM [MB99].

Alguns conceitos necessários para o entendimento das técnicas de segmentação baseadas em distâncias são apresentados a seguir:

- Componentes conectados: elementos conectados da imagem que, dentro do contexto deste trabalho, representam palavras inteiras, letras isoladas ou até mesmo segmentos de letras;

- *Bounding box*: menor retângulo que envolve totalmente uma forma (componente) de uma imagem;

- Sobreposição horizontal: ocorre quando existe pelo menos uma reta vertical capaz de tocar dois *bounding boxes* de uma imagem, ou seja, para uma dada coordenada do eixo horizontal, existe mais de um componente conectado. Assim, quando dois ou mais componentes podem ser tocados por uma ou mais retas verticais traçadas na imagem estes possuem região de sobreposição horizontal;

- Sobreposição vertical: ocorre quando existe pelo menos uma reta horizontal capaz de tocar dois *bounding boxes* de uma imagem.

3.2 *Bounding Box*

O método *Bounding Box* consiste no cálculo da distância horizontal entre os *bounding boxes* dos componentes conectados. Neste método, para *bounding boxes* que se sobrepõem horizontalmente, a medida de distância é zero.

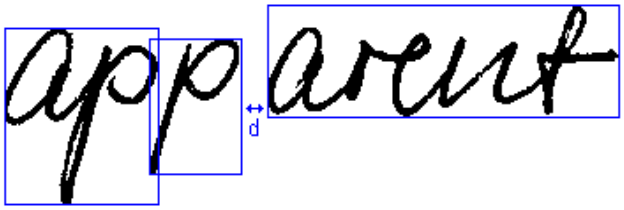


Figura 3.1 Cálculo da distância *Bounding Box*. "d" representa a distância entre os componentes "p" e "arent".

Na Figura 3.1 são mostrados os *Bounding Boxes* correspondentes aos três componentes presentes da palavra *"apparent"* ("ap","p" e "arent"). É possível notar que existe sobreposição horizontal entre os dois primeiros componentes ("ap" e "p") e que, apesar do fato destes componentes não se tocarem, a distância *Bounding Box* entre eles é zero. Ainda nessa figura, é mostrada a distância *Bounding Box* "d" entre os componentes "p" e "arent"

3.3 Distância Euclidiana

A distância Euclidiana é medida entre os dois pontos mais próximos entre dois componentes da imagem. O cálculo da distância Euclidiana é dado por: $\sqrt{h^2+v^2}$, sabendo que *"h"* e *"v"* são, respectivamente, a distância horizontal e vertical entre os dois pontos.

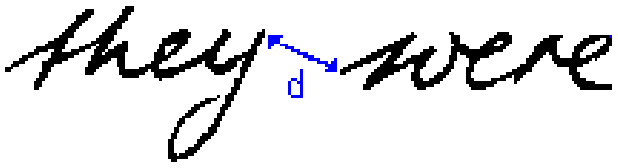


Figura 3.2 Distância Euclidiana entre os componentes "*they*" e "*were*" representada pela letra "d"

Na Figura 3.2 é mostrado um exemplo desse tipo de distância.

3.4 *Run-length* mínimo

Nesse método apenas as distâncias entre componentes que se sobrepõem verticalmente são calculadas. Para os demais casos, a distância *Bounding Box* é utilizada. A distância *Run-length* mínima é medida através da linha horizontal que toca os dois pontos mais próximos dos componentes na região de sobreposição vertical.

Figura 3.3 Exemplo de distância *Run-length* mínima

Na Figura 3.3, é mostrada a linha horizontal que toca os dois pontos mais próximos dos componentes desta imagem.

3.5 *Run-length* médio

Nesse método, as distâncias entre todos os pontos dos componentes dentro da região de sobreposição vertical são calculadas. Como resultado, é considerada a média das distâncias calculadas. Caso os dois componentes cuja distância está sendo medida não se sobreponham verticalmente, a distância *Bounding Box* é calculada.

Figura 3.4 Exemplo de distância *Run-length* média

Na Figura 3.4, são mostradas algumas das distâncias *Run-length* sendo calculadas na região de sobreposição dos dois componentes da imagem.

3.6 Distância *Run-length* com heurística *Bounding Box* (RLBB)

Esse método consiste no uso da distância *Run-length* mínima para o cálculo da distância entre componentes que se sobrepõem verticalmente, para os demais casos é usada uma heurística com a distância *Bounding Box*. A heurística consistem em:

- Caso os componentes não se sobreponham horizontalmente, é usada a distância *Bounding Box* horizontal;
- Se existir sobreposição horizontal (ocorre quando existe, pelo menos, uma reta vertical capaz de tocar dois *Bounding Boxes* de uma imagem), porém não total, é calculada a distância vertical entre os *Bounding Boxes* dos dois componentes;
- Para o caso em que um dos componentes está totalmente sobreposto horizontalmente em relação ao outro, é considerada a distância zero entre eles.

Na Figura 3.5, são mostrados casos de componentes que não são sobrepostos verticalmente, a diferença entre elas é que a Figura 3.5(a) possui um componente totalmente sobreposto horizontalmente a outro e, na Figura 3.5(b), a sobreposição entre os componentes é parcial.

(a) (b)

Figura 3.5 Componentes com sobreposição horizontal: (a) total e (b) parcial

Um ponto crítico foi identificado nesse método e está exemplificado através da Figura 3.6. Nela, são exibidos os diversos *Bounding Boxes* dos componentes da imagem (a mesma possui a palavra "*nominating*"). O pingo do segundo "i" está sobreposto verticalmente em relação ao componente "at", portanto, foi calculada a distância *Run-length* entre este pingo e o corpo da letra "t".

Figura 3.6 Caso crítico no método RLBB

Percebe-se, ainda neste exemplo, que o pingo do "i" está mais próximo do corpo do "i" do que do corpo do "t". Talvez fosse mais interessante calcular a distância para o corpo do "i" evitando que um valor de distância grande fosse dada a componentes que estão relativamente próximos.

Na Figura 3.7, é mostrado o resultado da segmentação da palavra através do RLBB. Pode-se observar que o pingo do "i" ficou erroneamente separado do corpo do "i".

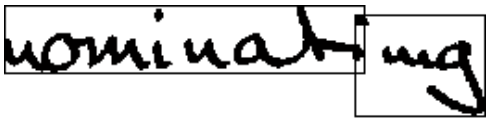


Figura 3.7 Segmentação errada da segunda letra "i" pelo RLBB

3.7 *Run-length* com heurística Euclidiana (RLE)

Esse método consiste em calcular a distância mínima de *Run-length* para componentes que são sobrepostos verticalmente em pelo menos 13,3% de polegada (em [SC94] não existem informações que justifique o uso deste valor). Para os demais casos, é utilizada a distância Euclidiana mínima. Portanto, o algoritmo precisa de detalhes quanto à resolução da imagem para calcular quantos pixel representam a citada porcentagem de polegada.

Para este trabalho, foi utilizada uma base de dados de imagens com resolução de 300 dpi. Assim, para componentes com mais de 40 pixels (300 * 0,133 = 39,9) de sobreposição vertical, é calculada a distância Run-length mínima entre eles.

Foram observadas algumas deficiências no método RLE, que motivaram a criação de duas heurísticas [SC94]. Essas, deram origem a dois novos métodos de cálculo de distância entre componentes conectados: RLE com heurística 1 e RLE com heurística 2.

3.7.1 RLE com heurística 1 (RLE(H1))

Consiste numa versão do RLE com uma heurística adicionada. Se dois componentes estão totalmente sobrepostos horizontalmente (Figura 3.5(a) exemplifica essa situação) a distância entre eles é zero.

A idéia desta heurística é garantir que componentes totalmente sobrepostos não sejam separados. Por exemplo, a letra "T" escrita de forma que a barra horizontal esteja separada da vertical seria obrigatoriamente considerada como um mesmo componente através desta heurística.

3.7.2 RLE com heurística 2 (RLE(H2))

Este método é uma evolução do RLE(H1) com uma heurística adicional. Essa tenta superar dificuldades encontradas em situações onde componentes que se sobrescrevem horizontalmente não são bem avaliados pelo método *Run-Length* convencional, já que o método não considera a proximidade vertical entre estes componentes.

Na Figura 3.8, é mostrado o caso em que a letra “i” está dividida em dois componentes, o pingo do “i” está mais próximo do corpo do “i” do que o corpo do “t”. O *Run-leght* convencional computa a distância para o corpo do “t” sem levar em conta a distância do corpo do “i”.

Figura 3.8 Componentes próximos mal avaliados por *Run-length*

Na tentativa de superar esta deficiência, uma heurística foi introduzida para que os componentes, que estão sobrepostos horizontalmente em pelo menos 13,3% de polegada, tenham sua distância Run-length reduzida a 60% do valor original calculado.

3.8 *Convex Hull*

Essa técnica surgiu a partir de observações feitas na técnica *Bounding Box*, que tem a desvantagem de calcular as distâncias através de retângulos (nesse procedimento perde-se informações da forma do objeto) e a técnica de *Run-length*, que por outro lado, é sensível à forma do objeto e ao posicionamento vertical dos mesmos [MN95].

Para suprir essas duas deficiências, são construídas formas que englobam os componentes não sendo retangulares e que evitem a sensibilidade do método *Run-length*. Essas formas são chamadas de “*Convex Hull*” (CH), que é o menor polígono que engloba o componente da imagem em questão. Assim, os cálculos de distâncias são realizados entre as “Cascas convexas”, que representam bem os componentes, mostrando as regiões ocupadas por eles na linha de texto.

Na Figura 3.9, são mostrados os centros de massa e cascas convexas dos três componentes da palavra *“apparent”*. A distância entre dois componentes usando a técnica *Convex Hull* está demonstrada através da letra d. Para a obtenção dessa distância, são estimados primeiramente os polígonos convexos para cada componente. Em seguida, uma linha reta é traçada entre os dois centros de massa dos componentes. A distância *Convex Hull* consiste na distância euclidiana entre os dois pontos de interseção entre a reta traçada e as cascas convexas.

Um detalhamento de todo o processo de segmentação de sentenças manuscritas em palavras usando a técnica *Convex Hull* é dado na Seção 3.8.1. Nessa, mais informações são dadas sobre a obtenção das distâncias *Convex Hull* entre os componentes conectados e também como calcular o limiar utilizado para a segmentação.

Figura 3.9 Métrica *Convex Hull*

Os experimentos em [MN95] mostram que em termos gerais de desempenho e robustez, a métrica *Convex Hull* é melhor que a *Bounding Box* e a *Run-length*. Esses experimentos também mostraram que, no grupo de imagens testadas, existiram palavras que foram corretamente segmentadas através de apenas uma das três métricas. Assim, os autores sugeriram o desenvolvimento de uma heurística para utilizar a melhor técnica de segmentação para diferentes tipos de imagens. A escolha da melhor técnica deve ser dada através de análise prévia das características.

A seguir, são detalhados todos os passos do algoritmo de segmentação Convex Hull apresentado por Marti e Bunke [MB01b].

3.8.1 Algoritmo de segmentação

Marti e Bunke [MB01b] apresentaram a segmentação de frases manuscritas com o uso da técnica *Convex Hull* através dos seguintes passos:

1. Binarize a imagem contendo a linha de texto através do algoritmo Otsu [Ots79];

2. Detecte todos os componentes conectados C da imagem;

3. Calcule as cascas convexas e os respectivos centros de gravidade de cada componente da imagem $\in C$;

4. Para cada par de componentes conectados C_x e $C_y \in C$, trace uma linha reta conectando os centros de gravidade;

5. Calcule todas as distâncias d entre os pontos P_x e P_y localizados na interseção entre a linha reta e as cascas convexas de cada par de componentes C_x e $C_y \in C$;

6. Construa um grafo completamente conectado G com os componentes C e distâncias d, para, então, calcular a Árvore de Amplitude Mínima (*Minimal Spanning Tree*) ST_{min}.

Essa representa o conjunto de distâncias mais curtas que ligam todos os componentes da imagem;

7. Calcule um limiar T_{seg} que determinará quais distâncias em d são de componentes de uma mesma palavra, e quais não são:

$$T_{seg} = \alpha \frac{W_l - W_s}{D_s},$$

W_l é a largura da linha, ou seja, a distância entre os pontos pretos mais extremos à direita e à esquerda da imagem. W_s é a largura média dos traços, calculado através da média do número de pixels pretos presentes nas linhas da imagem considerada. D_s é igual ao comprimento médio das carreiras brancas presentes na linha da imagem onde ocorre o maior número de transições preto-branco. Finalmente, α é uma constante que deve ser obtida experimentalmente;

8. Remova as arestas da ST_{min} maiores que o limiar T_{seg}. Cada remoção gera sub-árvores de componentes conectados;

9. Construa uma palavra para os componentes conectados de cada sub-árvore.

Assim, para cada componente C_i da imagem, são calculadas as distância para todos os componentes $C_1 ... C_n$. Com essas informações, um grafo completamente conectado é obtido, onde as arestas são os valores das distâncias de cada par de componentes.

O passo seguinte é a obtenção da Árvore de Amplitude Mínima para encontrar o grupo de arestas que conecta todos os componentes com a menor distância possível. Um limiar deve ser calculado para eliminar as arestas da Árvore de Amplitude Mínima que são maiores que o seu valor e, então, obtem-se sub-árvores de componentes conectados representando as palavras segmentadas da linha de texto manuscrito.

3.9 Análise de erros da técnica *Convex Hull* tradicional

Nesta seção, são apresentados alguns erros de segmentação produzidos pelo uso da técnica *Convex Hull* padrão. Duas fontes dos problemas encontrados foram detectadas através de observações: cálculo do limiar e estilo de escrita do usuário.

Para cada um dos problemas, uma heurística foi desenvolvida para melhor ajustar o limiar ou agrupar alguns componentes da imagem objetivando produzir um melhor resultado

de segmentação. É importante salientar que foi usado neste trabalho o mesmo algoritmo de cálculo de limiar T_{seg} apresentado por Marti e Bunke [MB01b].

3.9.1 Problema no limiar

São apresentados a seguir, dois problemas relacionados ao cálculo do limiar utilizado para a segmentação das palavras, são eles: Sub-segmentação e Sentenças contendo poucas palavras.

Sub-segmentação

Calcular o limiar não é uma tarefa fácil e, em alguns casos, o seu valor (quando super-estimado) faz com que o processo de segmentação gere uma única palavra ou algumas muito extensas na linha de texto. Ou seja, toda a linha de texto é segmentada como sendo uma única palavra ou mais de uma palavra é agrupada em uma só.

Na Figura 3.10, são mostrados alguns exemplos de segmentação realizada com um valor de limiar inadequado. Na Figura 3.10(a), a linha de texto foi segmentada completamente em uma única palavra e em duas palavras na Figura 3.10(b).

(a)

(b)

Figura 3.10 Problema de Sub-segmentação. (a) Uma linha de texto completa considerada como uma única palavra. (b) Uma linha de texto segmentada em duas palavras. Os *Bounding Boxes* representam as palavras encontradas pelo algoritmo de segmentação

No intuito de solucionar o problema existente, foi desenvolvida a seguinte heurística:

Heurística 1 (H1): Ao término do procedimento de segmentação da sentença manuscrita, as palavras resultantes devem ser verificadas e, em caso de haver "palavras grandes" (tamanho definido experimentalmente), o valor do limiar deve ser decrescido e o procedimento de segmentação deve ser repetido.

Nos experimentos realizados, o limiar considerado para a detecção das "palavras grandes" que produziu o melhor desempenho foi obtido da seguinte forma: $B_w = W_l * P_w$. Sendo B_w o valor em pixels do limiar, W_l é a largura da sentença manuscrita que está sendo segmentada e

P_w é o valor percentual considerado da largura da imagem.

Esta heurística pode ser descrita através do seguinte algoritmo:

1. Proceda com todos os passos de segmentação da técnica *Convex Hull* descrito na Seção 3.8.1;

2. Analise as palavras segmentadas. Existindo "palavras grandes":

   Decresça o limiar em 10%

   Execute a partir do Passo 8 do algoritmo de segmentação *Convex Hull*

   Volte para o início do Passo 2 dessa heurística.

Sentenças contendo poucas palavras

Quando o algoritmo de segmentação *Convex Hull* é utilizado para segmentar sentenças manuscritas que contém um pequeno número de palavras (quatro ou menos), o cálculo do limiar tende a gerar um valor pequeno, que, por conseqüência, induz a uma sobre-segmentação das palavras contidas na sentença manuscrita.

Esse problema é mostrado na Figura 3.11: uma estimativa errada do valor do limiar, causou a sobre-segmentação das palavras manuscritas.

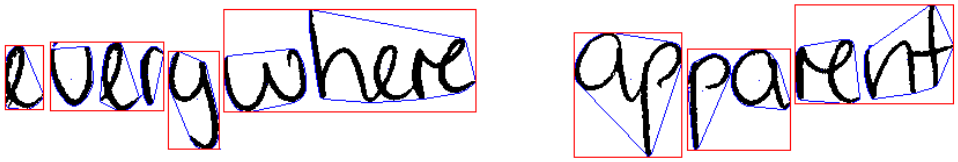


Figura 3.11 Estimativa inadequada do limiar em sentença curta (menos que quatro palavras). Os *Bounding Boxes* representam as palavras encontradas pelo algoritmo de segmentação

Heurística 2 (H2): Para imagens cujas sentenças possam conter um pequeno número de palavras, o valor do limiar deve ser aumentado antes de dar seqüência ao procedimento de segmentação.

Nos experimentos apresentados no Capítulo 5, o limiar da largura da imagem (T_w) em pixels, que foi considerado como determinante para sentenças curtas (contendo poucas palavras), foi de 800 pixels. O valor da constante utilizada para aumentar o limiar de segmentação que produziu o melhor resultado foi 5.

Essa heurística pode ser descrita através do seguinte algoritmo:

1. Todos os passos de segmentação descritos na Seção 3.8.1 devem ser executados até o cálculo do limiar T_{seg} (Passo 7);

2. Caso a imagem que contém a sentença manuscrita seja considerada como pequena (largura da imagem menor que T_w) então:

   Aumente o valor de T_{seg} (o incremento deve ser definido experimentalmente);

   Proceda com os passos seguintes de segmentação do método *Convex Hull* na Seção 3.8.1.

3.9.2 Problema no estilo de escrita

São apresentados a seguir, dois problemas relacionados ao estilo de escrita do escritor, que leva a problemas de segmentação das palavras.

Pontos do "i"

É bastante comum que letras como "i" ou "j", que são formadas por componentes separados, sejam segmentadas em palavras diferentes. Isso acontece quando a distância entres os componentes desses caracteres distam entre si acima do limiar de segmentação calculado T_{seg}.

Na Figura 3.12, é mostrado esse tipo de erro de segmentação. Nessa, a palavra *"city"* foi sobre-segmentada em três palavras porque o ponto do "i" encontra-se a uma distância do corpo do "i" maior que o limiar de segmentação T_{seg}.

Figura 3.12 Palavra *"city"* segmentada erroneamente. Os *Bounding Boxes* representam as palavras encontradas pelo algoritmo de segmentação

Heurística 3 (H3): Pequenos componentes presentes na imagem que sobrescrevem outros horizontalmente devem ser agrupados.

Assim, com o uso dessa heurística, componentes conectados da imagem cuja área é inferior ao limiar T_a (nos experimentos descritos no Capítulo 5, o melhor desempenho foi alcançado usando o valor de 600 $pixel^2$) devem ser agrupados a outro caso exista sobreposição horizontal entre eles.

Deve-se observar que o agrupamento de componentes conectados da imagem força o algoritmo de segmentação a considerá-los como pertencentes a uma mesma palavra.

Essa heurística pode ser descrita através do seguinte algoritmo:

1. Proceda com o passo inicial do algoritmo de segmentação descrito na Seção 3.8.1, encontrando todos os componentes C presentes na imagem binarizada da linha de texto manuscrita;

2. Agrupe todos os componentes pequenos (componentes cuja área é inferior ao limiar T_a) C_x, que sobrescrevem horizontalmente outros componentes $C_y \in C$;

3. Proceda com os passos seguintes do método de segmentação *Convex Hull*.

Hífen

É muito comum que, na segmentação de sentenças manuscritas, palavras separadas por hífen sejam dissociadas. Isso é considerado um erro de segmentação porque as palavras em questão devem ser consideradas como uma única.

Na Figura 3.13, é mostrado este erro de segmentação na palavra *"play-grounds"*, que foi segmentada em três diferentes palavras ao invés de uma única.

Figura 3.13 Palavra *"play-grounds"* segmentada erroneamente. Os *Bounding Boxes* representam as palavras encontradas pelo algoritmo de segmentação

Heurística 4 (H4): Componentes com características de hífen (pouca altura e uma relação entre largura/altura desejada) devem ser agrupados com os seus componentes vizinhos. Esta heurística força o método de segmentação a considerar estes componentes como pertencentes a uma mesma palavra.

A altura (h) e relação largura/altura (w/h), que determinam se o componente possui características de hífen, devem ser definidos experimentalmente. Os melhores resultados, obtidos nos experimentos descritos no Capítulo 5, foram alcançados com altura máxima de 16 pixels, e 1,6 de relação largura/altura do componente.

Essa heurística pode ser descrita através do seguinte algoritmo:

1. Encontre todos os componentes C presentes na imagem binarizada da linha de texto manuscrita (primeiro passo do algoritmo na Seção 3.8.1);

2. Agrupe todos os componentes C_i que apresentam características de hífen (relação largura/altura definida experimentalmente) com os seus vizinhos C_{i-1} e $C_{i+1} \in C$;

3. Proceda com os passos seguintes do método de segmentação *Convex Hull*.

3.9.3 Fluxograma do algoritmo de segmentação *Convex Hull* com as heurísticas

Na Figura 3.14, é mostrado um fluxograma que representa a execução do sistema de segmentação de palavras em sentenças manuscritas através do método *Convex Hull* tradicional associado às heurísticas propostas neste trabalho.

O sistema recebe inicialmente a imagem que contém a linha de texto manuscrita. Na próxima etapa, são detectados todos os componentes conectados presentes na imagem.

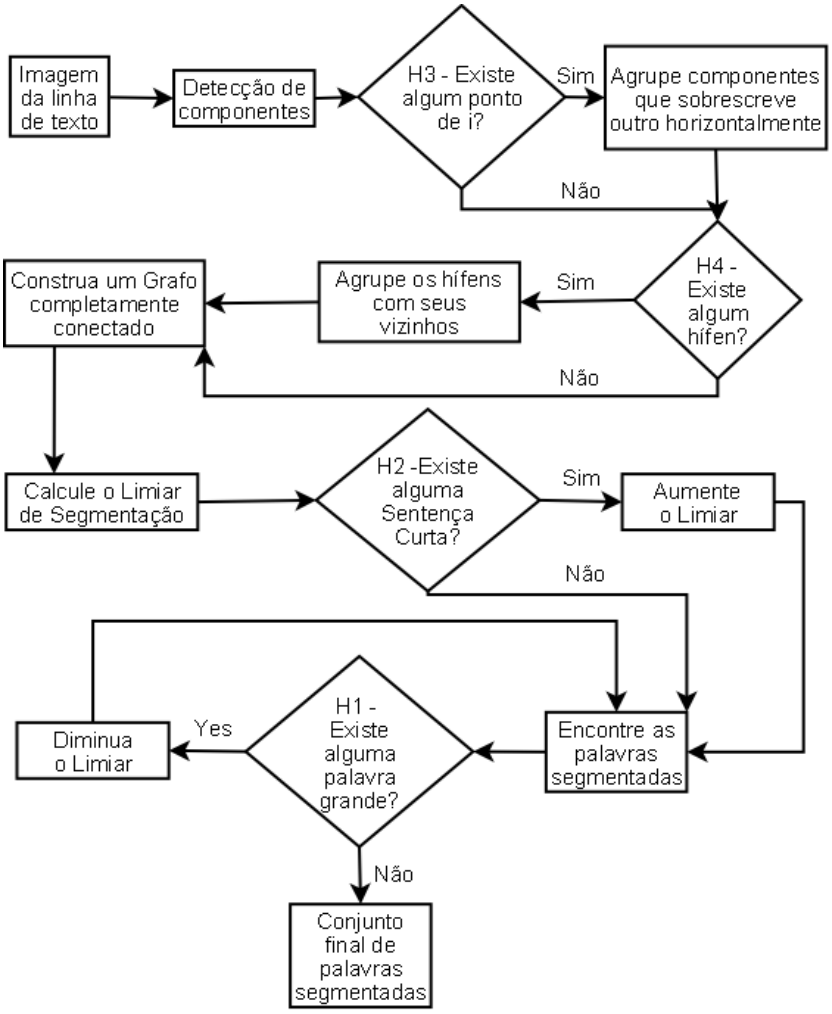


Figura 3.14 Fluxograma de execução do sistema de segmentação *Convex Hull* com heurísticas

A Heurística 3 (H3), que visa agrupar componentes pequenos que se sobrepõem a outros horizontalmente, é executada em seguida. Ainda visando agrupar componentes que devem pertencer a uma mesma palavra, a verificação sugerida pela Heurística 4 (H4) é realizada para unir componentes com características de hífen aos seus vizinhos.

Um grafo completamente conectado é formado através das informações das distâncias de

todos os componentes da imagem entre si. Em seguida, o limiar T_{seg} é inicialmente estimado.

A Heurística 2 (H2) sugere que deve-se aumentar o limiar T_{seg} caso a sentença manuscrita seja curta. O sistema realiza essa tarefa caso a largura da imagem que está sendo segmentada seja inferior ao limiar T_w utilizado por esta heurística.

Através da construção da Árvore de Amplitude Mínima (ST_{min}) e do cálculo do limiar de segmentação (passo 8 do algoritmo de segmentação *Convex Hull*), são construídas sub-árvores de componentes conectados que representam as palavras segmentadas.

Essas palavras segmentadas são então analisadas para que seja constatado que não existe nenhuma "palavra grande" conforme proposto pela Heurística 1 (H1). Sendo encontrada alguma palavra nessa condição, o limiar é diminuído para que novas sub-árvores sejam geradas a partir da ST_{min}.

Finalmente, tem-se o conjunto final de palavras segmentadas a partir da imagem que contém a frase manuscrita.

CAPÍTULO 4
Usando Redes Neurais para Segmentar Sentenças

4.1 Introdução

Técnicas tradicionais de segmentação, como *Gap Metrics*, baseiam-se em regras pré-estabelecidas para realizar a segmentação de sentenças manuscritas em palavras. É muito comum, o uso de heurísticas para melhorar o desempenho e adaptar o sistema para diferentes aplicações de segmentação [LVB07].

Outras abordagens de segmentação de palavras, que baseiam-se em um procedimento interativo de segmentação e reconhecimento como a apresentada por Morita *et al*. [MSBS04], possuem limitação de vocabulários. Isto torna tais abordagens restritas a determinados domínios de aplicação.

Tentando superar essas desvantagens, foi desenvolvida uma abordagem baseada em Redes Neurais Artificiais para segmentação de sentenças manuscritas em palavras. Por ser baseada em aprendizado, essa abordagem não necessita de regras pré-estabelecidas para a segmentação de palavras, não havendo também restrição de vocabulário.

4.2 Modelo proposto

Segmentação de palavras através do uso de Medidas de Distâncias (*Gap Metrics*) baseia-se nas distâncias entre os componentes presentes na imagem. Assim, a segmentação consiste na determinação de um valor limiar para separar as distâncias intra palavras e as inter palavras.

Ao trabalhar com Redes Neurais Artificiais (RNA), a abordagem de segmentação passou a classificar conjuntos de características extraídos da imagem, que contém a sentença manuscrita, como pertencentes a uma palavra ou espaçamento entre palavras.

Na Figura 4.1, é dada uma visão geral do modelo proposto neste trabalho para o uso de RNA para segmentar sentenças manuscritas em palavras. Considera-se que as imagens contendo o texto manuscrito já foram adquiridas previamente, e que cada imagem possui apenas uma linha de texto. Tais imagens são usadas para a segmentação das palavras através do modelo proposto.

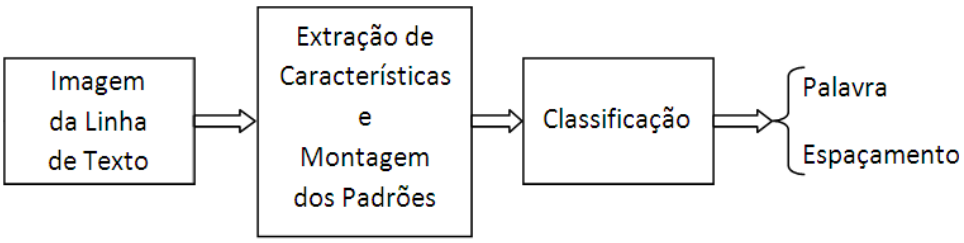


Figura 4.1 Usando RNA para segmentação de sentenças manuscritas em palavras

A primeira atividade que deve ser realizada é a extração de características da imagem. Essa etapa visa criar representações da imagem original que servirão como entrada para o classificador. Na montagem dos padrões, dentre outras atividades, é realizada a união das características extraídas com sua respectiva classificação.

Por último, é feita a construção do classificador que é responsável pela classificação dos padrões. Para o problema de segmentação de palavras manuscritas, são necessárias apenas duas classes para a classificação dos padrões: a Classe "0", que representa as palavras e a Classe "1" representando os espaçamentos entre palavras.

4.3 Visão geral do sistema

Esta seção tem como objetivo apresentar o sistema baseado em RNA implementado neste trabalho para a segmentação de palavras em sentenças manuscritas. São mostradas em detalhes as atividades executadas para a segmentação de palavras, que vão desde a criação de padrões para a Rede Neural Artificial até o cálculo do erro de segmentação.

4.3.1 Base de dados de padrões

A primeira etapa que deve ser realizada para se trabalhar com segmentação de sentenças manuscritas é a criação da base de dados de padrões de treinamento e teste da Rede Neural Artificial.

Para uma melhor explicação, é mostrado um fluxograma na Figura 4.2, que apresenta as atividades iniciais do sistema desenvolvido para a obtenção dos padrões para a RNA. A explicação de cada etapa é dada em detalhes a seguir.

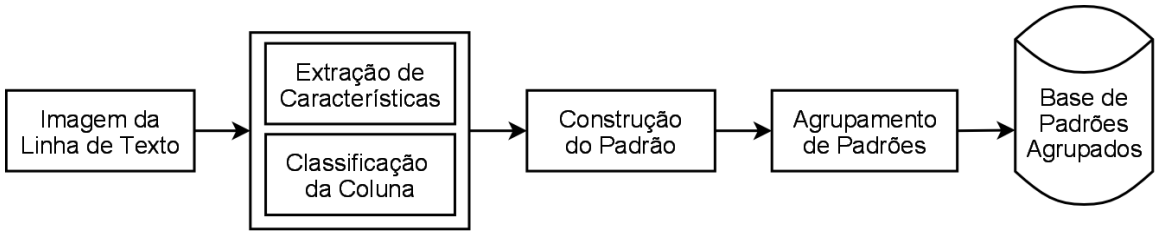


Figura 4.2 Criação de base de dados de padrões

Extração de Características

Inicialmente, o sistema recebe uma imagem contendo a linha de texto manuscrito e realiza a "Extração de Características". Uma dificuldade que surge ao trabalhar com RNA sobre imagens é como extrair um conjunto de características representativo do problema que servirá como entrada para o classificador. Decidiu-se usar um conjunto de nove grandezas geométricas baseadas em um artigo de Marti e Bunke [MB01a]. Essas características foram extraídas no sentido da esquerda para a direita (mesmo sentido de escrita dos manuscritos da base de dados IAM [MB99]) através de uma janela deslizante na largura de um pixel e altura da imagem. Assim, após a extração de características, a imagem da sentença manuscrita passa a ser representada por uma seqüência de vetores de características de dimensão nove multiplicado pela quantidade de colunas da mesma.

Na Figura 4.3, é mostrada uma imagem contendo a palavra manuscrita *"everywhere"*. A janela deslizante com largura de um pixel e os conjuntos de características extraídos das colunas da imagem estão representados nessa figura. Nesse sistema proposto, apenas um conjunto de característica por vez é inserido como entrada para a RNA.

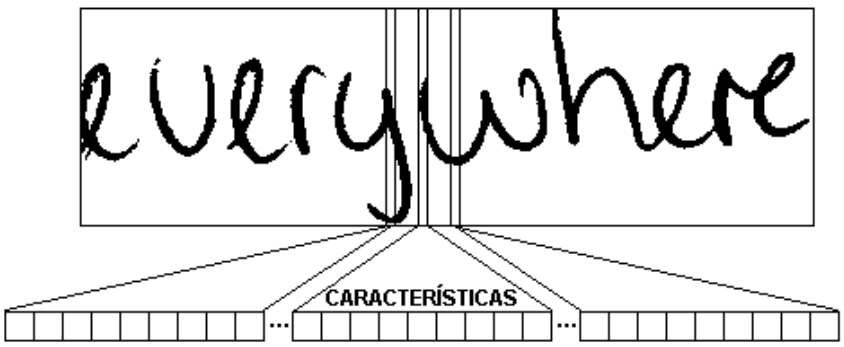


Figura 4.3 Construção do vetor de características

As nove características extraídas de cada coluna são as seguintes:

1. Peso da janela: total de pixels pretos presentes na janela.

$$f_1 = \sum_{y=1}^{m} p(x,y),$$

   x e y representam a coluna e a linha da imagem, respectivamente, p é a função que recupera o valor do pixel (0 ou 1) da coordenada da imagem e m é a altura da imagem.

2. Centro de gravidade.

$$f_2 = \frac{1}{m} \sum_{y=1}^{m} y \cdot p(x,y)$$

3. Momento de segunda ordem.

$$f_3 = \frac{1}{m^2} \sum_{y=1}^{m} y^2 \cdot p(x,y)$$

4. Posição do contorno superior: coordenada do pixel preto mais alto da janela.

5. Posição do contorno inferior: coordenada do pixel preto mais baixo da janela.

6. Gradiente do contorno superior: direção (para cima, nulo ou para baixo) obtida através da comparação das posições dos contornos superiores da coluna anterior e a atual.

7. Gradiente do contorno inferior: direção (para cima, nulo ou para baixo) obtida através da comparação das posições dos contornos inferiores da coluna anterior e a atual.

8. Transições preto-branco: número total de transições preto-branco observadas no sentido de cima para baixo da janela.

9. Pixels pretos entre os contornos superior e inferior da janela.

São mostradas na Figura 4.4 duas colunas cujas características estão sendo extraídas e suas vizinhas à esquerda respectivamente. Pode-se observar, no exemplo da esquerda, que existem de forma separada os contornos superior e inferior. Observando os contornos presentes na coluna antecedente, é possível determinar os gradientes desses contornos como “para cima” e “para baixo”, respectivamente. O total de transições preto-branco é igual a três e existe um pixel preto entre os contornos.

No exemplo da direita da Figura 4.4, pode-se constatar que as posições dos contornos superior e inferior coincidem. Outras informações que podem ser obtidas através desse exemplo

é que o gradiente de ambos os contornos é “nulo”, o número de transições preto-branco é um e não existe pixel preto entre os contornos superior e inferior.

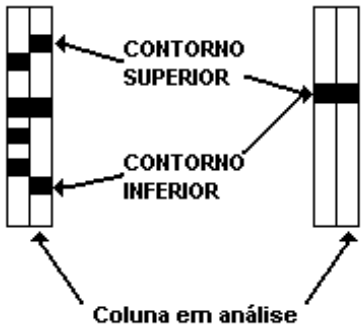


Figura 4.4 Características: contorno superior e contorno inferior

Foram adotadas algumas considerações para atribuir características em situações específicas:

1. Os gradientes dos contornos da primeira coluna da imagem devem ser considerados como “nulo’.

2. Na inexistência de contornos na coluna antecedente, os gradientes são considerados como “nulo”.

3. Na ausência de pixels pretos na coluna em análise, a posição dos contornos deve ser zero.

No modelo atual do sistema, as imagens das linhas de texto manuscrito não são tratadas com qualquer tipo de normalização, tal como: inclinação das letras, inclinação da linha de texto ou largura da imagem. Ao invés disso, foram feitas modificações em algumas das características propostas em [MB01a] na tentativa de equalizar a influência de cada uma delas na classificação das colunas. Basicamente, foram modificadas as características em questão através da adição de um fator de normalização:

- Para as características 1, 4, 5 e 9 o fator de normalização utilizado foi: $\frac{1}{\text{altura da imagem}}$.

- Para as características 2 e 3: o fator de normalização foi: $\frac{1}{\text{valor máximo alcançado pela característica}}$. O valor máximo ocorre quando todos os pixels da janela são pretos. Como as alturas das imagens utilizadas variam, é necessário calcular esse valor para cada uma delas.

Classificação da coluna

O próximo passo de execução é a obtenção da classificação desejada de cada coluna da imagem (“Classificação da coluna”). As colunas são classificadas como pertencentes à Classe “0” quando sua coordenada pertence a uma palavra, caso contrário, são classificadas como pertencentes à Classe “1”. É necessário, portanto, ter conhecimento das coordenadas das palavras presentes na imagem do texto manuscrito para a realização desta etapa.

Neste trabalho, foi possível realizar a classificação das colunas de forma automática, pois foram utilizadas imagens de sentenças manuscritas da base de dados IAM 3.0 [MB99] (maiores detalhes das imagens utilizadas são dados no capítulo dos experimentos). Esta base de dados possui meta-informações que descrevem, dentre outros itens da imagem, os *Bounding Boxes* das palavras existentes na linha de texto manuscrito.

Construção do padrão

A “Construção do padrão” é a etapa que consiste na união das nove característica de cada coluna com sua respectiva classificação. Assim, obtém-se a representação completa de um padrão para ser utilizado pela RNA.

Na Figura 4.5, são mostrados três padrões: o primeiro e o último referem-se a colunas pertencentes a palavras, portanto são classificados como Classe “0”; já o padrão central é classificado como pertencente à classe de espaçamento entre palavras (Classe “1”).

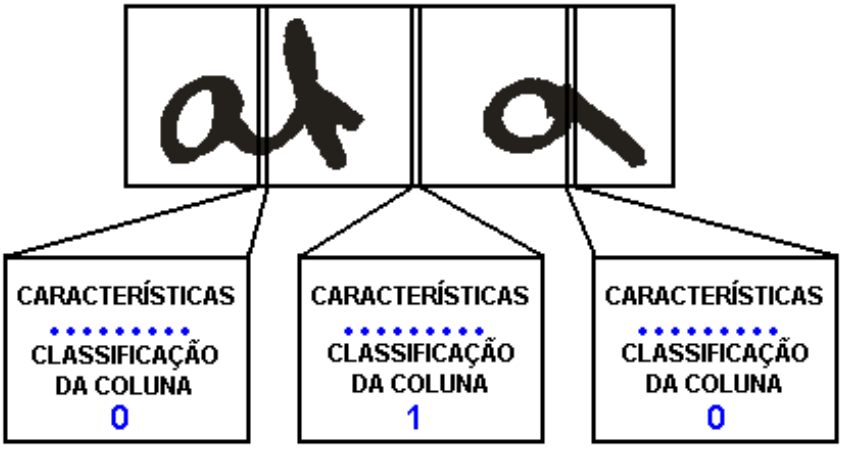


Figura 4.5 Montagem do padrão: atribuindo classes

Agrupamento de padrões

Classificar um padrão como pertencente a uma palavra ou espaçamento entre palavras é uma tarefa muito difícil quando este é observado de forma isolada, isto é, sem que seja levado em consideração as informações da vizinhança a qual ele pertence. Visando melhorar o desempenho de classificação dos padrões pela RNA, desenvolveu-se uma metodologia chamada de "Agrupamento de padrões".

Um padrão é composto originalmente por nove características e a classe a qual ele pertence. Após o processo de agrupamento de N padrões, o novo padrão irá possuir: $N * 9$ características e um identificador de classe. A nova classificação será a mesma do padrão central dentre os N padrões originais.

A Figura 4.6 exemplifica um agrupamento de padrão de tamanho três ($N = 3$). É possível observar que antes do agrupamento de padrões existiam sete padrões representados pelos conjuntos de características F_1 a F_7 e suas respectivas classes (0 ou 1). Nesse exemplo, o agrupamento de padrões de tamanho três procura unir as características de três em três padrões para gerar um novo padrão.

É mostrado na Figura 4.6 que depois do agrupamento de padrões, cinco padrões resultantes foram criados. Esses padrões estão representados através dos conjuntos de características G_1 a G_5. A classificação de cada padrão agrupado (G_i) é a mesma do padrão original central do agrupamento. Exemplificando, tem-se que o novo padrão G_3 possui 27 características (união de F_3, F_4 e F_5) e a mesma classificação do padrão F_4 que pertence à Classe "1".

O último símbolo do fluxograma na Figura 4.2 diz respeito à "Base de padrões agrupados". Essa base armazena todos os padrões que foram criados através do "Agrupamento de padrões", que são efetivamente utilizados na fase de treinamento e teste da RNA.

4.3.2 Treinamento e teste da RNA

Uma vez que as imagens que contêm as sentenças manuscritas estão agora representadas por padrões, é possível realizar o treinamento e teste da RNA. Em seguida, pode-se avaliar o sistema quanto à sua capacidade de segmentar sentenças manuscritas em palavras.

Foi utilizado neste trabalho o modelo de rede neural *Multilayer Perceptron* (MLP) totalmente conectado com algoritmo de aprendizado *Resilient Backpropagation* (RPROP) [RB93].

Essa segunda fase do sistema é ilustrada através do fluxograma presente na Figura 4.7. Primeiramente, três conjuntos de padrões são recuperados ("Recuperação dos conjuntos de padrões") da base de dados de padrões agrupados: treinamento, validação e teste. É funda-

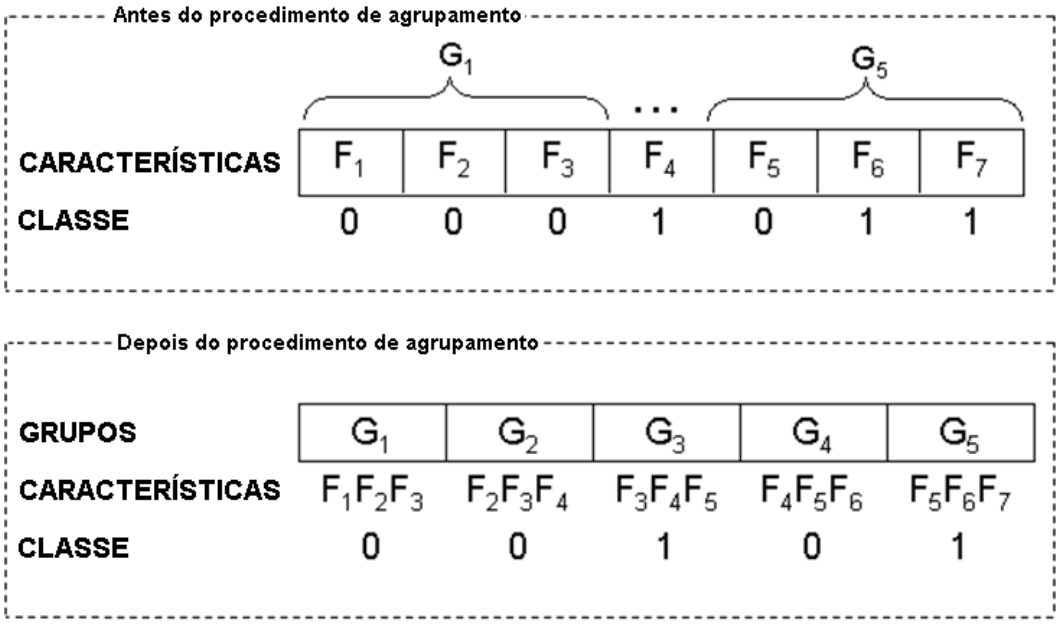


Figura 4.6 Agrupamento de padrão de tamanho três

mental que os padrões gerados a partir de uma mesma imagem fiquem agrupados mantendo a mesma seqüência das colunas correspondentes da imagem original. Assim sendo, não irão existir padrões de uma mesma imagem distribuídos em conjuntos de dados diferentes.

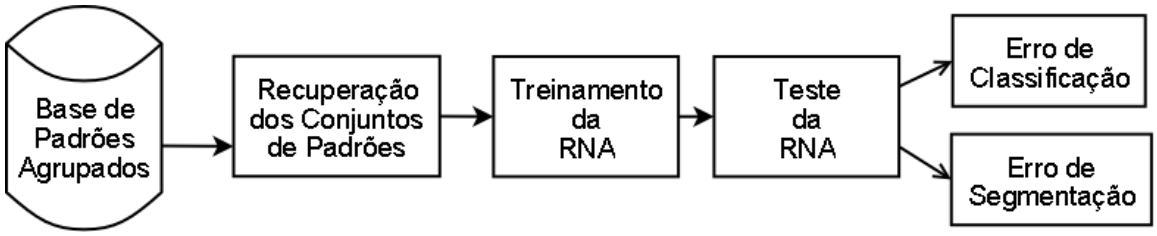


Figura 4.7 Segunda fase do sistema, que objetiva realizar o Treinamento e Teste da RNA

O treinamento (“Treinamento da RNA”) é feito utilizando os conjuntos de treinamento e validação. O teste é então realizado através do conjunto de teste (“Teste da RNA”). Como produto final dessa fase do sistema, são obtidos dois tipos de erros calculados a partir da classificação dos padrões do conjunto de teste:

- “Erro de Classificação” refere-se à percentagem de padrões classificados de forma errada;
- “Erro de Segmentação” considera o número de carreiras de padrões classificadas erroneamente. Uma carreira é representada por uma seqüência de padrões pertencentes à mesma classe.

Para o problema de segmentação de sentenças manuscritas, o erro de classificação de padrão não fornece informações conclusivas quanto à precisão do sistema, pois as seqüências de padrões devem ser analisadas ao invés de um padrão isoladamente. Note que um ou mais padrões, classificados de forma errada, pertencentes a uma mesma palavra representam apenas um erro de segmentação.

Na Figura 4.8 são mostradas as classificações de padrões de uma linha de texto fictícia. Através da classificação desses padrões, é possível exemplificar como o erro de segmentação é calculado. Pode-se observar que existem cinco carreiras de padrões, sendo três delas referentes a palavras (Classe "0") e duas referentes ao espaçamento entre palavras (Classe "1").

Observando a classificação dada pela RNA na segunda linha, é possível observar que houve erro de segmentação em duas carreiras - a primeira e a terceira. Assim, para o exemplo em questão, obteve-se um erro de segmentação de 2/5 ou 40%.

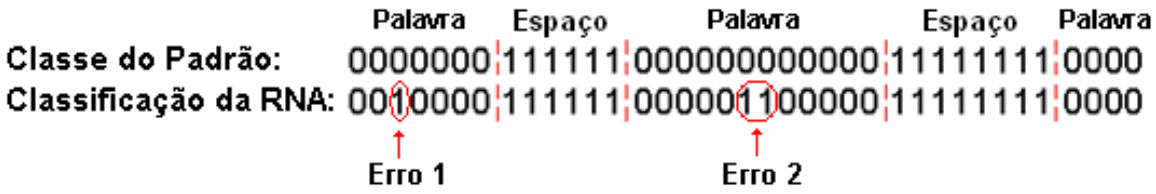


Figura 4.8 Exemplificação de classificação dos padrões de uma linha de texto, a partir da qual pode-se estimar o Erro de Segmentação

Fazendo uma comparação com o erro de classificação de padrões tem-se que esse foi de apenas 3/37 ou 8,1%. Observe com isto que mesmo tendo poucos padrões classificados de forma errônea o erro de segmentação foi alto. Assim sendo, optou-se por desconsiderar o erro de classificação de padrões realizando a análise do sistema apenas através do erro de segmentação.

No exemplo da Figura 4.8, não foi adotada margem de erro nas fronteiras das palavras. Isto significa que todos os padrões classificados errados irão ser considerados no cálculo de erro de segmentação. A margem de erro deve ser interpretada como a região de tolerância, no início e fim das palavras, ao erro de classificação. Assim sendo, caso fosse considerada uma margem de erro de uma coluna, significaria que um padrão que foi classificado de forma errada e está localizado na fronteira da palavra (início e fim da carreira de padrões) não influenciaria no Erro de Segmentação calculado (detalhes adicionais serão dados na Seção 5.3).

4.3.3 Pós-processamento

Objetivando melhorar o desempenho de segmentação, foi desenvolvida uma técnica de pós-processamento que consiste no uso de uma janela deslizante sobre a seqüência de padrões classificados para modificar sua classificação.

O funcionamento dessa técnica consiste em: se os padrões situados na vizinhança da janela têm a mesma classificação, então o sistema modifica a classificação dos padrões localizados dentro da janela para a mesma classe dos padrões vizinhos. Caso contrário, nenhuma modificação é realizada. A idéia dessa técnica é remover ruídos através da modificação da classificação dos padrões que diferem de seus vizinhos. O tamanho da janela deslizante deve ser definido experimentalmente.

Na Figura 4.9, é mostrado um exemplo de uso da janela de pós-processamento. Na linha "Antes do pós-processamento", é dada a classificação do sistema para uma seqüência de padrões e o retângulo sobre o terceiro padrão representa a janela deslizante. Em seguida, pode-se ver nessa figura o resultado do pós-processamento, que provocou a mudança de classificação do terceiro padrão (passando da Classe "0" para a "1").

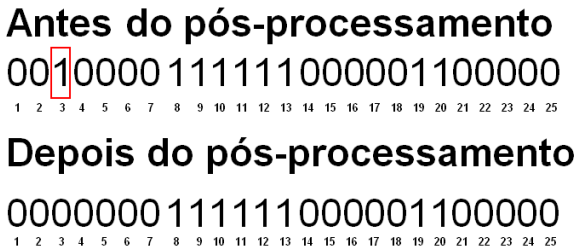


Figura 4.9 Janela deslizante de pós-processamento

O aumento da largura da janela de pós-processamento provoca maiores modificações na classificação original dada pelo sistema. Por exemplo, caso fosse adotada uma janela deslizante de largura dois, no exemplo exposto na Figura 4.9, haveria a mudança na classificação não só do terceiro padrão, mas também dos padrões das posições de número dezenove e vinte.

Na Figura 4.10, são mostrados os erros de sobre-segmentação e subsegmentação obtidos com o uso da técnica de pós-processamento. O eixo horizontal representa o tamanho da janela deslizante e o vertical a taxa de erro produzida. Tais taxas de erro foram obtidas através da média de erro de todas as linhas de texto testadas nos experimentos apresentados no Capítulo 5.

É possível observar que a taxa de erro de subsegmentação aumenta conforme a janela deslizante fica mais larga. Esse fato ocorre porque a técnica de pós-processamento força que seqüências maiores de padrões sejam classificados como uma única palavra ou espaçamento entre palavras. O comportamento oposto pode ser observado com a taxa de sobre-segmentação.

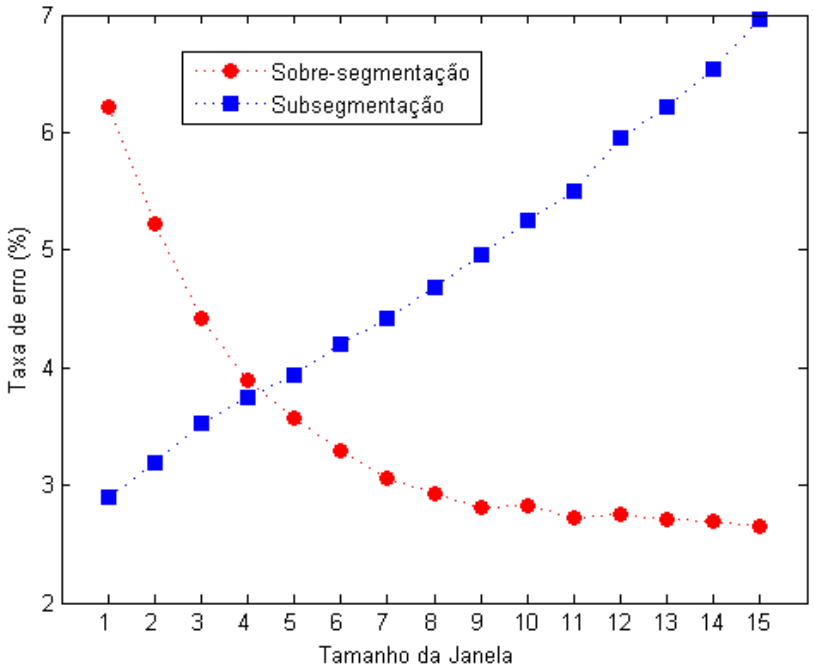


Figura 4.10 Taxa de erro de sobre e subsegmentação após a aplicação do Pós-processamento

Conforme pode ser visto na Figura 4.10, o sistema pode ser adaptado para aumentar o erro de sobre-segmentação diminuindo o de subsegmentação, ou vice-versa. Esse tipo de ajuste pode ser útil para adaptar o sistema a diferentes tipos de estilo de escrita. Usando o tamanho de janela 4, as taxa de sobre-segmentação e subsegmentação se igualam (erro de sobre e subsegmentação $\approx 4\%$).

CAPÍTULO 5
Experimentos e Resultados

5.1 Introdução

Este capítulo tem como finalidade exibir todos os experimentos realizados neste trabalho na linha de pesquisa de segmentação de sentenças manuscritas em palavras. A intenção desse estudo foi desenvolver uma técnica capaz de produzir melhores resultados de segmentação de palavras que as existentes atualmente.

Detalhes da base de dados utilizada nos experimentos são mostrados inicialmente. Em seguida, as condições dos testes realizados com os métodos de segmentação desenvolvidos neste trabalho são apresentados em detalhes: *Convex Hull* com as quatro heurísticas (explicado na Seção 3.8) e o método baseado em Redes Neurais Artificiais (apresentado no Capítulo 4).

5.2 Banco de dados IAM

A base de dados utilizada para a realização dos testes, que são apresentados neste trabalho, foi a *IAM Handwriting Database* 3.0[1]. A motivação para a criação dessa base de dados deu-se na necessidade de existência de uma base padrão para desenvolvimento, avaliação e comparação de diferentes técnicas de reconhecimento de caracteres.

A base de dados de IAM contém textos manuscritos na língua inglesa. A mesma pode ser usada para o treinamento e teste de sistemas de reconhecimento de texto, identificação do escritor, dentre outras possibilidades [MB99].

Essa base de dados é composta por formulários de textos manuscritos produzidos por 657 escritores. São 1539 páginas de textos que foram digitalizadas com uma resolução de 300 dpi (*dots per inch* - pontos por polegada) e armazenadas no formato de imagem PNG com 256 tons de cinza. Segmentando essa base em linhas, são obtidas 13.353 imagens, que ao todo são formadas por 115.320 palavras, isso dá uma média de 9 palavras por linha.

Na Figura 5.1, é mostrado um exemplo de formulário, que contém um trecho de texto em

[1]Disponível em: <http://www.iam.unibe.ch/fki/databases/iam-handwriting-database>

inglês impresso e abaixo dele o seu texto manuscrito correspondente.

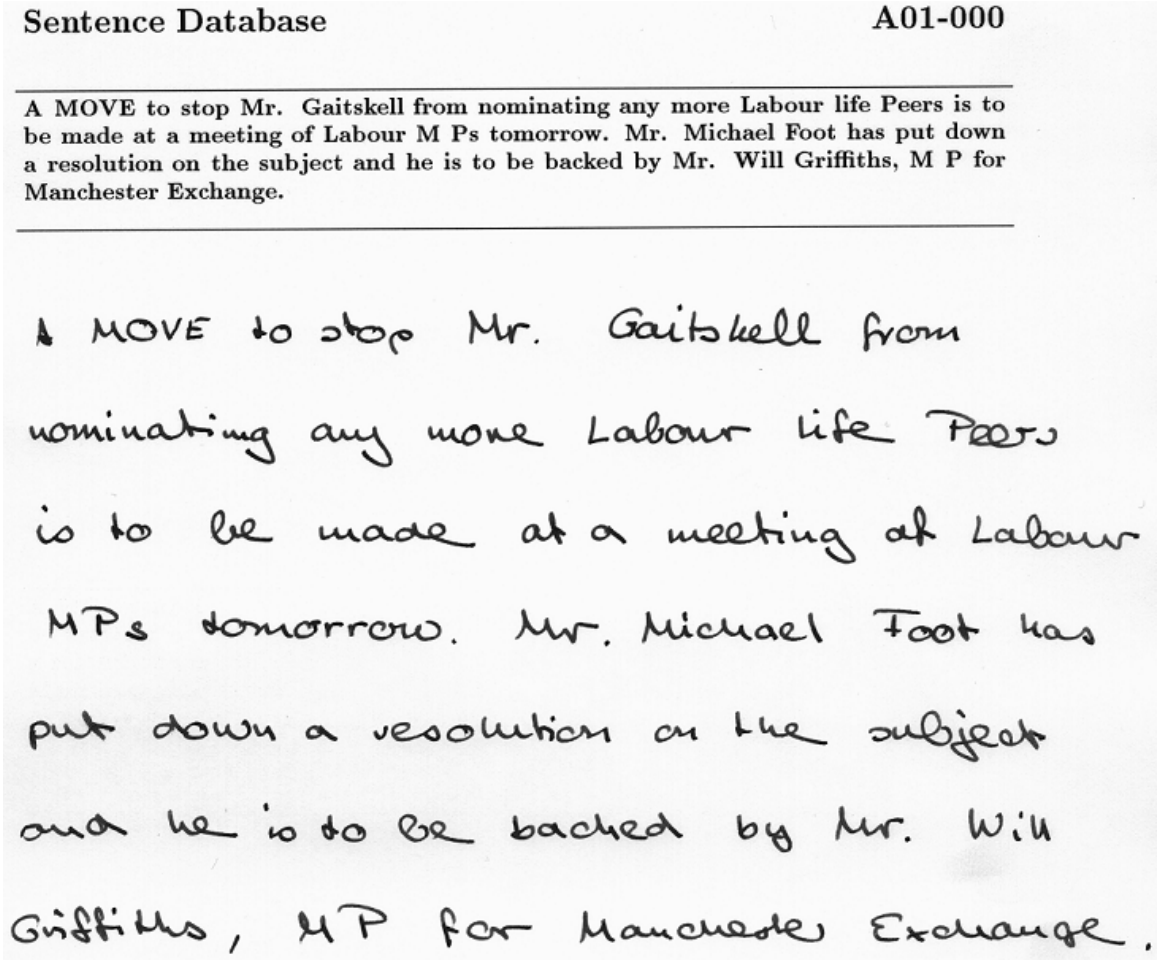


Figura 5.1 Exemplo de Formulário do banco de dados IAM

Todos os formulários, bem como as linhas de texto, palavras e sentenças extraídas, estão disponíveis para *download*, assim como o correspondente arquivo de meta-informação em formato XML.

Na Figura 5.2, estão exibidas seis imagens de escritores diferentes da base de dados IAM. Pode-se verificar a diversidade de escrita dos usuários que compuseram a base de dados IAM. A variação no estilo de escrita é uma das maiores dificuldades a ser superada na construção de sistemas de segmentação de palavras manuscritas, pois os mesmos não podem ter regras baseadas em uma única maneira de escrita.

As meta-informações existentes no arquivos XML descrevem, entre outras coisas, todas as palavras contidas nas linhas de texto. E, relativamente às palavras, são descritos todos os seus componentes conectados com suas respectivas coordenadas de localização na imagem.

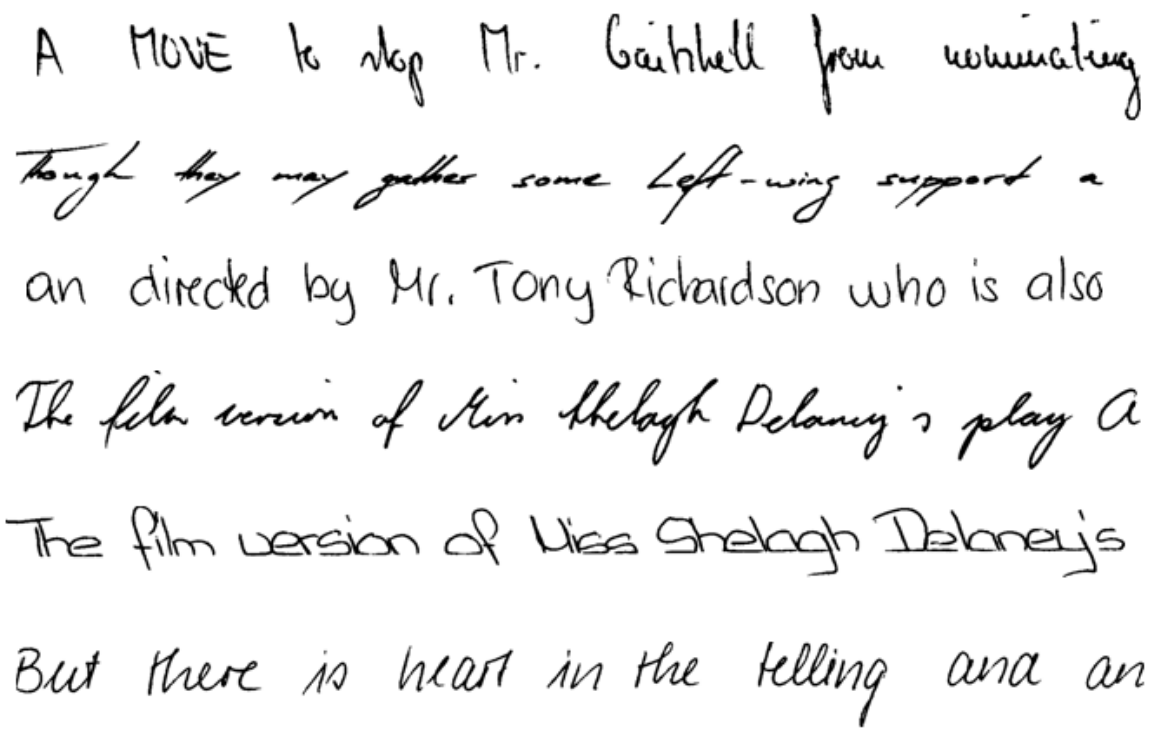


Figura 5.2 Diferentes estilos de escrita dos escritores da base de dados IAM

Dessa forma, foi possível automatizar o processo de avaliação dos algoritmos de segmentação desenvolvidos neste trabalho. Para tal, foram comparados os resultados de segmentação obtidos pelas técnicas implementadas com as informações contidas no XML.

Está exibido a seguir, um trecho do XML que descreve informações dos componentes presentes numa imagem da base de dados IAM de nome “a01-000u-05”. A qual contém a seguinte frase manuscrita: “Mr. Will.”. Pode-se observar que, no trecho do XML em questão, cada palavra é representada por uma *tag* (elemento estrutural do XML) “word”, que, por sua vez, é formada por um conjunto de *tags* “cmp” contendo o(s) *bounding box(es)* do(s) componente(s) da palavra.

A base de dados de manuscrito IAM é acessível de forma pública e gratuita para propósitos de pesquisa não comerciais. Caso a base de dados seja usada em trabalhos científicos, é exigido que seja colocada uma referência à mesma.

```
<line id="a01-000u-05" text="Mr. Will." >
  <word id="a01-000u-05-07" tag="NPT" text="Mr.">
    <cmp x="1846" y="1647" width="122" height="62" />
    <cmp x="1971" y="1702" width="10" height="11" />
    <cmp x="2086" y="1651" width="61" height="56" />
  </word>
  <word id="a01-000u-05-08" tag="NP" text="Will">
    <cmp x="2166" y="1688" width="15" height="22" />
    <cmp x="2176" y="1657" width="7" height="12" />
    <cmp x="2197" y="1662" width="36" height="48" />
  </word>
  <word id="a01-000u-05-09" tag="." text=".">
    <cmp x="2200" y="1690" width="9" height="10" />
  </word>
</line>
```

5.3 Método de avaliação utilizado

Da mesma forma que Marti e Bunke [MB01b] e Manmatha e Rothfeder [MR05], os experimentos deste trabalho foram realizados em um subconjunto de 60 formulários (c03-???[a-f][2]) da base de dados IAM [MB99]. Em ambos trabalhos, os resultados da segmentação foram manualmente avaliados (nenhuma informação adicional sobre este procedimento foi dada). Porém, comparando-se a avaliação automática com a manual, tem-se que o primeiro tipo é mais confiável por ser menos suscetível a erros humanos, além de ser menos custoso. Assim sendo, a avaliação dos experimentos foi realizada de forma automatizada.

Os formulários desse subconjunto são compostos de 549 linhas de textos manuscritos. Dentro de cada XML da base de dados IAM existe uma *tag* que indica se as coordenadas das palavras da linha de texto em questão estão corretas ou não. Portanto, para evitar distorções nos experimentos realizados, foram descartadas todas as imagens de linha de texto manuscrito cujo XML indicava que as coordenadas das palavras contidas nele não estavam corretas. Assim, passou-se a utilizar 489 imagem de linhas de texto manuscrito do total de 549 da subcategoria C03.

Adotou-se uma margem de erro de três pixels para a esquerda e para a direita nos experimentos realizados, assim uma palavra foi tida como corretamente segmentada caso as coordenadas de início e fim obtidas pelo sistema de segmentação coincidirem com as existentes no XML, ou estarem divergindo em menos que três pixels. Caso contrário, caracterizava-se um

[2]O símbolo "?" é uma expressão regular representando números.

erro de segmentação.

Na Figura 5.3, é mostrada a palavra "*pavements*" com seu *Bounding Box*. As duas linhas tracejadas em cada extremidade da palavra representam a margem de erro considerada.

Figura 5.3 A linha tracejada representa a margem de erro adotada para o procedimento de avaliação automática e o retângulo representa o *Bounding Box* da palavra informado no XML

Vale salientar também que no procedimento de avaliação automática foram desconsiderados todos os elementos de pontuação das imagens utilizadas, pois não foi implementado um algoritmo de detecção de pontuação neste trabalho. Contudo, não é difícil implementar um classificador para detecção de pontuação como o desenvolvido por Seni e Cohen [SC94]. Dessa forma, o total de palavras (rejeitando pontuação) da subcategoria C03 usadas nos experimentos foi de 3470.

5.4 Experimentos usando o método *Convex Hull*

Nesta seção, são apresentados os resultados dos experimentos realizados usando o método de segmentação *Convex Hull*. Esse método foi testado nas mesmas condições descritas em [MB01b] e [MR05] e, em seguida, testado com a integração das heurísticas propostas neste trabalho.

Na Tabela 5.1, é mostrada a taxa de erro alcançada pelo método sem a utilização das heurísticas. Nessa, pode-se observar o número total de palavras usadas nos testes, número de palavras segmentadas erradas por sobre-segmentação e subsegmentação e a taxa de erro total.

O resultado alcançado foi de 9,94% de taxa de erro. Comparando esse resultado com o obtido em [MB01b] (4,44% de erro) e [MR05] (5,60% de erro) sobre a mesma base de dados, percebe-se que não foram alcançadas as mesmas taxas. Contudo, não foi possível identificar exatamente a razão de tal divergência, porque não foram dados maiores detalhes sobre o procedimento de avaliação. Em ambos os trabalhos foram realizadas inspeções visuais para se obter os erros de segmentação, enquanto neste trabalho foi utilizado um procedimento automático.

Na Tabela 5.2, são exibidas as taxas de erro alcançadas com o método *Convex Hull* associado às heurísticas apresentadas na Seção 3.8. Usando apenas a heurística H1, a taxa de erro decresceu em 1,44 pontos percentuais, passando de 9,94% para 8,50%. O uso da heurística

Tabela 5.1 Método *Convex Hull* sem heurísticas

Número total de palavras	3470
Total de palavras sobre-segmentadas	159
Total de palavras sub-segmentadas	186
Taxa de erro total	9,94%

H3 produziu uma melhora moderada alcançando 9,34% de taxa de erro, isso representa uma diferença de 0,6 pontos percentuais quando comparada à técnica tradicional. Aplicando as heurísticas H1 juntamente com a H3, pode-se observar que o sistema alcançou uma melhora de 2,64 pontos percentuais na taxa de erro. É importante notar que este percentual de melhora foi maior que a soma das melhorias alcançadas pelas heurísticas H1 e H3 quando usadas separadamente.

Pode-se observar que o resultado tornou-se cada vez melhor com a associação das demais heurísticas. Assim, combinando H1, H2 e H3, foi produzida uma taxa de 7,30%. E, finalmente, quando todas as heurísticas foram usadas conjuntamente, a taxa de erro alcançada foi de 6,80%.

	H1	H3	H1 e H3	H1, H2 e H3	Todas as heurísticas
Número total de palavras	3470	3470	3470	3470	3470
Total de palavras sobre-segmentadas	168	132	135	126	106
Total de palavras subsegmentadas	127	192	128	128	130
Taxa de erro total	8,50%	9,34%	7,50%	7,30%	6,80%

Tabela 5.2 Método *Convex Hull* com heurísticas

Como pode ser visto, as heurísticas melhoraram o desempenho do método de segmentação *Convex Hull*. O melhor resultado foi alcançado com o uso de todas as heurísticas conjuntamente. Assim, pode-se observar uma variação da taxa de acerto de 90,06% para 93,20% quando foram associadas as heurísticas desenvolvidas neste trabalho ao método *Convex Hull*. Essa diferença significa que, na base de dados usada, o uso das heurísticas aumentou o número total de palavras corretamente segmentadas em 109 palavras quando comparada com o método tradicional. Outro ponto importante é que nenhuma heurística provocou o decrescimento da taxa de acerto, obtendo-se uma melhora no desempenho do método *Convex Hull* tradicional.

A seguir, são mostrados os parâmetros utilizados em cada heurística. Vale salientar que as imagens utilizadas nos experimentos (presentes na subcategoria C03 da base de dados IAM) apresentam o tamanho médio das palavras de 199 pixels e a largura média das imagens de 1.747

pixels.

1. Palavras grandes

   Parâmetro: percentual da largura da imagem considerado como tamanho máximo das palavras.

   Intervalo testado: de 25% até 50% variando de 5%.

   Melhor resultado obtido: percentual da largura = 35%.

2. Sentenças curtas

   Parâmetros: Largura da imagem e fator multiplicador do limiar.

   Intervalo testado:

   Largura da imagem: 500 a 850 pixels variando de 50;

   Fator multiplicador: 2 a 7 variando de 1.

   Melhor resultado obtido:

   Largura da imagem: 800 pixels;

   Fator multiplicador: 5.

3. Pontos do "i"

   Parâmetro: área do componente conectado que deve ser agrupado a outro.

   Intervalo testado: 50 a 800 pixels^2 variando de 50.

   Melhor resultado obtido: área = 600.

4. Hífen

   Parâmetro: Relação altura/largura do componente conectado.

   Intervalo testado: 16/1,6; 15/2 e 10/3.

   Melhor resultado obtido: 16/1,6.

5.5 Experimentos usando Redes Neurais Artificiais

Nessa seção, são apresentados os experimentos realizados usando o método de segmentação baseado em Redes Neurais Artificiais apresentado em [CC08a] e explicado no Capítulo 4. Para

uma melhor avaliação da abordagem proposta, a mesma base de testes (subcategoria C03 da base de dados IAM) foi utilizada nesses experimentos.

Por se tratar de uma método de segmentação que utiliza-se de um classificador que deve ser treinado, optou-se por realizar os experimentos em duas condições distintas: *user-class dependent*, onde o método de segmentação é testado em imagens do mesmo usuário usado para o treinamento do classificador; e *user-class independent* onde o método de classificação é testado em imagens de usuários diferentes daquele utilizado para o treinamento do classificador.

5.5.1 Resultados com dependência de classe

Nos experimentos realizados com dependência de classe, as imagens de cada usuário, presentes na subcategoria C03 da base de dados IAM, foram utilizadas separadamente para treinamento e teste do sistema.

O conjunto de imagens utilizado é composto por 489 linhas de texto manuscrito de 6 usuários diferentes. Em média, têm-se aproximadamente 82 imagens por usuário. O treinamento do sistema foi realizado utilizando duas imagens de cada usuário, outras duas para validação e as remanescentes foram utilizadas para teste do sistema.

Da mesma forma que os experimentos realizados com o método *Convex Hull*, foi adotada uma margem de erro de três pixels (ver Figura 5.3).

Dois parâmetros foram empiricamente definidos para se alcançar a menor taxa de erro de segmentação com o uso do método baseado em RNA:

1. Número de neurônios na camada escondida: no intervalo testado [5;50], o número que produziu o melhor resultado foi 30 neurônios.

2. Tamanho da camada de entrada: a quantidade de padrões (explicado na Seção 4.3.1) usados como entrada da RNA que produziu o melhor resultado foi 40. O intervalo testado foi [5;50].

Na Figura 5.4, pode-se observar os erros de classificação obtidos com a variação dos parâmetros da Rede Neural Artificial. Pode-se observar que a determinação da melhor configuração da RNA não foi trivial, pois não houve uma indicação clara da melhor combinação dos parâmetros conforme pode ser visto no gráfico. Decidiu-se adotar os parâmetros que produziram o menor dos erros de classificação dentre os testados.

Para uma melhor avaliação do método de segmentação baseado em RNA, foi realizada a comparação do mesmo como a técnica *Convex Hull*. A taxa de acerto de ambos os métodos

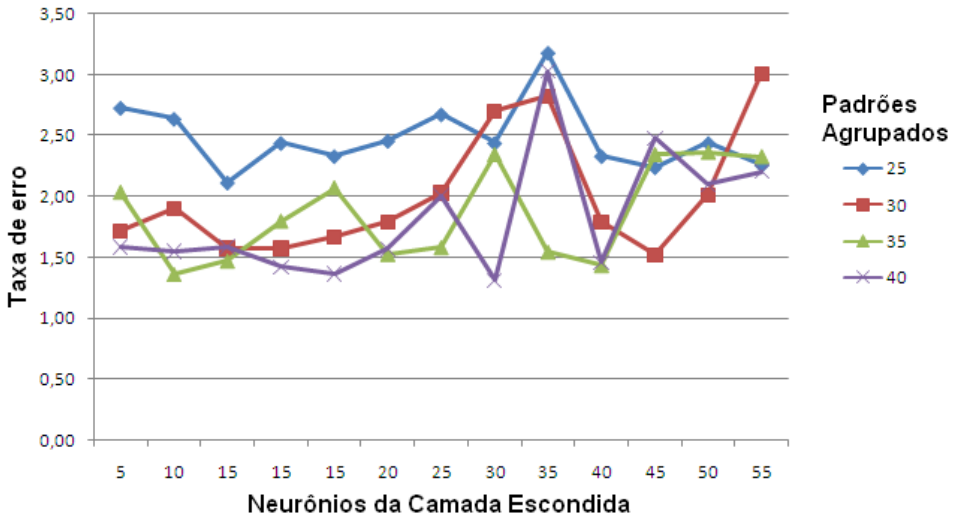


Figura 5.4 Erros de classificação obtidos através da variação dos parâmetros da Rede Neural Artificial

foram obtidas nas mesmas condições: mesmo conjunto de imagens e considerando um margem de erro de três pixels.

Na Tabela 5.3, são mostradas as taxas de erro alcançadas pelos métodos testados com dependência de classe:

1. Técnica *Convex Hull* descrita em [MB01b] na melhor configuração;

2. Método baseado em Redes Neurais Artificiais. As taxas de erro do método baseado em RNA correspondem à média de 10 execuções do sistema. Em cada execução foi realizado o treinamento da RNA. Como a inicialização dos pesos foi realizada de forma aleatória, cada execução do sistema gerou taxa de erro diferente.

É possível observar que o método baseado em RNA alcançou melhores resultados para dois dos seis usuários testados, o usuário 151 e 155. Pode-se observar também que a média geral de erro, apresentada na última linha da Tabela 5.3, indica um melhor desempenho do método *Convex Hull*. Tais observações motivaram o desenvolvimento da técnica de pós-processamento apresentada na Seção 4.3.3.

Avaliação da técnica de pós-processamento

A técnica de pós-processamento foi avaliada respeitando as mesmas condições de dependência de classe, margem de erro de três pixels sobre a subcategoria C03 da base de dados IAM. O tamanho da janela foi testado no intervalo de 1 a 15. Na Tabela 5.4, são mostradas as taxas de erro obtidas para as janelas de pós-processamento de tamanho 6 e 9.

Tabela 5.3 Taxas de erro em pontos percentuais do método *Convex Hull* e o baseado em RNA com dependência de classe

		Redes Neurais Artificiais		
ID	CH	Sobre-segmentação	Subsegmentação	Total
150	**10,34**	11,73	3,16	14,89
151	18,08	7,33	6,16	***13,49***
152	***2,66***	7,93	1,43	9,36
153	***3,62***	5,54	1,87	7,40
154	**4,53**	7,87	0,54	8,41
155	24,44	11,33	0,29	**11,62**
$\bar{x}$	**10,61**	8,62	2,24	10,86

Tabela 5.4 Taxas de erro em pontos percentuais alcançadas nos experimentos do método *Convex Hull* e o baseado em RNA com pós-processamento na condição de dependência de classe

		Janela 6			Janela 9		
ID	CH	Sobre-segmentação	Subsegmentação	Total	Sobre-segmentação	Subsegmentação	Total
150	10,34	4,13	4,10	8,23	3,30	4,53	***7,83***
151	18,08	3,00	13,55	***16,55***	2,80	16,19	18,99
152	***2,66***	2,79	1,93	4,72	2,54	2,03	4,56
153	***3,62***	3,54	2,66	6,21	3,51	3,04	6,55
154	4,53	2,29	1,53	3,82	1,54	2,02	***3,55***
155	24,44	4,00	1,45	5,45	3,16	1,93	***5,08***
$\bar{x}$	10,61	3,30	4,20	***7,50***	2,80	4,95	7,76

Pode-se observar que o método de segmentação baseado em RNA alcançou melhores resultados para quatro dos seis usuário testados: 150, 151, 154 e 155 quando comparado ao método *Convex Hull*. Outra informação que pode ser obtida através da comparação desses resultado com aqueles presentes na Tabela 5.3 é que a técnica de pós-processamento melhorou o desempenho do sistema para cinco usuários dentre os seis testados: 150, 152, 153, 154 e 155. As linhas de texto manuscrito do usuário 151 foram melhor segmentadas sem a técnica de pós-processamento obtendo 13,49% de taxa de erro conforme pode ser visto na Tabela 5.3.

A última linha da Tabela 5.4 mostra a média aritmética de cada coluna. Com 7,50%, pode-se observar que o método de segmentação baseado em RNA com janela de pós-processamento de tamanho 6 alcançou a melhor taxa de erro médio para a base de dados testada contra 10,61% da técnica *Convex Hull*.

Na Figura 5.5, são mostrados os *box-plots* das taxas de acerto obtidas com a variação do tamanho da janela de pós-processamento sobre as imagens testadas. Cada *box-plot* está relacionado ao conjunto de imagens de um usuário da subcategoria C03. Pode-se observar que quase todos os *box-plots* sugerem que um tamanho ideal de janela de pós-processamento pode ser obtido para cada usuário. Por exemplo, o tamanho de janela 9 é a melhor escolha de pós-processamento para o usuário 154, alcançando 96,45% de acerto. O mesmo comportamento não ocorre para o usuário 151, que, por sua vez, possui o maior desvio padrão e menores taxas de acerto.

5.5.2 Resultados sem dependência de classe

O sistema também foi avaliado sem dependência de classe nos experimentos do método baseado em RNA. O mesmo conjunto de imagens (subcategoria C03 da base de dados IAM) foi utilizado para os treinamentos e testes.

Para que a condição de independência de classe fosse respeitada, o sistema foi treinado usando duas linhas de texto manuscrito de um único usuário, outras duas para validação e as demais imagens dos usuários foram usadas para testar o sistema.

O método *Convex Hull*, com sua melhor configuração, testado sobre as imagens de linha de texto manuscrito da subcategoria C03, alcançou 9,94% de erro conforme pode ser visto na Tabela 5.1.

As taxas de erro relativas ao método baseado em RNA foram obtidas a partir da média de 10 execuções do sistema para cada usuário. As taxas de erro com o sistema treinado com as linhas de texto dos usuários 150, 151, 152, 153 e 155 não foram melhores que o método *Convex Hull* tradicional. Isso sugere que tais usuários possuem estilos de escrita que não se assemelham

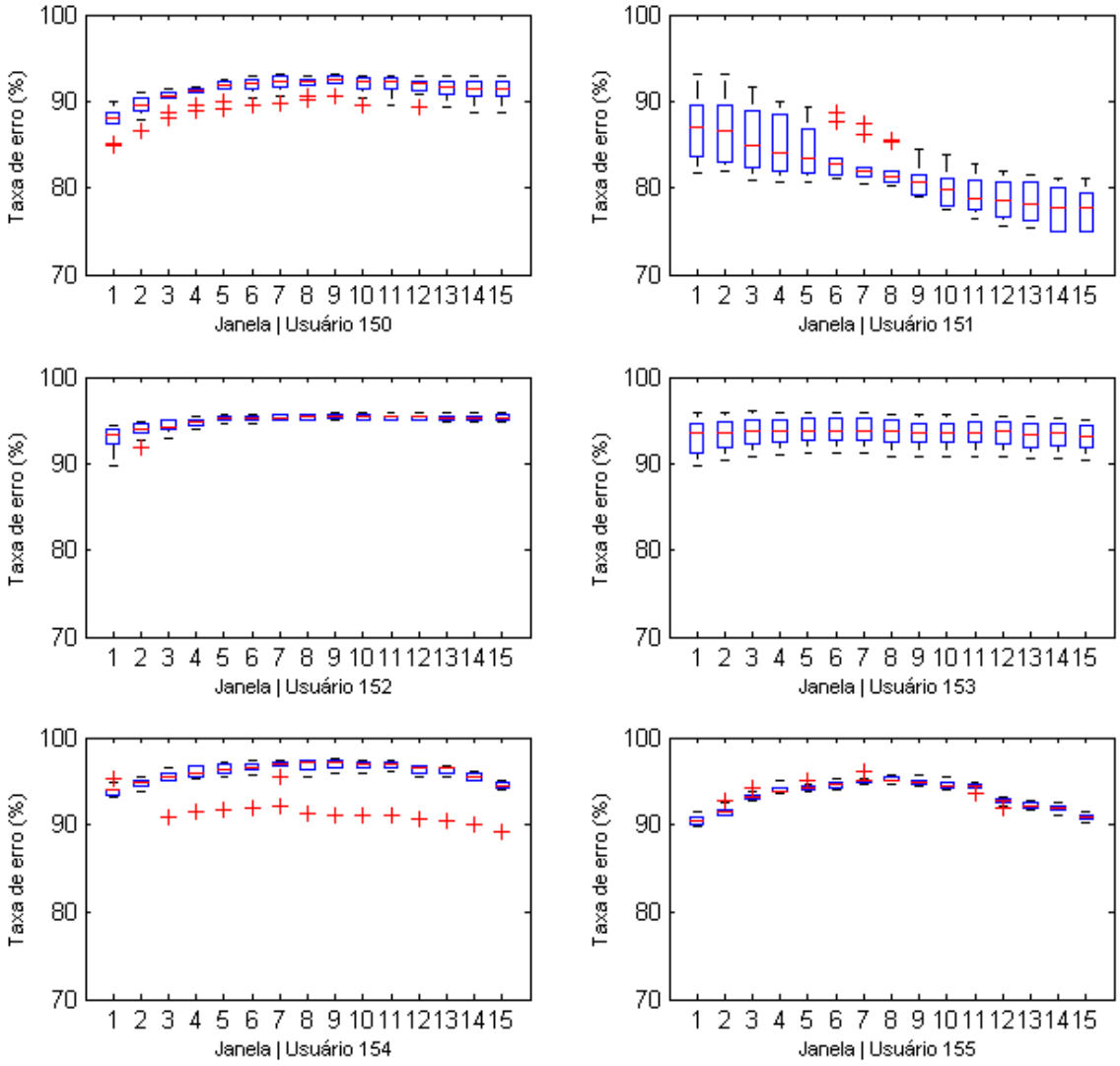


Figura 5.5 *Box-plots* das taxas de acerto com o uso da técnica de pós-processamento

Tabela 5.5 Taxas de erro em pontos percentuais nos experimentos sem dependência de classe

	Janela 0			Janela 5		
ID	Sobre-segmentação	Subsegmentação	Total	Sobre-segmentação	Subsegmentação	Total
150	8,30	5,65	13,95	3,59	8,74	***12,33***
151	17,41	4,28	21,69	8,82	7,06	***15,88***
152	19,86	10,36	30,22	12,50	14,21	***26,71***
153	26,71	4,86	31,57	17,41	7,51	***24,92***
154	13,33	0,85	14,18	6,05	2,48	***8,53***
155	39,82	16,93	56,75	33,08	21,63	***54,71***
$\bar{x}$	20,91	7,15	28,06	13,57	10,27	23,85

com os dos demais. O comportamento contrário pode ser observado com o usuário 154, onde a melhor taxa de erro foi obtida (8,53% contra 9,94% do *Convex Hull*).

5.5.3 Resultados do sistema treinado com todos usuários

Como última análise, foi utilizada uma linha de texto manuscrito de cada usuário da subcategoria C03 para treinar o sistema. Este procedimento tem como objetivo melhorar a capacidade de generalização do sistema para a segmentação de diferentes estilos de escrita.

Dependência de classe

O teste do sistema foi realizado utilizando todas as imagens remanescentes da subcategoria C03, portanto, na condição de dependência de classe. Os resultados mostraram que o método de segmentação baseado em RNA alcançou uma melhora significativa da taxa de acerto quando comparada ao método *Convex Hull*.

São mostradas na Figura 5.6 as taxas de erro obtidas pelo método *Convex Hull* sem heurísticas e o método baseado em RNA treinado com uma linha manuscrita de cada usuário. A taxa de 9,94% de erro do método CH tradicional foi superada em 3,56 pontos percentuais pelo método baseado em RNA usando uma janela de pós-processamento de tamanho 5 e 0,30 pontos percentuais sem a utilização da técnica de pós-processamento. Assim, tem-se que o melhor resultado alcançado na segmentação de palavras da subcategoria C03 da base de dados IAM foi de 6,38% de erro, obtido através do uso do sistema baseado em RNA treinado com uma linha de texto de cada usuário.

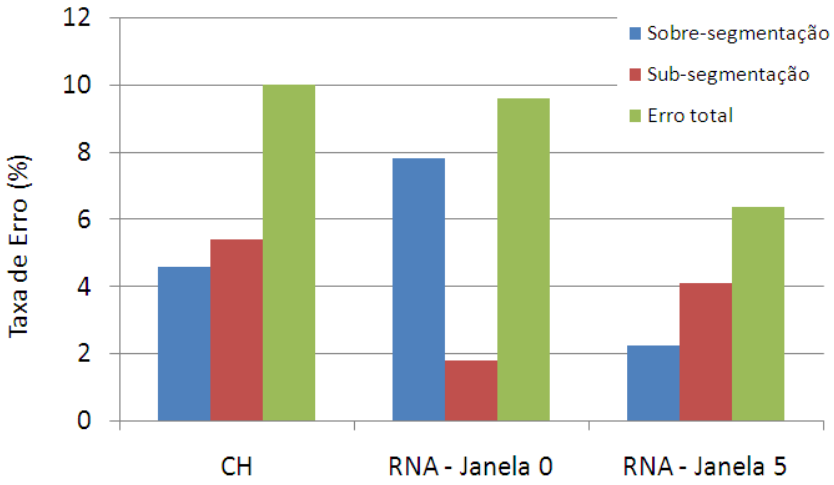


Figura 5.6 Comparação da taxas de erro do método *Convex Hull* e o método baseado em RNA treinado com uma linha manuscrita de cada usuário da subcategoria C03

Independência de classe

O teste do sistema na condição de independência de classe foi realizado utilizando as imagens presentes na subcategoria C04 da base de dados IAM. Essa é composta por formulários gerados por dezesseis escritores num total de 167 imagens de linhas de texto. As imagens que possuíam as informações de segmentação incorretamente informadas no XML foram descartadas.

Na Figura 5.7, são mostradas as taxas de erro obtidas pelo método *Convex Hull* sem heurísticas e o método baseado em RNA treinado com um linha de texto de cada usuário da categoria C03. Nesse teste, foram utilizadas as mesmas Redes Neurais que produziram as taxas de erro presentes na Figura 5.6, modificando apenas o conjunto de imagens testadas.

Pode-se observar que o desempenho do método de segmentação desenvolvido neste trabalho obteve um desempenho significativamente melhor que o *Convex Hull* tradicional, obtendo uma taxa de erro de 9,25% contra 17,78%. Outro ponto importante a ser considerado é que a técnica de pós-processamento aumentou o desempenho de segmentação em 7,77 pontos percentuais visto que a ausência da técnica produziu uma taxa de erro de 17,02%.

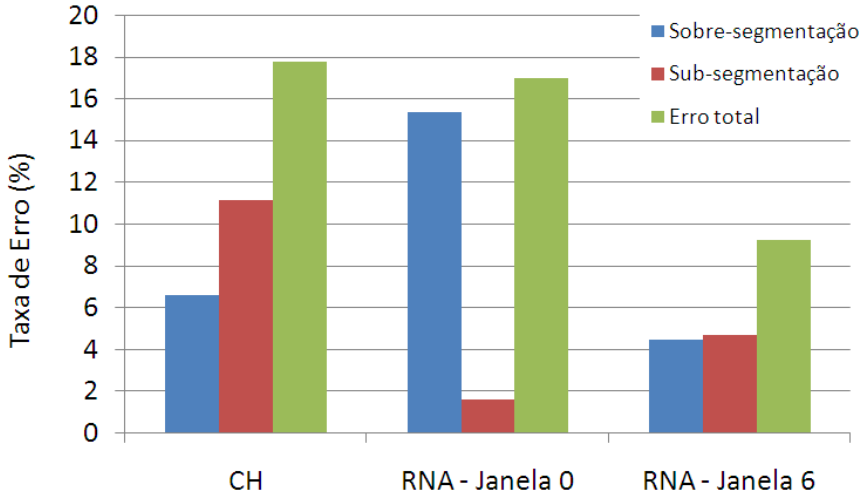


Figura 5.7 Comparação da taxas de erro do método *Convex Hull* e o método baseado em RNA treinado com uma linha manuscrita de cada usuário da subcategoria C03 e testado na subcategoria C04

5.6 Considerações finais

Neste trabalho foram apresentadas algumas técnicas de segmentação de sentenças manuscritas em palavras. Dentre os métodos baseados em distâncias, foi explicado o método de segmentação *Convex Hull* que foi apresentado em [MB01b] e posteriormente implementado e avaliado em [MR05]. Tal abordagem foi implementada e avaliada sobre o mesmo conjunto de dados nos nossos experimentos. A seguir, são citadas algumas diferenças entre a forma de testar o sistema utilizada neste trabalho e a feita em [MB01b] e [MR05]:

1. Utilizou-se neste trabalho uma avaliação automática;

2. Foram desconsideradas as imagens cujas descrições (em XML) apresentavam informações incorretas quanto às coordenadas das palavras presentes na linha de texto;

3. Na avaliação realizada neste trabalho, foram desconsideradas as informações de pontuação das sentenças manuscritas.

Todas essas diferenças explicam o porquê da divergência das taxas de erro de segmentação no método *Convex Hull*. Vale salientar que não foram dadas informações precisas sobre considerações feitas nas avaliações manuais realizadas em [MB01b] e [MR05].

Posteriormente, o método CH foi evoluído com a adição de heurísticas objetivando melhorar desempenho de segmentação. Com isto, conseguiu-se diminuir em 3,14 pontos percentuais a taxa de erro deste método.

Por fim, foi desenvolvida uma abordagem de segmentação baseada em um classificador através do uso de uma Rede Neural Artificial. Diversos testes foram realizados: cada usuário isoladamente (com dependência de classe), treinando o classificador com imagens de um usuário e testando com as imagens dos demais (sem dependência de classe) e finalmente foi utilizada uma imagem de cada usuário para treinar as RNA testando-as sobre toda a subcategoria C03 (dependência de classe) e posteriormente na C04 (independência de classe). Nestes testes foram alcançadas melhores taxas de segmentação comparativamente ao método *Convex Hull* tradicional, melhores em 3,56 e 8,53 pontos percentuais, respectivamente.

Na Tabela 5.6, podem ser vistas as taxas de erro alcançadas pelos métodos de segmentação envolvidos no estudo deste trabalho. Outra informação que pode ser vista diz respeito ao tempo de processamento gasto para a segmentação das palavras presentes nas imagens da subcategoria C03 da base de dados IAM (total de 489 linhas de texto usadas para os experimentos).

Tabela 5.6 Comparações entre a técnica CH tradicional, com heurísticas e o métodos baseado em RNA na subcategoria C03

Técnica	Taxa de erro	Tempo de Execução
CH em [MB01b]	4,44%	Não informado
CH em [MR05]	5,60%	Não informado
CH sem heurística	9,94%	1632 segundos
CH com heurísticas	6,80%	1101 segundos
Método baseado em RNA	6,38%	162 segundos

Com relação ao tempo de processamento, pode-se observar que o método baseado em RNA foi bem mais rápido que o *Convex Hull*, porém não foi contabilizado nesta informação o tempo gasto para a criação dos padrões e o treinamento da Rede Neural. A extração de características das imagens e criação dos padrões correspondentes a cada coluna das imagens foi realizada no tempo médio de 0,33 segundos/imagem (total de 55 segundos). Adicionalmente, o tempo gasto para o agrupamento dos padrões levou um tempo médio de 458 segundos/imagem (total de 76.642 segundos). Portanto, tem-se que de forma geral, a técnica de segmentação *Convex Hull* é menos custosa quanto ao tempo de processamento geral. Comparando os resultados da técnica Convex Hull tradicional com a que utiliza heurísticas, observa-se que a segunda teve um tempo de execução inferior (1632 e 1101 segundos, respectivamente). O motivo dessa diferença está no fato da diminuição dos componentes conectados da imagem que serão analisados pelo algoritmo Convex Hull devido aos agrupamentos de componentes propostos pelas heurísticas H3 e H4 (vide Seção 3.9).

Vale salientar que os métodos de segmentação foram desenvolvidos e testados através do ambiente de desenvolvimento Matlab. O desenvolvimento de tais algoritmos em outra linguagem de melhor desempenho certamente irá reduzir o tempo de execução dos mesmos, tornando mais viável a execução *On-line* do sistema baseado em Redes Neurais Artificiais.

CAPÍTULO 6
Conclusão

6.1 Introdução

O presente trabalho trata do problema de segmentação de sentenças manuscritas em palavras, onde foram desenvolvidas técnicas cujo objetivo era produzir o menor erro de segmentação possível. Foram apresentados diversos problemas presentes em texto manuscritos tais como: variação do tamanho dos caracteres, não uniformidade de espaçamento entre as letras, ruídos, inclinação, dentre outros. Todos esses fatores mostram que o desenvolvimento de sistema de segmentação não é uma tarefa de fácil resolução.

Neste capítulo serão apresentadas as principais contribuições desse trabalho bem como alternativas para trabalhos futuros.

6.2 Contribuições

Duas abordagens de segmentação foram desenvolvidas e testadas neste trabalho. A primeira baseia-se na métrica de distância *Convex Hull* e a segunda é uma abordagem baseada em Redes Neurais Artificiais.

6.2.1 Método *Convex Hull* com heurísticas

O algoritmo *Convex Hull* é bastante conhecido para o tratar o problema de segmentação de sentenças manuscritas. O baixo custo computacional e o bom desempenho obtido por este, quando comparado a outras técnicas de segmentação, foram fatores que motivaram o seu uso neste trabalho.

Contudo, esta abordagem apresenta algumas deficiências, que foram mostradas no Capítulo 3: i) problemas na estimativa do limiar para uma melhor segmentação das palavras pertencentes à sentença, e; ii) o algoritmo original apresenta problemas de adaptação a características inerentes ao estilo de escrita do usuário. Em seguida, foram apresentadas quatro heurísticas objetivando superar tais problemas.

Os resultados experimentais mostraram a efetividade das heurísticas propostas. O algoritmo de segmentação de frases manuscritas proposto por Marti e Bunke [MB01b], no qual este trabalho baseou-se, foi comparado com a abordagem associada a heurísticas. Em todos os casos, as heurísticas superaram a abordagem original. O melhor resultado foi obtido através da combinação de todas as heurísticas propostas. Essa configuração alcançou 6,80% de taxa de erro, que representa uma melhora de 3,14 pontos percentuais quando comparado ao resultado obtido com a versão original do algoritmo de segmentação.

6.2.2 Método de segmentação baseado em Redes Neurais Artificiais

A segunda abordagem de segmentação desenvolvida neste trabalho utiliza-se de um classificador para segmentar as palavras manuscritas. A seguir, são apresentadas algumas problemas que motivaram o desenvolvimento do método baseado em Redes Neurais Artificiais:

- Um problema comum existente na abordagem CH está na necessidade do uso de heurísticas para otimizar e adaptar o sistema às diferentes aplicações de segmentação de sentenças manuscritas;
- Outras abordagens que utilizam-se de procedimento interativo de segmentação e reconhecimento, como a apresentada por Morita *et al*. [MSBS04], possuem limitação de vocabulário.

Os experimentos realizados para a avaliação da abordagem baseada em RNA foram executados nas seguintes condições:

1. Duas linhas de texto manuscrito de um único usuário utilizadas para treinamento:

   Dependência de usuário: nesse teste, as linhas de texto manuscrito utilizadas para o teste do sistema eram do mesmo usuário do treinamento. Através dos resultados obtidos, pôde-se observar que o método de segmentação baseado em RNA apresentou melhores resultados que o algoritmo de segmentação *Convex Hull* para a maioria dos usuários testados (4 de 6 usuários).

   Independência de usuário: nesse teste, linhas de texto manuscrito, de outros usuários foram utilizadas para o teste do sistema. O método CH alcançou melhores resultados que a abordagem baseada em RNA para a maioria dos usuário testados (5 de 6 usuários).

2. Uma única linha de texto manuscrito de cada usuário utilizada para o treinamento:

Nessa condição de teste, o método baseado em RNA alcançou uma melhora significativa da taxa de acerto do sistema. O método CH alcançou 90,04% de precisão contra 93,62% da abordagem baseada em RNA.

O fato da abordagem de segmentação que usa RNA ser baseada em aprendizagem sugere que esta é mais apropriada para diferentes tarefas de segmentação quando comparada a outras técnicas.

6.2.3 Método de avaliação automática das técnicas de segmentação

Uma grande dificuldade que houve durante a avaliação das técnicas de segmentação desenvolvidas neste trabalho foi o fato de existirem poucas informações relacionadas ao método de avaliação das técnicas apresentadas por Marti e Bunke [MB01b] e Manmatha e Rothfeder [MR05]. Pôde-se encontrar apenas indicações do uso do método de inspeção visual sobre determinado conjunto de imagens. Nenhuma informação quanto à consideração de margens de erro ou eliminação de pontuação foi encontrada.

Neste trabalho, foram apresentadas em detalhes a forma de obtenção das taxas de erro. Além de mostrar que o uso da base de dados IAM possibilitou o desenvolvimento de uma abordagem automática de avaliação com o uso de meta-informações contidas em arquivos XML. Este tipo de avaliação é menos suscetível a erros humanos aumentando, portanto, a confiabilidade dos resultados experimentais.

6.3 Trabalhos Futuros

Durante o desenvolvimento do presente trabalho, verificou-se a oportunidade de desenvolvimento de alguns itens como os listados a seguir:

- O conjunto de informações geradas na fase de extração de características foi obtido através de uma janela de largura de um pixel. O uso de diferentes larguras de janela pode gerar características que melhorem o desempenho do classificador, já que informações de vizinhança das colunas serão consideradas;

- Desenvolver um método de escolha automática do tamanho da janela de pós-processamento. Tal abordagem poderia ser desenvolvida através do uso do conjunto de validação. Assim, o tamanho ideal pode ser determinado com a configuração que gera o menor erro de classificação;

- Testar a abordagem baseada em Redes Neurais Artificiais sobre um conjunto de dados com mais usuários para uma avaliação mais abrangente.

Referências Bibliográficas

[CC08a] César A. M. Carvalho and George D. C. Cavalcanti. An artificial neural network approach for user class dependent off-line sentence segmentation. *International Joint Conference on Neural Networks. Hong Kong*, pages 2723–2727, June 2008.

[CC08b] César A. M. Carvalho and George D. C. Cavalcanti. Using an artificial neural network approach for off-line sentence segmentation. *International Conference on Frontiers in Handwriting Recognition. Montreal*, pages 427–432, August 2008.

[CF95] Rafael C. Carrasco and Mikel L. Forcada. A note on the nagendraprasad-wang-gupta thinning algorithm. *Pattern Recognition Letters*, 16:539–541, 1995.

[CRT98] K. Chinnasarn, Y. Rangsanseri, and P. Thitimajshima. Removing salt-and-pepper noise in text/graphics images. *IEEE Transaction on Circuits and Systems*, pages 459–462, November 1998.

[DH72] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[Hu62] Ming-Kuei Hu. Visual pattern recognition by moment invariants. *IEEE Transactions on Information Theory*, 8(2):179–187, 1962.

[JC94] B. K. Jang and R. T. Chim. Reconstructable parallel thinning. *Thinning Methodologies for pattern recognition*, pages 181–217, 1994.

[Jr73] G. D. Forney Jr. The viterbi algorithm. *Proceedings of the Institute of Electrical and Electronics Engineers*, 61:268– 278, 1973.

[KG98] G. Kim and V. Govindaraju. Handwritten phrase recognition as applied to street name images. *Pattern Recognition*, 31:41–51, 1998.

[LVB07] F. Lthy, T. Varga, and H. Bunke. Using hidden markov models as a tool for handwritten text line segmentation. *International Conference on Document Analysis and Recognition. Curitiba*, 1:8–12, 2007.

[MB99] U. Marti and H. Bunke. A full english sentence database for off-line handwriting recognition. *International Conference on Document Analysis and Recognition. Bangalore*, pages 705 – 708, 1999.

[MB01a] U. V. Marti and H. Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[MB01b] U.V. Marti and H. Bunke. Text line segmentation and word recognition in a system for general writer independent handwriting recognition. *International Conference on Document Analysis and Recognition. Seattle*, pages 159–163, 2001.

[MGS05] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):23–35, 2005.

[MN95] U. Mahadevan and R. C. Nagabushnam. Gap metrics for word separation in handwritten lines. *International Conference on Document Analysis and Recognition. Montreal*, 1:124–127, 1995.

[MR05] R. Manmatha and J. L. Rothfeder. A scale space approach for automatically segmenting words from historical handwritten documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1212–1225, 2005.

[MSBS04] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Segmentation and recognition of handwritten dates: an hmm-mlp hybrid approach. *International Journal on Document Analysis and Recognition*, 6(4):248–262, 2004.

[Nib86] W. Niblack. *An Introduction to Image Processing*, pages 115–116. Prentice-Hall, 1986.

[NK05] Ifeoma Nwogu and Gyeonghwan Kim. Word separation of unconstrained handwritten text lines in pcr forms. *International Conference on Document Analysis and Recognition. Seoul*, pages 715–719, 2005.

[NM94] H. Nishida and S. Mori. A model-based split-and-merge method for character string recognition. *Document Image Analysis*, pages 209–226, 1994.

[O'G92] L. O'Gorman. Image and document processing techniques for the rightpageselectronic library system. *Pattern Recognition*, 2:260 – 263, 1992.

[OK97] Larry O'Gorman and Rangachar Kasturi. *Document Image Analysis: An Executive Briefing*. IEEE Computer Society, July 1997.

[Oli99a] L. S. Oliveira. Estado da arte dos sistemas de reconhecimento automático de manuscritos. Technical report, Pontifícia Universidade Católica do Paraná, March 1999.

[Oli99b] L. S. Oliveira. Processamento automático de digitos manuscritos. Technical report, Pontifícia Universidade Católica do Paraná, December 1999.

[Ots79] N. Otsu. A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9:62–66, January 1979.

[PGS99] J. Park, V. Govindaraju, and S. N. Srihari. Efficient word segmentation driven by unconstrained handwritten phrase recognition. *International Conference on Document Analysis and Recognition. Bangalore*, pages 605–608, 1999.

[PS00] Réjean Plamondon and Sargur N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):63–84, 2000.

[Rab89] L. R. Rabiner. A tutorial on hidden markov and selected applications in speech recogniton. *Proccedings of the IEEE*, 77:257–258, 1989.

[RB93] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *In Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591, 1993.

[SC94] G. Seni and E. Cohen. External word segmentation of off-line handwritten text lines. *Pattern Recognition*, 27:41–52, 1994.

[SP00] J. Sauvola and M. Pietaksinen. Adaptive document image binarization. *Pattern Recognition*, 33:225–236, 2000.

[SRI99] Tal Steinherz, Ehud Rivlin, and Nathan Intrator. Offline cursive script word recognition – a survey. *International Journal on Document Analysis and Recognition*, 2(2-3):90–110, December 1999.

[SS04] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, January 2004.

[ZS84] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 27(3):236–239, 1984.