FACULDADE DE TECNOLOGIA DE SÃO PAULO

Luiz Fernando Miranda Vieira Lins

RECONHECIMENTO ÓTICO DE CARACTERES (OCR) E ANÁLISE DE SISTEMAS OCR BASEADOS EM CÓDIGO ABERTO

São Paulo
2012

FACULDADE DE TECNOLOGIA DE SÃO PAULO

Luiz Fernando Miranda Vieira Lins

RECONHECIMENTO ÓTICO DE CARACTERES (OCR) E ANÁLISE DE SISTEMAS OCR BASEADOS EM CÓDIGO ABERTO

Monografia apresentada à Faculdade de Tecnologia de São Paulo (FATEC-SP) para obtenção do Grau de Tecnólogo em Processamento de Dados.

Orientador: Prof. Dr. Maurício Amaral de Almeida

São Paulo
2012

"Reparta o seu conhecimento.
É uma forma de alcançar a imortalidade."
Dalai Lama

AGRADECIMENTOS

Muito especialmente, desejo agradecer ao meu orientador Prof. Doutor Maurício Amaral de Almeida, pela disponibilidade, atenção dispensada, paciência e profissionalismo,. um Muito Obrigado.

À minha esposa, Ana Maria, pelo apoio e compreensão por todo este período.

À minha família e meus pais.

Aos meus amigos, em especial Augusto e Mara pelas madrugadas adentro jogando Catan.

Aos meus colegas de FATEC.

A todos os demais.

RESUMO

Um dos grandes anseios do ser humano é criar máquinas que possam fazer, cada vez mais, o trabalho braçal e repetitivo. As primeiras formas de escrita datam de mais de 5.000 anos, nosso alfabeto ocidental começa a aparecer a partir do séc. VII a.C.

Na Suméria começam a aparecer as primeiras formas de prensa para impressão de rubricas. Por volta de 1439, Johannes Gutemberg cria os chamados “tipos móveis” e cria a prensa móvel. Assim, criou-se uma forma mecânica para cópia de textos.

No final do séc. XIX aparecem as primeiras experiências para captura de imagem e isso caminhou de forma tímida até antes da Segunda Guerra Mundial.

Após a Segunda Guerra, houve o interesse comercial para desenvolvimento e pesquisa. Com a melhoria dos sistemas de hardware com relação a processamento e custo, bem como o aumento de pesquisas voltadas para o reconhecimento de caracteres criou-se sistemas de OCR cada vez mais eficientes e baratos.

O que antes eram equipamentos caros, viraram pacotes de software disponíveis a computadores pessoais e com a internet e filosofias de desenvolvimento coletivo, apareceram vários softwares OCR que são gratuitos e possuem código-fonte aberto (*open source*).

Este trabalho destina-se a descrever alguns métodos de OCR bem como apresentar e avaliar três modelos disponíveis gratuitamente e cujo código-fonte é aberto, *open source.*

Palavras-chave: Caracteres, Inteligência Artificial, OCR, Open Source.

ABSTRACT

One of the greatest desires of the human being is to create machines that can do more and more manual labor and repetitive. The earliest forms of writing date back over 5,000 years, our Western alphabet begins to appear from century. VII B.C.

In Sumer begin to appear the first forms of press for printing headings. Around 1439, Johannes Gutenberg created the so-called "movable type" and creates the printing press. Thus was created a mechanical way to copy texts.

At the end of the XIX century, early experiences appear to capture this image and walked timidly up to before the Second World War.

After WWII, there was commercial interest for research and development. With the improvement of hardware systems with respect to processing and cost, as well as the increase in research aimed at created character recognition OCR systems are becoming more efficient and inexpensive.

What were once expensive equipment, software packages became available for personal computers and the Internet and collective philosophies of development, appeared several OCR software that are free and have open source code.

This paper is intended to describe some methods of OCR as well as present and evaluate three models available for free and whose source code is open, open source.

Keywords: Characters, Artificial Intelligence, OCR, Open Source.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

ANN	Artificial Neural Network
RNA	Redes Neurais Artificiais
SPR	Reconhecimento Estatístico de Padrões
OCR	Reconhecimento Ótico de Caracteres
UNLV	Universidade de Nevada, Las Vegas
URL	Uniform Resource Locator
PDF	Portable Document Format
GPL	GNU Public License
DFKI	Centro de Pesquisa Alemão para Inteligência Artificial

SUMÁRIO

1. Introdução

1.1. Motivação

A utilização de máquinas para replicar funções humanas, como a leitura, é um sonho antigo. A partir dos anos 1950, a leitura por máquinas passou de sonho para realidade. O uso de Reconhecimento Ótico de Caracteres (em inglês, OCR) tornou-se uma das aplicações mais bem sucedidas da tecnologia no campo de reconhecimento de padrões de inteligência artificial [AIM,2000]. Com barateamento dos computadores e aumento da capacidade de processamento [RUMELT, 2002], os sistemas de OCR estão cada vez mais acessíveis às pessoas e vários novos algoritmos bem como técnicas para pré-processamento, extração de dados além de poderosos métodos de classificação foram criados [CHERIET, 2007].

Já existem muitos sistemas comerciais e gratuitos disponíveis para uma variedade de aplicações: desde a transformação de textos originalmente em papel para um formato digital sem a necessidade de digitação do mesmo até passando por uso em radares de trânsito para identificação das placas dos automóveis e sua verificação junto às bases de dados municipais para verificação de que está tudo quites com a prefeitura e estado [ROBERTI, 2010]. O entendimento dos processos de OCR e descrição de como tais métodos são utilizados em alguns sistemas *open source* são a principal motivação deste trabalho.

1.2. Objetivos

Ao procurar conteúdo sobre técnicas de OCR bem como publicações sobre análise de softwares, foi basicamente encontrado referências a conteúdos (livros e sites) no exterior e sem maior explicação sobre o que realmente fazem estes sistemas.

Assim, a principal finalidade deste trabalho é realizar uma compilação sobre os diversos métodos que compõe um sistema de OCR, posteriormente apresentar três sistemas disponíveis gratuitamente no formato de open source para depois discorrer sobre estes três fazendo uma comparação mais aprofundada entre eles.

1.3. Organização

O restante deste trabalho está organizado conforme descrito a seguir:

1.3.1. Introdução

Apresentação da motivação para realização do trabalho, objetivos principais e organização do conteúdo.

1.3.2. História do OCR.

Apresentação de conceitos de OCR, como origem, fundamentos e alguns algoritmos utilizados durante a história.

1.3.3. Metodologia de OCR

Apresenta os componentes de um sistema de OCR e descreve as várias abordagens utilizadas em cada um deles.

1.3.4. Apresentação dos Sistemas OCR open source

Apresenta quais são os sistemas baseados em open source que serão analisados, como obtê-los, instala-los e opera-los do ponto de vista de um usuário.

1.3.5. Classificação dos Sistemas OCR open source

Apresenta a classificação dos algoritmos dos sistemas citados com base nas metodologias previamente expostas.

1.3.6. Conclusão

Conclusão do trabalho com base nos resultados obtidos.

2. História do OCR.

Neste capítulo é apresentado um contexto histórico e a evolução da tecnologia de OCR assim como serão mostradas algumas técnicas utilizadas quer eram as principais características de cada geração de OCR.

2.1. Nascimento do OCR.

Reconhecimento Ótico de Caracteres (OCR) possui uma história relativamente longa. Esta tecnologia foi inventada no início dos anos 1800, quando foi patenteada como ajuda de leitura para pessoas cegas. Em 1870, C. R. Carey patenteou um sistema de transmissão de imagem (um scanner de retina) usando um mosaico de fotocélulas e, em 1890, P.G. Nipkow inventou o scanner sequencial, aonde uma imagem era analisada linha a linha [MERESHA, 2008].

Em 1929, Gustav Tauschek obtinha a patente de sua "Maquina de Leitura" [TAUSCHEK, 1935].

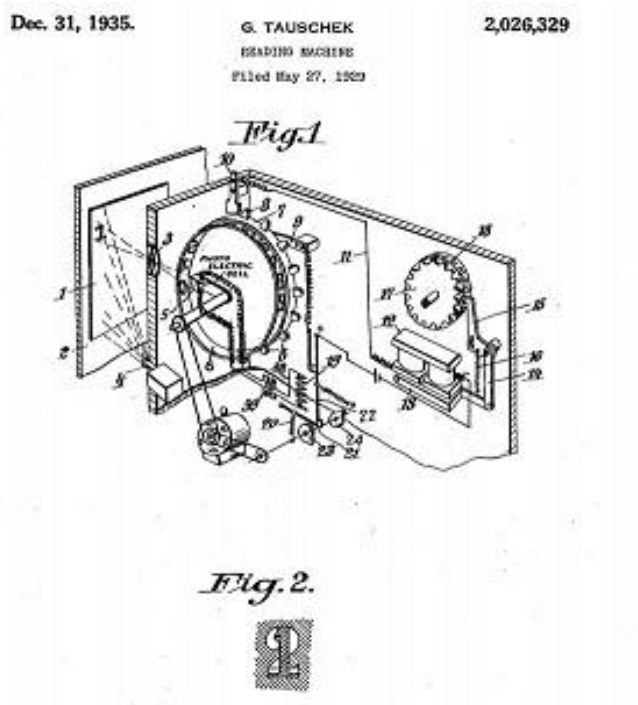

Figura 1 – Patente da Máquina de Leitura de G. Tauschek

Em 1933, Handel também registra uma patente de tecnologia de OCR ao apresentar sua "Máquina Estatística" [HANDEL, 1933].

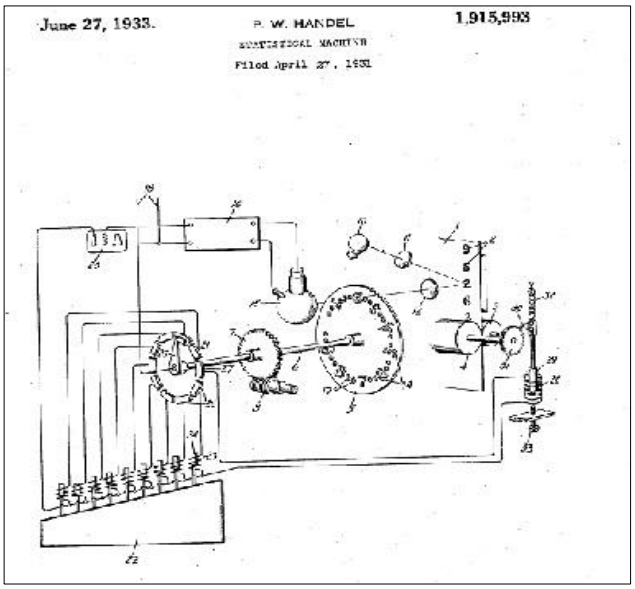

Figura 2 - Máquina Estatística de Handel

Estes são os primeiros conceitos da ideia de OCR da forma que conhecemos. Naquele tempo, algumas pessoas imaginavam uma máquina que poderia ler caracteres e numerais. Isso permaneceu no campo das ideias até que a era dos computadores chegou a partir dos anos 1950. A partir daí, podemos dividir os sistemas de OCR em gerações.

2.2. Primeira Geração

Os sistemas de OCR comerciais apareceram no período entre 1960 e 1965, que pode ser chamado de primeira geração de OCR. Esta geração de equipamentos OCR era caracterizada principalmente pela leitura de um formato restrito de letras. Os símbolos eram especialmente desenhados para leitura dos equipamentos e os primeiros não tinham um formato muito natural. Com o passar do tempo, máquinas capazes de ler múltiplas fontes apareceram e conseguiam processar até dez tipos diferentes de fontes. A variedade de fontes era limitada pelo método de reconhecimento do padrão utilizado, identificação de padrões, que compara a imagem do caractere com uma biblioteca de protótipos de imagens para cada caractere de cada fonte [EIKVIL, 1993].

Os sistemas de OCR de primeira geração tinham como restrição principal o desempenho dos hardwares disponíveis. O primeiro computador comercial, UNIVAC I, foi

instalado e entrou em funcionamento em 1951 [MORI, 1992].

Para que se conseguisse simplificar a quantidade de dados a ser processada, em 1956 Marvin H. Glauberman e Robert C. Kelner usaram uma fenda registradora magnética para transformar uma informação bidimensional (um caractere) em uma informação unidimensional [MORI,1992]:

Um caractere impresso é escaneado por um fotodetector através de uma fenda. A luz refletida no papel permite o fotodetector segmentar o caractere calculando a proporção da quantidade de preto detectada pela fenda. O valor desta proporção é enviado para um registrador que converte o valor analógico para digital. Estas amostragens seriam, então, comparadas com um padrão utilizando a soma total das diferenças entre cada valor de amostragem e o valor correspondente do padrão[COOPER,1984].

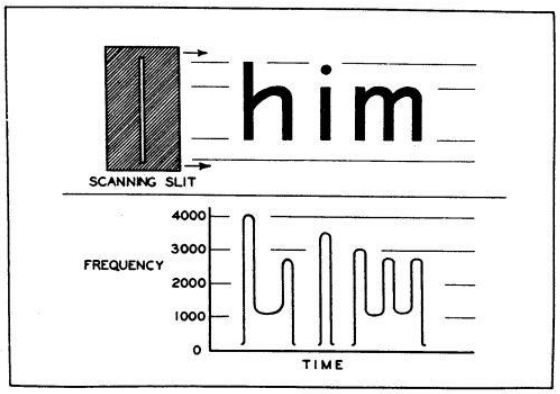

Figura 3 Método da Fenda e Gráfico de Proporção de Preto

Em 1957 a Solatron Electronics Group Ltd. anunciou o primeiro sistema de OCR baseado na técnica de "peephole". O sistema tinha o nome de ERA (Electric Reading Automaton).

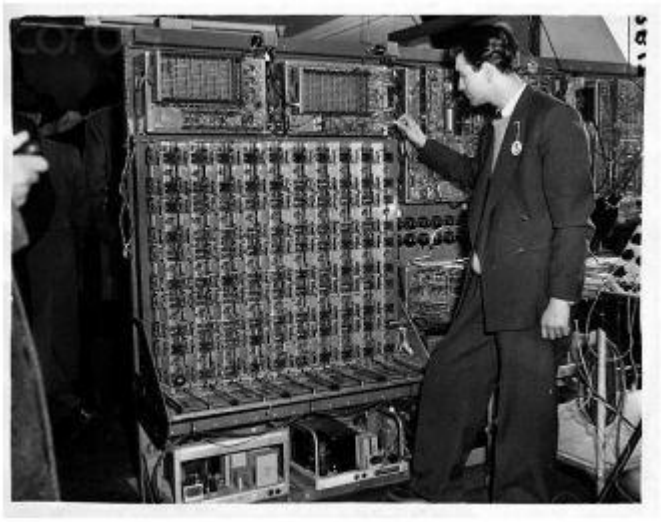

Figura 4 ERA

A técnica de “peephole” é o método mais simples do ponto de vista lógico de identificação de padrões: Pixels de diferentes zonas de um caractere binarizado são comparados com os caracteres padrões. Binarização é um importante pré-processamento na tecnologia de OCR. Idealmente, um caractere possui dois níveis de densidade. A um deles é atribuído o valor zero (“0”) e ao outro o valor um (“1”). Para ilustrar melhor, utilizaremos a letra “A”, imagine um plano de memória bidimensional aonde os valores binarizados desta letra “A” são armazenados seguindo uma regra previamente definida. Idealmente, um ponto preto sempre terá o mesmo valor e o mesmo vale para a os pontos brancos (isto é, desconsidera-se a existência de tons de cinza). Então, pixels são escolhidos e seus valores analisados e confrontados com os padrões de letras já armazenados pelo sistema (Figura 2) [MORI, 1992].

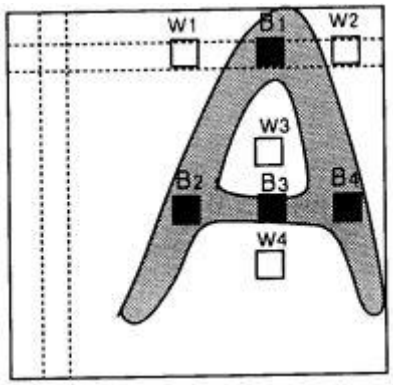

Figura 5 Técnica de Peephole

2.3. Segunda Geração

De acordo com [EIKVIL,1993], a segunda geração dos sistemas de OCR apareceu nos meados da década dos anos 1960 e início dos anos 1970. Estes sistemas eram capazes de reconhecer caracteres que tenham sido impressos por outras máquinas comuns e também alguma capacidade de reconhecimento de caracteres escritos à mão. Ainda considerando caracteres escritos a mão, o reconhecimento se restringia a algumas poucas letras e símbolos além de numerais.

O primeiro e mais famoso sistema deste tipo era o IBM 1287, que foi exibido pela pri IBM- 1287meira vez na Feira Mundial de Nova Iorque em 1965 (Figura 6).

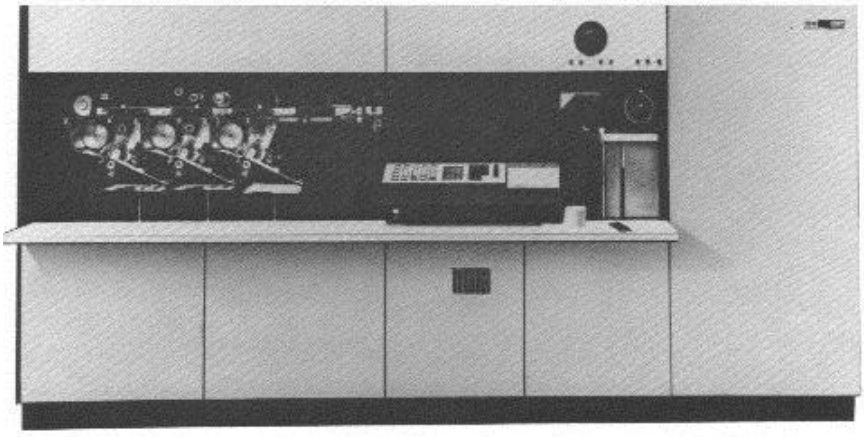

Figura 6 IBM- 1287

Também, neste período, a Toshiba desenvolvia seu primeiro equipamento para separação de cartas por código de endereçamento postal e a Hitachi criava a primeira máquina de OCR de alto desempenho e baixo custo.

O método abordado pela Toshiba foi o de análise estrutural

Neste período já era feito um esforço significante na área da padronização. Em 1966 um estudo aprofundado dos requisitos para OCR havia sido completado em um padrão americano de caracteres para OCR estava definido: o OCR-A (Figura 7).

Figura 7 OCR A.

Esta fonte era muito estilizada e seu desenho foi pensado tanto para facilitar o reconhecimento ótico quanto a leitura por pessoas.

Os europeus também criaram o seu padrão, OCR-B (Figura 8), que possuía um desenho mais natural que o padrão americano. Algumas tentativas foram feitas a fim de combinar as duas fontes em um padrão, mas ao invés disso, máquinas capazes de ler os dois padrões começaram a surgir.

Figura 8 OCR B.

2.4. Terceira Geração

Nos meados dos anos 1970, o desenvolvimento dos sistemas de OCR passaram a focar o problema dos documentos com documentos de má qualidade e grandes conjuntos de caracteres impressos e escritos à mão, como os caracteres chineses [MORI, 1992]. Ao mencionar documentos, se trata de qualquer um que contenha palavras, como nomes, endereços e comandos. Estes objetivos foram parcialmente atingidos durante este período já apareciam alguns sistemas comerciais com estas características entre 1975 e 1985. Com o foco em custo baixo e alto desempenho, alavancado pela crescente disponibilidade de processamento a um custo cada vez menor,por conta do advento de chips com tecnologia LSI, que proporcionou a criação de CPUs de velocidade maior e memórias ROM.

Ainda, de acordo com [MORI,1992], o problema de leitura de documentos de má qualidade de impressão foi resolvida tanto pela Toshiba quanto pela IBM, com abordagens diferentes:

No caso da Toshiba, utilizou-se uma solução baseada em teoria. Eles construíram, em 1971, um equipamento, OCR-V100, que tinha internamente um algoritmo de correlação usando o método de múltiplas similaridades. O ponto forte deste método é a criação de uma máscara (ou dicionário). Isso é feito ao se utilizar a média dos dados armazenados que sejam pertencentes à uma mesma classe de caracteres.

A IBM, em 1975, desenvolveu o IBM-1975. Este equipamento utilizava o modelo de correspondência lógica, uma evolução do método de "peephole" já descrito anteriormente. Pela primeira vez se utilizaria medições de características dos caracteres como parte do processo de reconhecimento de caracteres.

Foi também na terceira geração que começaram a aparecer os primeiros sistemas de OCR na forma de pacotes de software para computadores PC. Um dos primeiros softwares a fazer isso foi o OmniPage, da Caere Corporation [New York Times, 1988].

2.5. Os anos 1990 e Adiante

A partir de 1990 as mudanças nos sistemas de OCR ocorreram devido a melhores desempenhos das workstations UNIX e nos computadores pessoais. Ainda o escaneamento de imagem e pré-processamento eram feitos via hardware, o software já tinha implementação de uma boa parte da etapa de reconhecimento e isso estava já disponível aos computadores em geral. Algoritmos de reconhecimento passaram a ser escritos nas linguagens C e C++, fazendo com que mais pessoas pudessem desenvolver estes algoritmos, sofisticando-os e permitindo uma participação maior da comunidade acadêmica nas pesquisas. Técnicas de reconhecimento de documentos escritos à mão passaram a ser alvo de estudos e desenvolvimento e os resultados destes esforços passaram a ser utilizados por bancos na leitura de cheques e de leitores para os correios. Sistemas mais avançados de análise em camadas permitiram o uso de reconhecimento ótico em uma maior variedade de formulários comerciais [FUJISAWA, 2008].

3. Metodologia de OCR

3.1. Componentes de um sistema de OCR.

De acordo com [EIKVIL, 1993], um sistema de OCR é composto basicamente de:

- Escaneamento Ótico;
- Localização e Segmentação;
- Pré-processamento;
- Extração de Características;
- Classificação;
- Pós-processamento;

Todo o processo ocorre da seguinte maneira: Um documento é digitalizado por um escâner ótico, o que foi digitalizado então passa por um processo de localização das regiões que contém o texto, dentro de cada região é então segmentada e são extraídos os símbolos. Cada símbolo extraído pode passar por um pré-processamento que envolve a eliminação de todo o ruído possível a fim de facilitar a extração das características deste símbolo. O símbolo é então identificado ao comparar as características pertencentes a ele com as descrições das classes de símbolos que são obtidas através de uma fase de aprendizado, ocorrida antes do início do processo de OCR. Por fim a

informação contextual é utilizada para reconstruiar as palavras e números do texto original. Isso pode ser visto melhor na Figura 9.

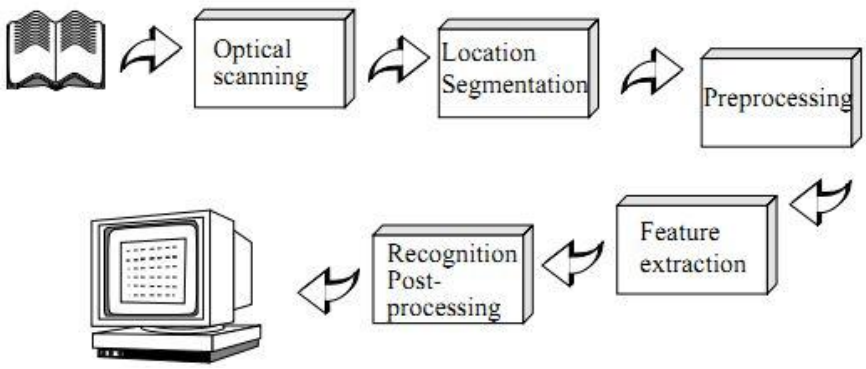

Figura 9 Metodologia de um sistema de OCR

3.2. Escaneamento Ótico

É através do escaneamento óptico que obtemos uma imagem digitalizada a partir de um documento. Um escâner ótico consiste basicamente de um mecanismo de transporte combinado com um sensor que iráconverter a luz refletida pelo documento em níveis de cinza. Comumente documentos impressos são em preto-e-branco por isso é uma boa prática converter uma imagem de múltiplos níveis (colorida) em uma imagem de níveis de cinza ou até preto-e-branco. Este processo, chamado de "thresholding" é feito para otimizar o uso de memória e esforço computacional[EIKVIL,1993].

Apesar de muito simples, o processo de "thresholding" é fundamental para o resultado do processo de reconhecimento. Ele é feito da seguinte maneira: um valor limítrofe é definido. Níveis de cinza abaixo deste valor são tratados como preto e valores acima deste valor são tratados como branco. Em documentos com alto contraste entre o preto e o branco, um nível limítrofe predefinido pode ser suficiente, mas a maioria dos documentos não possui esta característica. Assim, os valores limítrofes são modificados dinamicamente através de métodos adaptativos a fim de obter um bom resultado. Estes métodos levam em conta outras propriedades do documento como contraste e brilho [EIKVIL, 1993]. Um exemplo de como o "thresholding" influencia a qualidade do resultado da imagem digitalizada pode ser visto na Figura 10, Figura 11 e Figura 12.

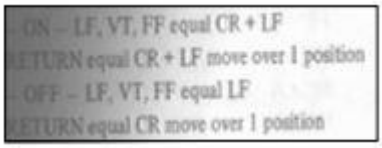

Figura 10 – Imagem escaneada com má iluminação

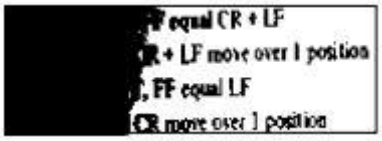

Figura 11 Imagem corrigida com valor fixo de Thresholding

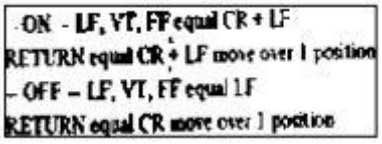

Figura 12 – Imagem corrigida usando método adaptado de Thresholding

3.3. Localização e Segmentação

Segmentação é um processo que determina os constituintes de uma imagem. É necessário localizar as regiões dos documentos aonde os dados foram impressos e distingui-los de figura e gráficos [EIKVIL,1993].

O ponto-chave de um algoritmo de segmentação é encontrar um meio de medir a diferença entre os pixels que pertencem ao texto e os pixels do fundo [CHEN, LUETTIN, SHEARER, 2000].

Os métodos de segmentação podem ser genericamente divididos em explícitos e implícitos. A segmentação explícita tenta separar as imagens das palavras nas bordas dos caracteres e comumente resulta em uma enorme lista de candidatos a pontos de segmentação. Assim, a segmentação explícita também é conhecida como "oversegmentation". Por outro lado, a segmentação implícita corta a imagem em vários pedaços de comprimentos iguais. Os pedaços, cada qual representado por seu vetor de características, são agrupados em caracteres por reconhecimento[CHERIET, 2007].

Os principais problemas em segmentação podem ser divididos em quatro grupos [EIKVIL,1993];

- Extração de caracteres unidos ou fragmentados – Isso pode levar a interpretação de dois ou mais caracteres unidos como sendo um único ou que um pedaço de um caractere é um símbolo por si só.
- Distinção entre texto e ruído – Pontuação pode ser tomado como ruído ou imagem de fundo e vice-versa.
- Interpretar imagem ou gráfico como texto – Isso leva a algo que não existe para ser reconhecido.
- Interpretar texto como sendo gráfico ou imagem – O texto não será interpretado pelo sistema de OCR na sua totalidade. Isso geralmente ocorre quando gráficos e caracteres estão unidos.

3.4. Pré-processamento

A imagem resultante do processo de digitalização pode conter uma certa quantidade de ruído. Dependendo da resolução do scanner e da técnica de "thresholding" utilizada, os caracteres podem aparecer manchados ou incompletos. Alguns destes defeitos resulta numa menor taxa de reconhecimento. A fim de diminuir e/ou até eliminar este problema, é utilizado um pré-processamento para melhorar os caracteres digitalizados [EIKVIL, 1993].

O pré-processamento envolve os seguintes processos[CHERIET,2007]:

- Suavização e remoção de ruídos;
- Análise do grau de enviesamento e correção;
- Análise de inclinação;
- Normalização do caractere;
- Análise/definição de contorno;
- Afinamento.

Abaixo segue o detalhamento de cada um dos processos de acordo com o livro de Cheriet:

3.4.1. Suavização e remoção de ruídos

Operações de suavização nos níveis de cinza das imagens dos documentos são utilizados para "blurring" e redução de ruídos. "Blurring" é usado na etapa de pré-processamento para a remoção de pequenos detralhes em uma imagem. Em imagens de documentos que foram binarizadas (só possuem preto e branco) . A suavização é utilizada para reduzir o ruído ou para diminuir o tamanho da borda dos caracteres como por exemplo preencher pequenos vazios ou remover pequenos desalinhamentos nos contornos dos caracteres. Suavização e remoção de ruídos podem ser feitas através de filtragem. Filtragem é uma operação feita com a análise de vizinhança, aonde o valor de qualquer pixel na imagem de saída é determinado ao se aplicar algum cálculo aos valores dos pixels que estão na vizinhança. Existem duasformas de abordagem: a linear e a não-linear. A linear tem este nome pois é feita uma combinação linear dos valores do pixel central com os que são da vizinhança [CHERIET,2007].

$$\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Figura 13 - Exemplo de aplicação de filtro para suavização.
O valor da soma de todos os coeficientes é igual a 1.

Figura 14 - Aplicação de filtro em uma imagem escaneada. A imagem processada (direita) possui contornos mais suaves e é mais suave que a original (esquerda).

3.4.2. Análise do Grau de Enviesamento e Correção

3.4.2.1. Grau de Enviesamento

Grau de enviesamento é a o valor de inclinação da linha de base do texto escaneado com relação à horizontal. Isso ocorre durante o processo de escaneamento do documento aonde a folha que contém o texto acaba inclinando em relação ao sensor e, consequentemente, todo o texto aparece com um grau fixo de enviesamento em relação ao eixo x

Figura 15 - Exemplos de enviesamento de imagem

.

A detecção da existência do enviesamento e respectiva avaliação do grau pode possui dois principais métodos: a detecção de componentes conectados (grosso modo, pode-se dizer que um componente conectado é o equivalente a um caractere) e determinar a média dos ângulos das linhas que conectam os centroides destes componentes. Outro método diz respeito à análise do perfil da projeção. Os métodos relacionados a perfil da projeção são mais diretos. Neste procedimento, um documento é projetado em vários ângulos e a variância no número de pixels pretos projetados em cada linha são determinados. A projeção paralela ao correto alinhamento irá fornecer a maior variância [CHERIET,2007].

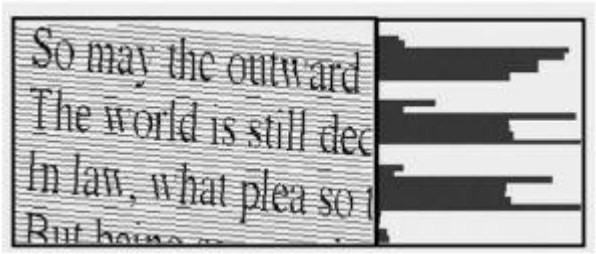

Figura 16 Exemplo de projeção de perfil e variância do preto em documento enviesado

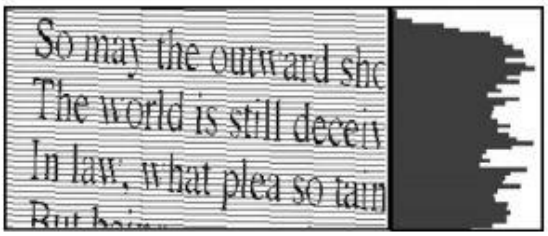

Figura 17 Projeção do perfil na direção do enviesamento. A variância do preto é máxima, indicando ser este o grau de enviesamento do documento

3.4.2.2. Correção do Enviesamento

Após determinação do ângulo de enviesamento, a página deve ser rotacionada a fim de corrigir isso. O algoritmo de rotação tem que ser ao mesmo tempo rápido e preciso. Possuindo o valor do ângulo, a posição do pixel em coordenadas (x,y), o novo posicionamento dos pixels é regido pela fórmula [CHERIET,2007].

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

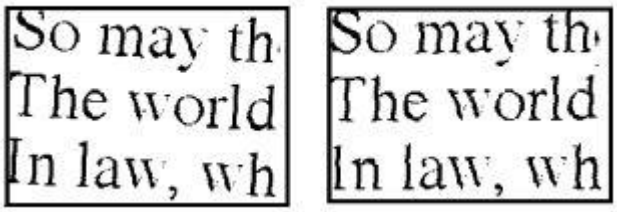

Figura 18 - Antes e depois da correção de enviesamento.

3.4.3. Inclinação

A inclinação de caractere é encontrado normalmente em escritas cursivas. A correção desta inclinação é um passo importante no estágio de pré-processamento para o reconhecimento tanto de palavras quanto de cadeias de números. O objetivo por trás desta correção é reduzir a variação de códigos e especificamente melhorar a qualidade dos candidatos à segmentação das palavras e numerais o que acaba seconvertendo numa maior taxa de precisão [CHERIET,2007].

O método matemático por trás desta técnica é a seguinte: para cada caractere, cria-se um retângulo que possa envolvê-lo.

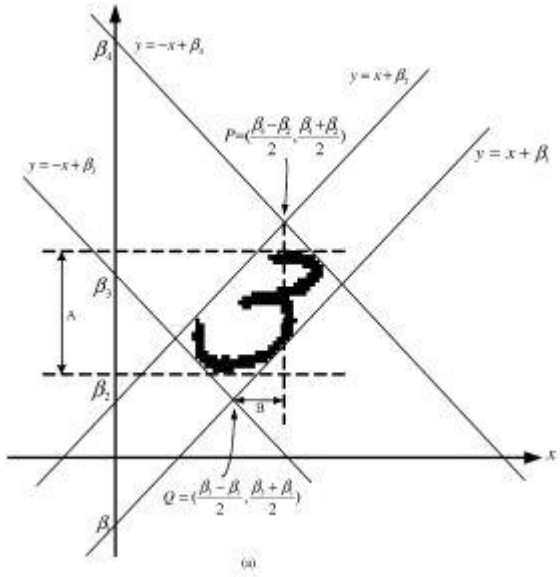

Figura 19 Um dígito circunscrito por um retângulo

Cada uma das retas pertencentes ao retângulo irá possuir uma equação própria:

$$y = x + \beta_1,$$
$$y = x + \beta_2,$$
$$y = -x + \beta_3,$$
$$y = -x + \beta_4,$$

Então o ângulo de inclinação é dado pela fórmula:

$\theta = \arctan(B/A)$.

Sendo que o valor A é a altura do caractere (ou dígito) e B obtido dado pela fórmula:

$B = (\beta 4 + \beta 1 - \beta 3 - \beta 2)/2$.

A correção da inclinação é feita após determinar o ângulo (θ). A fórmula de transformação de coordenadas é a seguinte:

$x' = x - y \cdot \tan(\theta)$,

$y' = y$.

Aonde o par ordenado (x',y') corresponde ao novo valor do pixel após transformação.

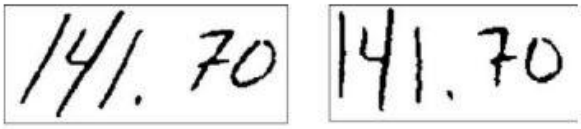

Figura 20 ~Dígitos antes e depois da correção de inclinação

3.4.4. Normalização

A normalização de caracteres é considerada a mais importante operação de pré-processamento para o OCR. Geralmente a imagem do caractere é mapeada em um plano padrão (de dimensões já conhecidas) a fim de se ter uma representação com dimensões fixas para classificação. O objetivo da normalização é reduzir as variações de formatos dos caracteres dentro de uma mesma classe a fim de facilitar o processo de extração de características e aumentar a precisão de todo o sistema. Basicamente, existem duas diferentes abordagens para o processo de normalização: as lineares e as não lineares.

As duas abordagens se valem do cálculo da proporção da imagem (ou "aspect ratio") que nada mais é que a relação entre o menor valor de largura ou comprimento e o maior valor de largura ou comprimento da imagem do caractere [CHERIET,2007].

$$R_1 = \frac{\min(W_1, H_1)}{\max(W_1, H_1)},$$

A partir deste resultado R_1, chega-se a um valor de proporção R_2 que será obtido dependendo do método de normalização utilizado.

Tabela 1 - Funções de cálculo da Proporção da Imagem

Method	Function
Fixed aspect ratio	$R_2 = 1$
Aspect ratio preserved	$R_2 = R_1$
Square root of aspect ratio	$R_2 = \sqrt{R_1}$
Cubic root of aspect ratio	$R_2 = \sqrt[3]{R_1}$
Sine of aspect ratio	$R_2 = \sqrt{\sin(\frac{\pi}{2} R_1)}$

Com o cálculo do valor de R_2, aplicando as fórmulas correspondentes para obtenção do novo posicionamento dos pixels chega-se à imagem normalizada.

Tabela 2- Cálculo de coordenadas a partir dos métodos de normalização.

Method	Forward mapping	Backward mapping
Linear	$x' = \alpha x$ $y' = \beta y$	$x = x'/\alpha$ $y = y'/\beta$
Moment	$x' = \alpha(x - x_c) + x'_c$ $y' = \beta(y - y_c) + y'_c$	$x = (x' - x'_c)/\alpha + x_c$ $y = (y' - y'_c)/\beta + y_c$
Nonlinear	$x' = W_2 h_x(x)$ $y' = H_2 h_y(y)$	$x = h_x^{-1}(x'/W_2)$ $y = h_y^{-1}(y'/H_2)$

Na tabela acima, os valores de α e β correspondem aos valores de proporção da imagem, antes e depois da normalização, respectivamente [CHERIET,2007].

3.4.5. Determinação do Contorno

Determinação do Contorno é também conhecido por acompanhamento de borda ou acompanhamento de limite, é uma técnica aplicada a um objeto para extrair somente o contorno externo de um objeto ou padrão, como um caractere. Para conseguir se definir os pixels aonde estão as bordas, é definido um conjunto de pixels e feito a comparação entre o pixel central e seus vizinhos. Existem dois tipos de padrões para determinação de borda: borda de 4 pixels e borda de 8 pixels [CHERIET,2007].

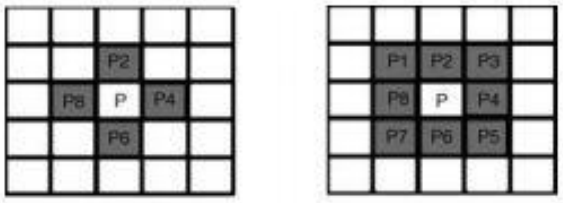

Figura 21- Borda de 4 pixels (esqueda) e Borda de 8 pixels (direita)

Com a determinação de onde estão os pixels de borda no objeto analisado, este pode ser submetido aos algoritmos de definição de contorno.

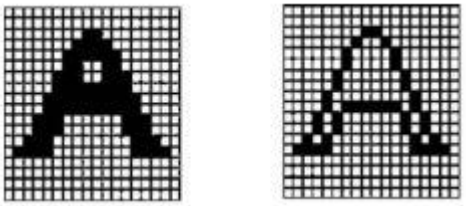

Figura 22 Imagem original (esquerda) Imagem após definição de contorno (direita)

A determinação do contorno é uma das técnicas de pré-processamento aplicadas na imagem a fim de se extrair informações importantes a respeito de seu formato. Uma vez o contorno é extraído, suas diferentes características serão analisadas e serão posteriormente utilizadas para classificação.

Por utilizar menos quantidade de informação para processamento a técnica de determinação de contorno auxilia de forma significativa e rápida no reconhecimento de caracteres [CHERIET,2007].

3.4.6. Afinamento

O afinamento é um processo de retirar de um padrão o máximo de pixels possível sem afetar o aspecto geral deste mesmo padrão. Ou seja, mesmo com a remoção de vários pixels, o padrão ainda é reconhecível. O afinamento também é conhecido por esquelitização. Este nome foi dado porque o processo algoritmo faz com que o padrão do resultado obtido tenha as seguintes propriedades:

- Ser o mais fino possível;
- Os pixels devem estar todos conectados;
- Os pixels devem estar todos centralizados.

Quando estas três condições são obtidas o algoritmo é interrompido.

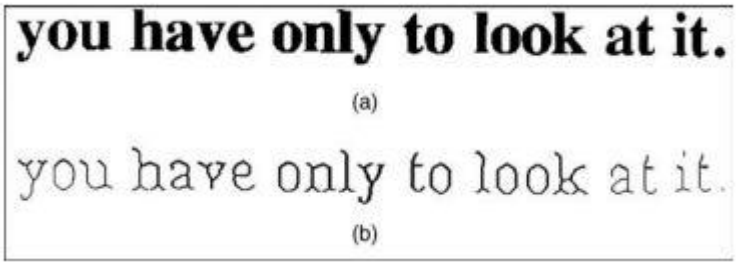

Figura 23 Processo de afinamento ou esquelitização

O esqueleto resultante mostra o formato geral do padrão e a partir disso algumas características importantes podem ser extraídas: intersecções, número de ramificações, posição relativa. Este processo também é útil quando se interessa a posição relativa da marcação do caractere e não o tamanho dele [CHERIET,2007].

4. Extração de Características

Segundo Eikvil, o objetivo da extração de características é capturar as propriedades essenciais dos símbolos. Isto é tido como um dos dificeis problemas para os sistemas de OCR. A forma mais diret de descrever um caractere é por sua imagem rasterizada. Uma outra forma é através da obtenção de características que ainda permitam a determinação dos símbolos mas deixando de fora atributos que não são importantes. As técnicas são frequentemente divididas em três grupos, aonde as características são obtidas:

- Distribuição dos pixels – Esta categoria cobre técnicas que extraiam características baseadas na distribuição estatística dos pontos. As características obtidas são frequentemente tolerantes a distorções e variações de estilo.
- Transformações e expansões em séries – Estas técnicas ajudam a reduzir a dimensionalidade do vetor característico e as características obtidas podem ser invariantes face a deformações globais, tais como translação e rotação. Nesta técnica as transformações utilizadas são as de Fourier, Walsh, Haar, Hadamard, Karhunen-Loeve, Hough e outras.
- Análise estrutural – Aqui as características obtidas são para descrever de forma geométrica e topológica os caracteres. Com estas informações pode-se descrever a parte física do caractere como ranhuras, voltas, pontos de término, intersecções de linhas. Comparada às outras técnicas, a análise estrutural fornece características com alta tolerância a ruídos e variações de estilo. Entretanto, não são tão robustas quando se trata de rotação e translação.

Os diferentes grupos de características podem ser avaliados de acordo com sua sensibilidade ao ruído e deformação, além da facilidade de implementação e uso [EIKVIL, 1993].

5. Classificação

De acordo com Cheriet, o objetivo final do reconhecimento de caracteres é obter as classes de códigos (etiquetas) dos padrões de caracteres. Ao segmentar padrões de caractere ou de palavras a partir de imagens de documentos, o trabalho de reconhecimento se torna ligar cada padrão de caractere ou palavra a um conjunto pré-definidos de classes. Enquanto muitos métodos de reconhecimento de palavras também possuam um esquema baseado em segmentação com modelagem de caractere ou reconhecimento de caractere acoplado, a performance do reconhecimento de caractere é de sumária importância para análise documental. Ao se mapear os padrões de entradas para um ponto no conjunto de características através da extração de características, o problema se torna um de classificação clássica.

A classificação de padrões tem sido o tema principal dentro da área de reconhecimento de padrões e frequentemente é vista como sendo o próprio "reconhecimento de padrões". Muita fundamentação teórica tem sido feita, principalmente com relação a reconhecimento estatístico de padrões (SPR) e muitos métodos eficientes tem se originado daí. Estas técnicas tem sido abordadas desde os anos 1970. A partir do final dos anos 1980, redes neurais artificiais (ANN) tem sido muito utilizadas para reconhecimento de padrões devido à redescoberta e bem-sucedidas aplicações de algoritmos com retro propagação para treinamento de redes de múltiplas camadas que são capazes de separar regiões de classes de distribuições arbitrariamente complicadas [CHERIET,2007].

5.1. Métodos Estatísticos

Segundo Cheriet, os métodos estatísticos são baseados na teoria de decisão de Bayes, que procura minimizar a perda de classificação quando dada uma matriz de perdas e probabilidades estimadas. De acordo com a abordadem de estimação de densidade probabilística, os métodos de classificação estatística são divididos em parametrizados e não parametrizados.

Para que se faça a tomada de decisão de Bayes, se requer o conhecimento prévio das probabilidades e das funções de densidade de probabilidades. As probabilidades podem ser estimadas como a porcentagem de amostras de uma classe em um conjunto de treinos e que sejam tomadas como igual para todas as classes. As funções de densidade de probabilidade podem ser estimadas por várias formas. Os métodos de classificações parametrizadas assumem formas funcionais para funções de densidade e estima os parâmetros pela maior

probabilidade, enquanto que os métodos não parametrizados podem estimar arbitrariamente distribuições adaptáveis as amostras de treinamento [CHERIET,2007].

5.2. Redes Neurais Artificiais

As redes neurais artificiais (RNA) foram inicialmente estudadas com a esperança de criar maquinas com percepção e cognição inteligente ao simular a estrutura física do cérebro humano. Os princípios e algoritmos das RNAs possuem infinidade de aplicações em diversos campos, incluindo reconhecimento de padrões e processamento de sinais. Uma rede neural é composta por um número de neurônios interconectados e a forma de interconexão diferencia os modelos de rede em redes alimentadas adiante e redes recorrentes [Gonçalves, 2010].

Uma rede com múltiplos neurônios interconectados de uma forma sofisticada pode calcular aproximações para funções descriminantes não lineares complexas. Redes de alimentadas adiante, incluindo redes de camadas simples, redes de múltiplas camadas, redes de função básica radial e redes de maiores ordens são diretas para cálculo de funções discriminantes através do aprendizado supervisionado.

6. Pós-Processamento

6.1. Agrupamento

O resultado do simples reconhecimento de símbolos em um documento é um conjunto de símbolos individuais. Entretanto estes símbolos, normalmente, não contém informação isoladamente. Ao contrário, deseja-se a associação de símbolos individuais com aqueles pertencentes à mesma cadeia, criando palavras e números. O processo de associar símbolos em cadeias é comumente chamado de agrupamento. O agrupamento de símbolos em cadeias é baseado na posição dos símbolos dentro do documento. Os símbolos que estão suficientemente próximos são colocados juntos.

Para fontes com tamanho padronizado, o trabalho é muito mais fácil uma vez que se sabe o espaço que cada caractere ocupa. Para caracteres com tipos determinados, a distância entre eles é variável, mas a distância entre uma palavra e outra é relativamente muito maior. Mesmo assim o agrupamento é possível. O problema ocorrre para caracteres escritos à mão ou quando o texto está inclinado.

6.2. Detecção de erros e correção

Mesmo os mais modernos sistemas de OCR não conseguem fornecer uma precisão absoluta na identificação de caracteres, mas alguns dos erros podem ser detectados ou até corrigidos pelo uso do contexto.

Existem dois mecanismos principais, um deles é feito pelo uso de regras definindo a sintaxe da palavra. Por exemplo, depois de um ponto final, deve se iniciar a frase com letra maiúscula. Também, para diferentes línguas a probabilidade de dois ou mais caracteres aparecem juntos pode ser calculada e usada para detectar erros. Por exemplo, em português não existe palavras que começam com "ç" ou palavras com três letras "s" juntas, então isso pode já ser classificado como erro.

Uma outra abordagem é o uso de dicionários, que já é provado ser a melhor forma de detectar e corrigir erros. Dado uma palavra, aonde um erro pode estar presente, ela é "olhada" no dicionário. Se a palavra não existe no dicionário, então o erro é detectado e pode ser corrigido alterando esta palavra para a mais similar. Se a palavra aparece no dicionário, isso não prova que não houve erro no processo. Um erro pode transformar uma palavra em outra que também exista no dicionário e, dessa forma, o erro não é detectado. A desvantagem dos métodos de dicionário é que a busca e comparação tomam maior tempo de processamento.

7. Os sistemas *open source.*

Será apresentado a seguir, os sistemas baseados em código aberto (open source). Para cada um deles será listado uma série de características e no próximo tópico serão classificados com relação as metodologias de OCR que foram descritas.

7.1. Tesseract OCR Engine

7.1.1. Histórico

Tesseract é um sistema OCR de código aberto que foi desenvolvido pela HP entre 1984 e 1994. Apareceu pela primeira vez em 1995 na *UNLV Annual Test of OCR Accuracy*.

Tesseract começou como um projeto de pesquisa de PhD no HP Labs e Bristol e ganhou importância como um possível software ou hardware adicional para a linha de escâneres de mesa da HP. A motivação era dada no fato que sistemas OCR ainda estavam engatinhando e sempre falhavam quando o documento de origem não possuía uma impressão excelente.

Após um projeto em conjunto entre a HP Labs Bristol e a divisão de escâneres da HP no Colorado, Tesseract tinha uma significante liderança na precisão com relação a outros produtos comerciais, mas não se tornou um produto. O próximo estágio de desenvolvimento ocorreu novamente no HP Labs Bristol como uma investigação do OCR para compressão. O trabalho concentrou-se mais em melhorar a eficiência de rejeição do que na precisão de entrada. Em 1994, o projeto foi cancelado. Em 1995 foi enviado a UNLV para o Teste Anual de Precisão de OCR aonde provou seu valor em face aos outros sistemas comerciais da época. Em 2005, HP liberou Tesseract como código-aberto [SMITH,2007].

7.1.2. Licenciamento

Tesseract OCR Engine está licenciado através da Apache Licence 2.0 (ANEXO-APACHE)[GOOGLE-TESSERACT,2012].

7.1.3. Plataformas

Tesseract OCR Engine está disponível para sistemas operacionais Windows (com o uso de VC++ Express ou CygWin), Linux e MacOS X[GOOGLE-TESSERACT,2012].

7.1.4. Programação

Tesserac OCR Engine foi codificado em C++[GOOGLE-TESSERACT,2012]..

7.1.5. Obtenção

O código-fonte do Tesseract OCR Engine está disponível para consulta no endereço http://tesseract-ocr.googlecode.com/svn/trunk/ (acessado em 8/12/12). Para download do programa, a URL é: http://code.google.com/p/tesseract-ocr/downloads/list (acessado em 8/12/12) [GOOGLE-TESSERACT,2012].

7.1.6. Instalação

Existem duas partes a serem instaladas: o próprio mecanismo Tesseract e os dados de treinamento para uma língua.

- Linux

Tesseract está disponível diretamente para várias distribuições de Linux. A instalação mais rápida é através dos chamados “packages” que se comunicam com os repositórios de distribuição. Após instalar o programa, é necessário fazer um download dos dados de treinamento, desempacotá-lo, e copiar o arquivo .traineddata para o diretório ‘tessdata’. O arquivo de treinamento está disponível em: http://code.google.com/p/tesseract-ocr/downloads/list.

- MacOS X

  A forma mais simples de instalar Tesseract é através do Homebrew (http://mxcl.github.com/homebrew/) Uma vez instalado o Homebrew, o Tesseract pode ser instalado simplesmente executando o comando “brew install tesseract”.

  Caso deseje um outro pacote de treinamento que não o que veio através do homebrew, faça o download em http://code.google.com/p/tesseract-ocr/downloads/list e depois copie o arquivo .traineddata no diretório `/usr/local/Cellar/tesseract/<version>/share/tessdata`

- Windows

  Existe um programa de instalação que está disponível na parte de downloads. O programa inclui o pacote de dados de treinamento na língua inglesa. Caso seja necessário obter um pacote para outra língua, acesse http://code.google.com/p/tesseract-ocr/downloads/list , abaixe o arquivo desejado, descomprima e copie o .traineddata para o diretório “tessdata”, geralmente é `C:\Program Files\Tesseract OCR\tessdata`.

7.1.7. Manuseio

Tesseract é um programa executado através de linha de comando, então tudo será feito através de um terminal ou janela de prompt de comando (ou Aviso do MS-DOS). O comando é basicamente este:

```
tesseract imagename outputbase [-l lang] [-psm pagesegmode] [configfile...]
```

Assim, para ler uma imagem chamada “myscan.png” e salvar o resultado em “out.txt”, o comando seria:

```
tesseract myscan.png out
```

Tesseract também possue um modo hOCR, que cria um código em HTML na coordenada de cada letra. Isso pode ser útil para criar um arquivo PDF, através do Hocr2PDF, cujo conteúdo seja buscável. Para usar, insira “hocr” na opção de configuração:

```
tesseract myscan.png out hocr
```

Outras opções estão disponíveis no ANEXO – TESSERACT.

7.2. OCROpus

7.2.1. Histórico

O OCROpus engine é baseado em dois projetos de pesquisa: um sistema de reconhecimento de escrita à mão de alto desempenho, desenvolvido na metade dos anos 1990 pelo escritório US Census e uma ferramenta de alto desempenho de métodos de análise. O projeto atualmente é patrocinado pela Google e encabeçado pelo Prof. Thomas na DFKI (Centro de Pesquisa Alemão para Inteligência Artificial, Kaisersleutern, Alemanha)[OCROPUS; 2012].

7.2.2. Licenciamento

O OCROpus engine possui seu licenciamento baseado na Apache 2.0 [OCROPUS; 2012].

7.2.3. Plataformas

O OCROpus engine está disponível somente para sistemas Linux [OCROPUS; 2012].

7.2.4. Programação

O OCROpus engine foi estrito em Python, NumPy e SciPy [OCROPUS; 2012].

7.2.5. Obtenção

Como é um programa voltado para Linux, sua obtenção é através de linha de comando:

```
hg clone -r ocropus-0.6 https://code.google.com/p/ocropus
```

O código-fonte pode ser obtido neste endereço:
<http://googlecode.blogspot.com.br/2009/04/mercurial-support-for-project-hosting.html>
Acessado em 08/12/12.

7.2.6. Instalação

A instalação, como dito antes, é feito através de linha de comando em Linux:

```
  $ hg clone -r ocropus-0.6 https://code.google.com/p/ocropus
    $ cd ocropus/ocropy
    $ sudo apt-get install $(cat PACKAGES)
    $ python setup.py download_models
    $ sudo python setup.py install
    $ ./run-test
```

7.2.7. Manuseio

Uma vez instalado, a operação é feita através de linha de comando. Para ler várias imagens escaneadas, o comando é:

```
$ ocropus-recognize-book *.png -o output.html
```

Para poder fazer somente o reconhecimento de linhas de texto, o commando é:

```
$ ocropus-lattices line1.png line2.png ...
$ ocropus-ngraphs line1.lattice line2.lattice ...
$ cat line1.txt line2.txt ...
```

"Para reconhecimento de caracteres individuais, deve ser carregado um arquivo que contenha o modelo chamado ".cmodel" através de uma função *cPicle.load*

```
cmodel = cPickle.load(open("models/en-uw3unlv-perchar.cmodel"))
assert cmodel.sizemode=="perchar"
normalized = ocrolib.classifier_normalize(character_image)
cmodel.coutputs(normalized)
```

7.3. GOCR

7.3.1. Histórico

GOCR apareceu no final dos anos 1990 e foi criado por Joerg Schulenburg. O nome GOCR significa GNU Optical Character Recognition [GOCR; 2012].

7.3.2. Licenciamento

GOCR é licenciado através de GPL. (ANEXO- GPL) [GOCR; 2012]

7.3.3. Plataformas

GOCR está disponível para Windows, OS/2 e Linux [GOCR; 2012].

7.3.4. Programação

GOCR foi desenvolvido em C++ [GOCR; 2012].

7.3.5. Obtenção

O programa pode ser abaixado no endereço http://sourceforge.net/projects/jocr/ [GOCR; 2012].

7.3.6. Instalação

A instalação é feita da seguinte maneira:

- Windows

  Acesse o endereço http://www-e.uni-magdeburg.de/jschulen/ocr/download.html e abaixe o arquivo binário para Windows.

- Linux

  Pode ser compilado utilizando o código-fonte, ou abaixando o arquivo correspondente à distribuição em http://www-e.uni-magdeburg.de/jschulen/ocr/download.html

- OS/2
- Acesse o endereço http://www-e.uni-magdeburg.de/jschulen/ocr/download.html e abaixe o arquivo binário para OS/2.

7.3.7. Manuseio

Após instalação, o manuseio é direto na linha de comando. O mais utilizado é gocr like -c

8. Análise comparativa qualitativa dos sistemas de código-aberto quanto aos métodos de OCR

Com base nos elementos teóricos descritos nos itens anteriores, e também em foi feita uma análise comparativa dos três sistemas de código-aberto os resultados tabulados são:

Tabela 3 Comparativo entre os Sistemas

Métodos	SISTEMA		
	Tesseract	OCROpus	GOCR
Localização e Segmentação	A imagem é analisada e contornos são definidos. Os contornos são agrupados em regiões "blob" que serão tratadas como palavras e linhas de texto.	Algoritmo de segmentação Run-Length Smearing	??
Pré-Processamento	Utiização de thresholding adaptativo; algoritmos de detecção de textos envesados;	Binarização; Detecção orientação do texto; Remoção de ruído; Correção de envezamento;	??
Extração de Características	Obtenção do posicionamento (x,y) e ângulo - para carateres e valores de posicionamento ângulo e comprimento para análise de protótipos	Algoritmos de extração: IFeatureMap (c++) IExtractor (Python)	??
Classificação	Utilização de classificador estático e adaptativo; análise linguistica para classificar em palavras;	Algoritmos implementados: Tesseract; MLP-Based; HMM-Based(proximas versões).	??
Pós-Processamento	Não implementado.	Uso de ferramentas de linguagem estatíticas com base na biblioteca OpenFST	??

De acordo com o que foi analisado, do ponto de vista qualitativo, o sistema OCROpus é completo em todas as metodologias de um sistema OCR, uma vez que a parte de pós-processamento é inexistente no Tesseract e um mesmo algoritmo para classificação (Tesseract) é utilizado pelo Tesseract e pelo OCROpus. Também pudemos notar a falta de documentação para o sistema GOCR. Durante o processo de elaboração deste trabalho, enviamos duas mensagens ao Senhor Joerg Schulenburg, que é responsável pelo projeto, pedindo maiores informações e documentação. Nenhuma das mensagens foi respondida.

9. Considerações Finais

Os sistemas de OCR já possuem um desenvolvimento de mais de 50 anos, muitas teorias matemáticas são aplicadas para resolver o principal problema enfrentado agora: o reconhecimento eficaz da escrita de mão. Já se faz uso do desenvolvimento colaborativo, como nos casos dos sistemas de código-aberto. Nossa análise, no escopo deste trabalho, ficou limitado à parte qualitativa. Isso abre um caminho de se partir para uma análise mais aprofundada e quantitativa, ao menos dos dois sistemas que são melhor documentados (Tesseract e OCROpus).

10. Referências Bibliográficas

AIM, Inc. (2000), **Optical Character Recognition**, Boletim Técnico da The Association for Automatic Identification and Data Capture Technologies, p.4, Set. 2000.

Eikvil, Line (1993). **OCR Optical Character Recognition**

Mori, Shunji et al. (1992). **Historical Review of OCR Research and Development**

Roberti, Bruno (2010) **Radares inteligentes passam a multar carros sem licenciamento** Disponível em: < http://quatrorodas.abril.com.br/noticias/radares-inteligentes-passam-multar-carros-licenciamento-275103_p.shtml> Acesso em 30. nov. 2012.

Rumelt, Richard P.(2002) **Gordon Moore's Law**. Disponível em: <http://www.anderson.ucla.edu/faculty/dick.rumelt/Docs/Cases/MooresLaw.pdf> Acesso em 30 nov. 2012.

Cheriet, Mohamed et al.;(2007) **Optical character recognition devices**.

Meshesha, Million (2008). **Recognition and Retrieval from Document Image** Collections. Disponível em: < http://cvit.iiit.ac.in/thesis/millionPHD2008/millionThesis2008.pdf> Acesso em 30. Nov. 2012

Tauschek, Gustav (1935) **Reading Machine, US Patent Office 2.026.329.** Disponível em : < http://history-computer.com/Library/US2026329.pdf> Acesso em 30. nov. 2012.

Handel, Paul (1933) **Statistical Machine. U.S. Patent Office 1.915.993.** Disponível em: <http://www.google.com/patents/US1915993>. Acesso em 30. nov 2012.

Cooper, Franklin S. (1984). **Evolution of Reading Machines for The Blind: Haskins Laboratories' Research as a Case History.**

Auerbach, Isaac L. (1961) **European Eletronica Data Processing – A Report on the Industry and The State-Of-The-Art**. Disponível em: <http://pdp8.co.uk/wp-content/library/bitsavers/pdf/auerbach/European_EDP_Jan61.pdf> Acesso em 30. nov. 2012.

Muda, Nadira et al.;2007, **Optical Character Recognition By Using Template Matching**; disponível em <http://www.academia.edu/714194/Optical_Character_Recognition_By_Using_Template_Matching_Alphabet_>; Acessado em 02/12/2012.

NEW YORK TIMES; 1988; **BUSINESS TECHNOLOGY; Now, PC's That Read A Page and Store It**; Disponível em < http://www.nytimes.com/1988/08/17/business/business-technology-now-pc-s-that-read-a-page-and-store-it.html>; Acessado em 03/12/2012

Fujisawa ,Hiromichi; 2008; **Forty years of research in character anddocument recognition---an industrial perspective**; Disponível em; <http://www.sciencedirect.com/science/article/pii/S0031320308000964> .Acessado em 03/12/2012

CHEN, D.; LUETTIN, J.; SHEARER, K.; 2000; A Survey of Texto Detection and Recognition in Images and Videos; Disponível em: < http://www.cs.cmu.edu/~datong/survey.pdf>; Acessado em: 03/12/2012.

Gonçalves, A.; 2010; **Redes Neurais Artificiais**; Disponível em <http://www.dca.fee.unicamp.br/~andreric/arquivos/pdfs/redes_neurais.pdf>; Acessado em 05/12/2012.

SMITH, Ray; 2007; **An Overview of the Tesseract OCR Engine**; Disponível em: <http://code.google.com/p/tesseract-ocr/> Acessado em 06/12/12.

GOOGLE-TESSERACT;2012; Site; Disponível em <http://code.google.com/p/tesseract-ocr/> Acessado em 06/12/12.
OCROPUS; 2012; Site ;Disponível em <http://www.ocropus.org> Acessado em 06/12/12.

GOCR; 2012; Site; Disponivel em <http://jocr.sourceforge.net/>
BREUEL;Thomas M.;2008; **The OCRopus Open Source OCR System**; Artigo Técnico.
LANGUAGES TECHNOLOGIES UNIT, Bangor University;2008; **An overview of the Tesseract OCR (optical character recognition) engine, and its possible enhancement for use in Wales in a pre-competitive research stage**; Disponível em: <http://www.saltcymru.org/english/saltcymru_document5.pdf> ; Acessado em: 08/12/12.

SHAFAIT, Faisal; **Document Image Analysis with OCROpus**. Disponível em: <http://www.dfki.uni-kl.de/~shafait/papers/Shafait-OCRopus-Tutorial-INMIC09.pdf> Acessado em 08/12/12.