http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2021.v13.n2.e356
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 13/03/2021 Revisado: 10/05/2021 Aceito: 20/07/2021

BUSCA DE PALAVRAS CHAVE EM IMAGENS DE LIVROS IMPRESSOS USANDO VISÃO COMPUTACIONAL

Keyword search in physical books with computer vision

Felipe Peruchi Simões[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Danillo Roberto Pereira[1], Mário Augusto Pazoti[1], Almir Olivette Artero[2], Marco Antônio Piteri[2]

[1]Faculdade de Informática de Presidente Prudente, Unoeste - Universidade do Oeste Paulista, Presidente Prudente
felipeperuchisimoes@gmail.com, chico@unoeste.br, llalmeida@unoeste.br, danilopereira@unoeste.br, mario@unoeste.br
[2]Faculdade de Ciências e Tecnologia, UNESP - Universidade Estadual Paulista
Departamento de Matemática e Computação, Presidente Prudente
almir.artero@unesp.br, marco.piteri@unesp.br

RESUMO – Com o uso cada vez mais frequente de livros no formato digital, as pessoas buscam os assuntos desejados de uma maneira mais rápida, se comparado à busca em livros impressos. Este trabalho almejou desenvolver um recurso computacional no formato de um aplicativo para smartphones Android, que a partir de uma imagem capturada de uma página de um livro, realiza buscas por palavras chave. O intuito de utilizar o aplicativo é de auxiliar o leitor a encontrar a informação desejada com mais agilidade. Foram utilizadas técnicas de Visão Computacional com o auxílio da biblioteca OpenCV no desenvolvimento de algoritmos para realizar a segmentação, correção da perspectiva da imagem da página do livro, identificação e retificação das linhas onduladas, reconhecimento e classificação de caracteres. Os resultados se mostraram promissores com uma taxa de acerto de mais de 88%.
Palavras-chave: Visão Computacional, OCR, Retificação, Android, Tesseract.

ABSTRACT – With the increasingly frequent use of books in digital format, people search for the desired subjects in a faster way compared to the search in physical books. This work aimed to develop a computational resource in the form of an application for Android smartphones, which, based on an image captured from a page in a book, performs searches by keywords. The purpose of using the application is to help the reader to find the desired information quickly. We use Computer Vision techniques with the aid of the OpenCV library in the development of algorithms to perform segmentation, correction of the perspective of the book page image, identification and rectification of the wavy lines, recognition and character classification. The results shown were promising with a hit rate of over 88%.
Keywords: Computer Vision, OCR, Rectification, Android, Tesseract.

1. INTRODUÇÃO

O uso de livros em formato digital vem aumentando cada vez mais, com isso, as pessoas buscam os assuntos desejados de uma maneira mais rápida, quando comparado à busca em livros impressos (KATZ, 2011). Acredita-se que a existência de um recurso computacional para facilitar e agilizar a busca de palavras em livros impressos seria de grande valia.

Existem vários problemas desafiadores a serem resolvidos na busca por palavras em imagens digitais, pois a captura da imagem de uma página de um livro pode apresentar variação de iluminação, diferentes ângulos da câmera, linhas de texto com aspecto curvilíneo, caracteres com fontes diferentes, ruídos, dentre outros (KUHN; CERVI; MANICA, 2018). A correção das imagens é necessária, e para isso pode ser feita a binarização da imagem, remoção de ruídos, segmentação e correção de perspectiva (LIANG; DOERMANN; LI, 2005). Agrawal e Kaur (2018) buscaram a extração de textos de imagens usando o algoritmo de Otsu (OTSU, 1979) para segmentação e o método de transformada de Hough para detecção de distorção. Oliveira *et al.* (2018) buscaram aplicar diversas técnicas para conseguir identificar textos em placas informativas de trânsito, como usar intervalos de cores para encontrar a placa na imagem, identificação de contornos e segmentação para eliminar partes não importantes. Também existe a atividade de reconhecimento dos caracteres para se ter a busca das palavras desejadas. Existem muitos trabalhos na literatura para o reconhecimento óptico de caracteres, como o trabalho de Silva *et al*. (2012) que propõe uma estratégia simples e rápida para modelar o comportamento dos caracteres usando apenas as transições que ocorrem entre os níveis de pixels adjacentes. Miranda *et al*. (2013) sugere extrair atributos das imagens e realiza a classificação usando uma abordagem baseada em valores máximos e mínimos de cada atributo. Khaoula, *et al*. (2013) propõe dois OCRs aplicados a textos presentes em cenas de vídeos e em imagens de cenas naturais, sendo a primeira abordagem destinada a segmentar imagem de texto em caracteres individuais antes de reconhecê-los, enquanto a segunda realiza esquema de varredura em várias escalas para localizar e reconhecer os caracteres, além de utilizar conhecimento linguístico para remover erros devido a confusões de reconhecimento. O Tesseract (SMITH, 2007) busca por características em cada caractere e compara com padrões de um determinado alfabeto.

Embora os algoritmos citados sejam de grande eficiência, os problemas demonstrados por Kuhn, Cervi e Manica (2018), Liang, Doermann e Li (2005), Miranda *et al*. (2013), e por Oliveira *et al.* (2018) podem atrapalhar muito os resultados, dificultando a identificação dos caracteres, e diminuindo a eficiência do algoritmo.

Este trabalho busca contribuir com uma solução computacional para auxiliar o processo de busca de palavras chave em livros impressos. Foram aplicadas técnicas de Visão Computacional no desenvolvimento de um algoritmo para realizam essa tarefa. A busca é realizada a partir de uma imagem capturada da página de um livro, utilizando um aplicativo para smartphones Android, que foi desenvolvido neste trabalho. Após a captura é realizado o processo de segmentação, correção de perspectiva, retificação das linhas e, por fim, o reconhecimento dos caracteres. Com os caracteres reconhecidos na imagem da página do livro, é feita a busca das palavras chave desejadas.

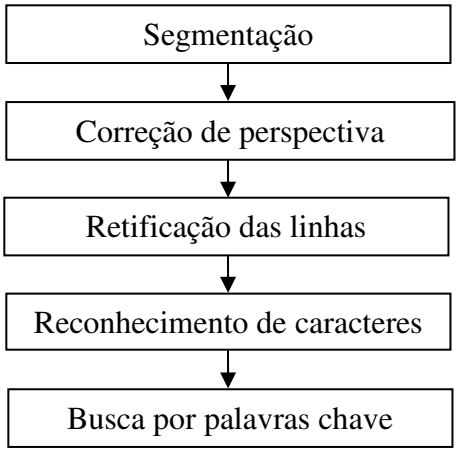
Após esta seção de introdução, o trabalho está organizado da seguinte maneira. Na Seção 2 é descrito o método proposto para realizar a busca por palavras chave em imagens de livros. Na Seção 3 são apresentados os experimentos realizados e resultados obtidos a partir da metodologia desenvolvida. Por fim, na Seção 4 encontram-se as conclusões e propostas de trabalhos futuros.

2. MÉTODO PROPOSTO

Para a construção do aplicativo foi usado o ambiente de desenvolvimento Android Studio com o auxílio da biblioteca de Visão Computacional OpenCV. A metodologia proposta aplicou algoritmos para pré-processamento da imagem capturada no smartphone, como segmentação para identificação do livro e da página. Posteriormente, foi realizada a correção de perspectiva da imagem da página e a retificação para eliminar ondulações das linhas do texto. Também foram aplicados filtros para eliminação de ruídos, o que favorece os passos finais, que é a detecção e identificação de caracteres das palavras chave.

O fluxograma da Figura 1 apresenta a sequência das etapas do método proposto.

Figura 1. Fluxograma de processos da metodologia desenvolvida neste trabalho.



Fonte: (Autor, 2020).

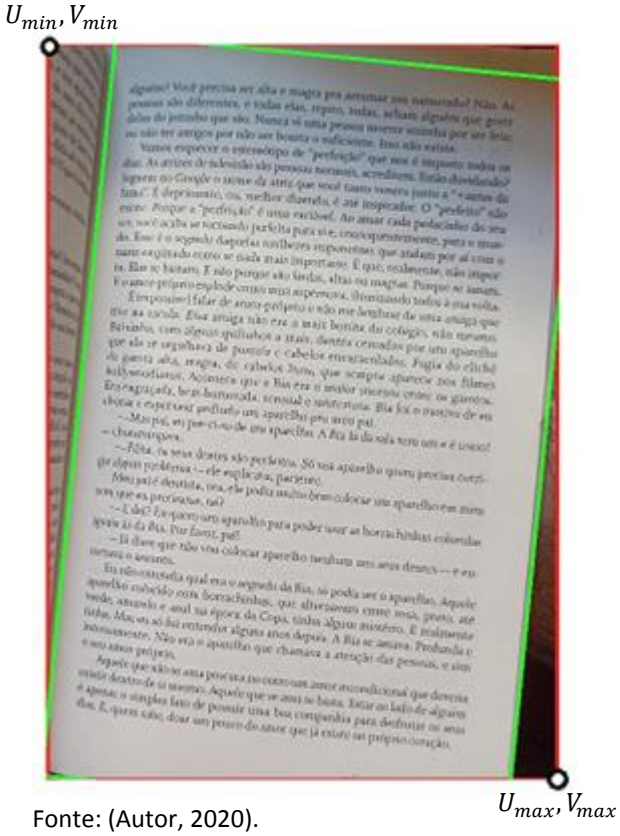
2.1. Segmentação

Nas imagens capturadas pelo smartphone podem conter muitas informações desnecessárias. O que importa para o algoritmo é apenas a página do livro, sendo que o fundo e qualquer outro elemento contido na imagem além da página, podem ser descartados.

A segmentação da página foi realizada utilizando a metodologia proposto por Sidhwa *et al.* (2018) em que primeiramente é feita a binarização da imagem, só então os contornos são encontrados e filtrados, e então é selecionado apenas o de maior dimensão.

Foi utilizado um algoritmo de limiarização (*threshold*) de Otsu (OTSU, 1979) com o intuito de obter o limiar mais adequado para separar a página do fundo. Após esse processo, foi aplicado o algoritmo para detecção de contornos *Border Fallowing* (SUZUKI; ABE, 1985) para obter as regiões de todos os objetos da imagem. Em seguida, essas regiões de interesse detectadas são ordenadas pelo tamanho e a maior região é escolhida como a página do livro (SIDHWA *et al*., 2018). Na Figura 2 é apresentado um exemplo com a região da página de um livro.

Figura 2. Exemplo da região da página de um livro selecionada.



Fonte: (Autor, 2020).

2.2. Correção da perspectiva

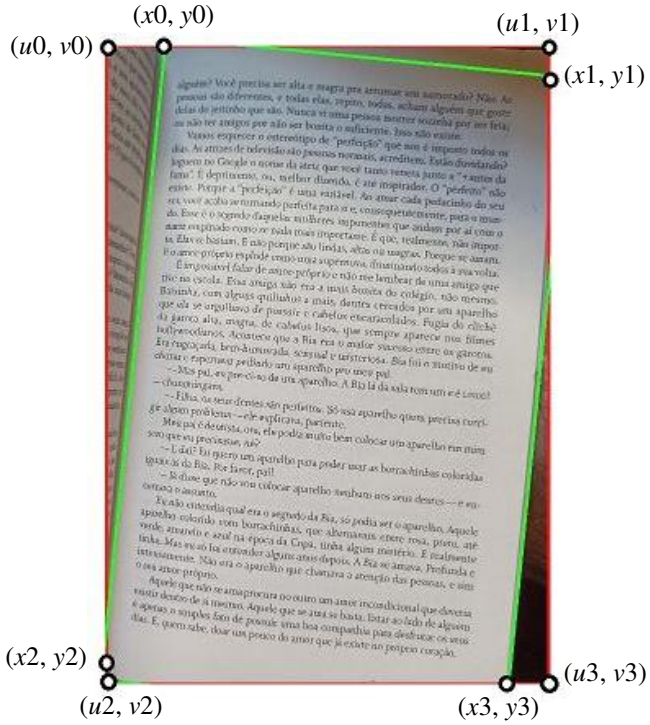
As imagens capturadas podem apresentar deformações devido ao ângulo da câmera e dificultar a identificação dos caracteres. Para evitar essas deformações é necessário corrigir a perspectiva. A correção de perspectiva foi realizada por meio de uma matriz de transformação (REIS *et al*., 2008). Foi aplicado um algoritmo de detecção de cantos *Harris Corner Detector* (HARRIS; STEPHENS, 1988), obtendo os valores das coordenadas *x* e *y* dos quatro cantos da página, que são utilizados na matriz de transformação. A matriz possui os valores dos quatro pontos de coordenada (*x*0, *y*0), (*x*1, *y*1), (*x*2, *y*2), (*x*3, *y*3) e também os quatro pontos da região de interesse (*u*0, *v*0), (*u*1, *v*1), (*u*2, v2), (*u*3, *v*3).

As coordenadas (*u*, *v*) são obtidas a partir dos valores $(U_{min}, V_{min}, U_{max}, V_{max})$ da região de interesse que contém a página segmentada e as coordenadas (*x*, *y*) foram determinadas a partir dessa região nas linhas que compõem os contornos da página na imagem.

Na Figura 3 são mostradas linhas na cor vermelha com as coordenadas (*u*, *v*) da região de interesse da página e linhas na cor verde com as

coordenadas (*x*, *y*) que representa o contorno da página.

Figura 3. Coordenadas (*u*, *v*) da região de interesse e (*x*, *y*) da página do livro**.**



Fonte: (Autor, 2020).

Para obter a transformação das coordenadas (*x*, *y*) da região contendo a imagem da página segmentada (origem), representadas como *A* na equação (1), para (*u*, *v*) da região de destino, representadas como *B* é utilizada a equação (1).

$$A.X = B \qquad (1)$$

que representa a matriz de transformação da equação (2).

A **X**

B

$$\begin{bmatrix} x0 & y0 & 1 & 0 & 0 & 0 & -x0u0 & -y0u0 \\ x1 & y1 & 1 & 0 & 0 & 0 & -x1u1 & -y1u1 \\ x2 & y2 & 1 & 0 & 0 & 0 & -x2u2 & -y2u2 \\ x3 & y3 & 1 & 0 & 0 & 0 & -x3u3 & -y3u3 \\ 0 & 0 & 0 & x0 & y0 & 1 & -x0v0 & -y0v0 \\ 0 & 0 & 0 & x1 & y1 & 1 & -x1v1 & -y1v1 \\ 0 & 0 & 0 & x2 & y2 & 1 & -x2v2 & -y2v2 \\ 0 & 0 & 0 & x3 & y3 & 1 & -x3v3 & -y3v3 \end{bmatrix} * \begin{bmatrix} a0 \\ a1 \\ a2 \\ b0 \\ b1 \\ b2 \\ c0 \\ c1 \end{bmatrix} = \begin{bmatrix} u0 \\ u1 \\ u2 \\ u3 \\ v0 \\ v1 \\ v2 \\ v3 \end{bmatrix} \qquad (2)$$

onde *X* representa um coeficiente desconhecido que é necessário para completar a matriz.

É formado um sistema linear para encontrar os valores de *X*, onde utiliza-se a decomposição de *LU* (*Lower* e *Upper*) (BUNCH; HOPCROFT, 1974), que separa a matriz em duas, como pode ser observado na equação (3), onde *L* é a matriz triangular inferior e *U* a matriz triangular superior.

$$A = L.U \qquad (3)$$

A fim de obter as matrizes *L* e *U* é utilizado o método de eliminação de Gauss (KAW; KALU; NGUYEN, 2018) que zera todos os elementos abaixo da diagonal principal da matriz, obtendo-se a matriz *U*. A matriz *L* é formada com os multiplicadores que foram usados para zerar os elementos da matriz *U* e com os valores acima da diagonal principal.

Após obter as matrizes *L* e *U* substitui-se a matriz *A* por *L* e *U* formando a equação (4). Substituindo (*U.X*) por *Y* (5) obtém-se a equação (6).

$$L.(U.X) = B \qquad (4)$$

$$U.X = Y \qquad (5)$$

$$L.Y = B \qquad (6)$$

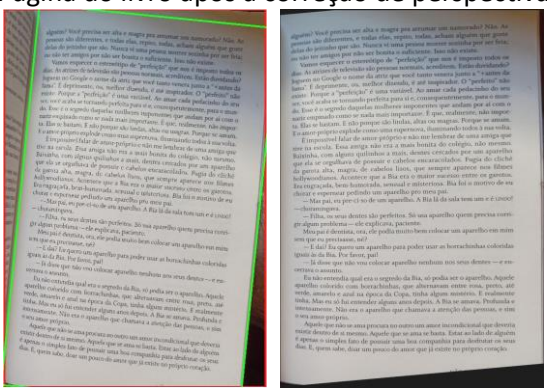
Resolvendo a equação (5) e (6) é possível encontrar os valores dos coeficientes desconhecidos e com eles utilizar as equações (7) e (8) com o objetivo de encontrar as novas coordenadas.

$$ui = \frac{(a0*xi+a1*yi+a2)}{c0*xi+c1*yi+1} \quad (7)$$

$$vi = \frac{(b0*xi+b1*yi+b2)}{c0*xi+c1*yi+1} \quad (8)$$

Para cada coordenada da região de origem é aplicada a equação (7) e (8) para se obter as novas coordenadas da região de destino. Ao final desse processo, a perspectiva da página é corrigida como mostrado na Figura 4.

Figura 4. a) Região da página segmentada. b) Página do livro após a correção de perspectiva.



a) b)

Fonte: (Autor, 2020).

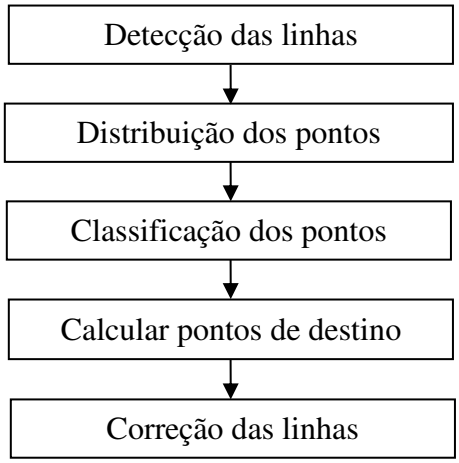
2.3. Retificação das linhas

As páginas dos livros podem ficar curvas, principalmente devido à encadernação contida nos livros impressos. Essas curvas atrapalham na identificação dos caracteres, por isso é de suma importância corrigi-las.

Para resolver esse problema, foi desenvolvido um algoritmo, em que foi necessário inicialmente, detectar as linhas das páginas, onde serão colocados os pontos de origem para fazer a distorção da imagem. Para cada ponto é preciso determinar a qual linha ele pertence, só assim é possível calcular o ponto de destino, e por fim, fazer a correção das linhas.

O fluxograma da Figura 5 apresenta a sequência das etapas do método proposto para realizar a retificação das linhas de texto na imagem da página de um livro.

Figura 5. Fluxograma dos processos para retificação das linhas desenvolvido nesse trabalho.



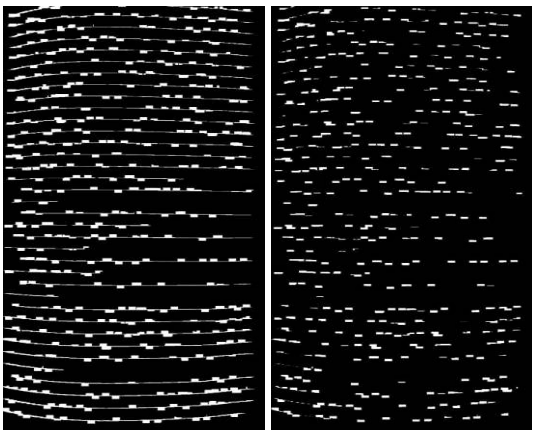
Fonte: (Autor, 2020).

Para a detecção das linhas, foi utilizada a operação de morfologia matemática dilatação, com um elemento estruturante 5x1 com o objetivo de esticar as linhas horizontalmente. Posteriormente, foi aplicada uma erosão, utilizando o elemento estruturante 1x3 para achatar as linhas verticalmente. Em ambos os casos a origem estava no centro. O resultado desse primeiro passo pode ser visto na Figura 6(a).

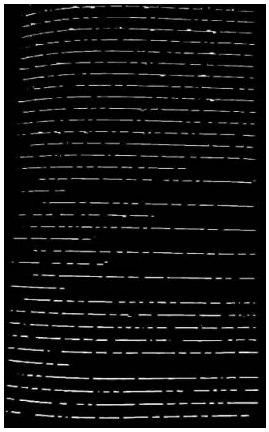
Como pode-se observar na Figura 6(a) surgem retângulos brancos por toda a imagem. Isso se deve ao fato de existirem letras que são maiores que as demais, tanto para baixo (‘g’, ‘q’, ‘ç’) como para cima (‘t’, ‘l’, ‘d’). Esses retângulos precisam ser eliminados para se obter apenas as linhas, e isso é feito com o processo a seguir.

Foi utilizada novamente uma erosão com o elemento estruturante 1x3, com isso as linhas desaparecem e restam apenas retângulos, porém, menores que os da Figura 6(a). Para retornar esses retângulos ao tamanho original é realizada uma dilação com o mesmo elemento estruturante 1x3 utilizado na erosão. O resultado, contendo apenas os retângulos, pode ser visto na Figura 6(b).

Figura 6. Etapas para encontrar as linhas. a) Linhas após sofrer o processo de dilatação e erosão. b) Retângulos encontrados. c) Linhas obtidas.



a) b)



c)

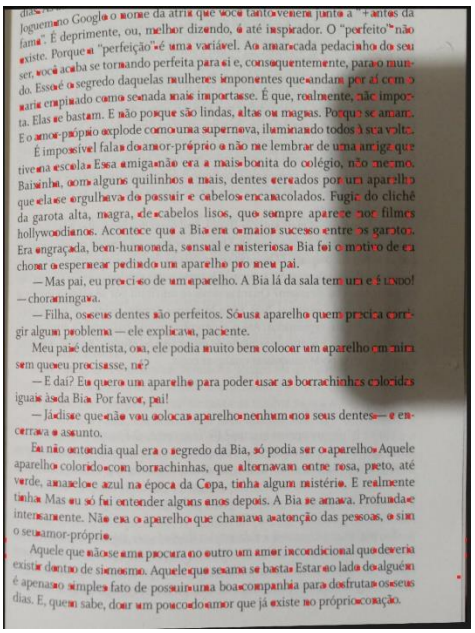
Fonte: (Autor, 2020).

Nota-se que na Figura 6(b) tem-se os retângulos da Figura 6(a). No próximo passo é utilizado o operador lógico XOR com a Figura 6(a) e 6(b) obtendo apenas as linhas. O resultado é apresentado na Figura 6(c).

Com as linhas detectadas na imagem da página do livro, pode-se então distribuir pontos sobre elas, que são os pontos de origem, usados para fazer a retificação das linhas.

Foi determinado de forma empírica uma distância de 50 pixels entre os pontos que são distribuídos sobre as linhas obtidas da Figura 6(c). Esses pontos podem ser vistos nas Figura 7.

Figura 7. Pontos colocados sobre as linhas da página do livro.

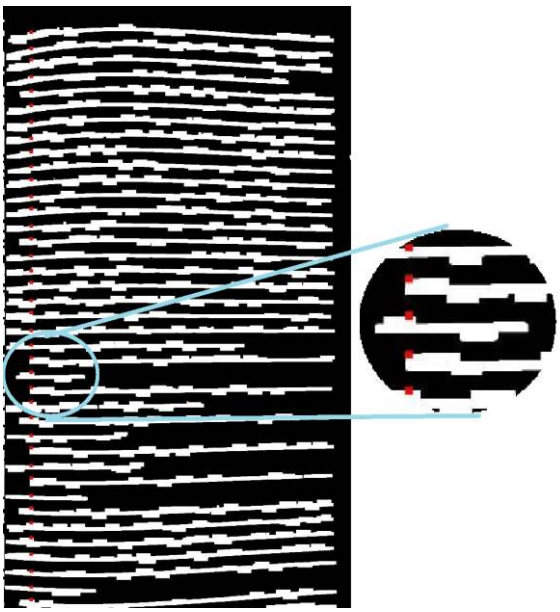


Fonte: (Autor, 2020).

Após colocar os pontos é preciso determinar quais desses pontos pertencem a uma mesma linha. Para isso foi preciso percorrer as linhas verificando quais pontos estavam contidos nelas.

O processo de percorrer as linhas se inicia encontrando um ponto inicial em cada linha. Para isso é preciso verificar em qual coluna (mesma coordenada *x*) é possível encontrar mais linhas. Em cada coordenada onde há intersecção dessa coluna encontrada com as linhas, é definido como um ponto inicial. Esse ponto inicial é o ponto de partida do algoritmo para o processo de percorrer as linhas. O resultado pode ser visto na Figura 8.

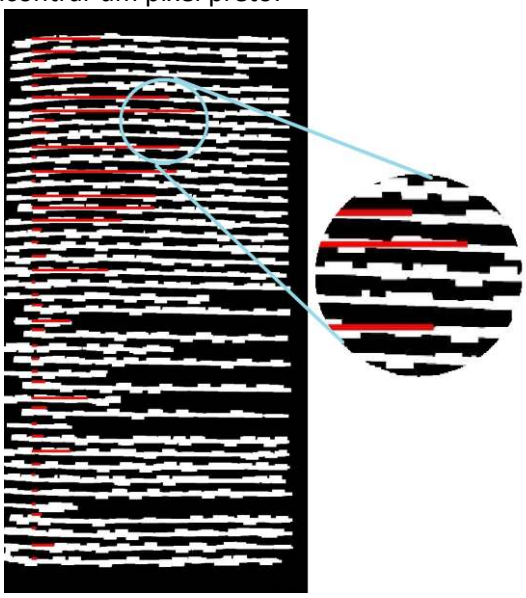
Figura 8. Pontos iniciais no processo de percorrer a linha.



Fonte: (Autor, 2020).

Após encontrar os pontos iniciais em cada linha, percorre-se a imagem à direita até encontrar um pixel preto, como mostra a Figura 9.

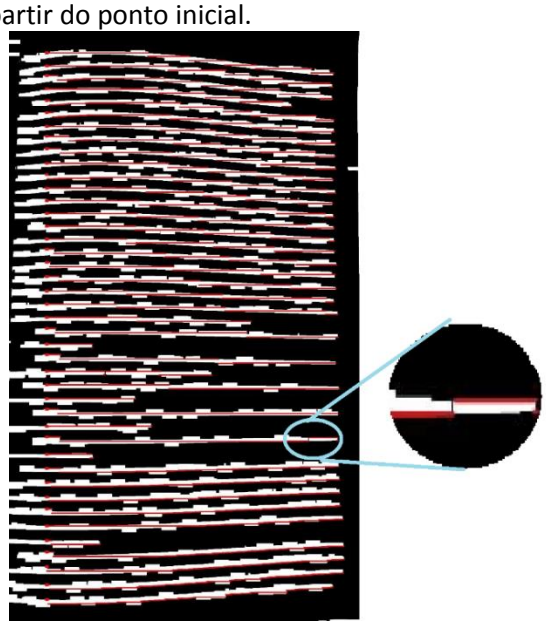
Figura 9. Percorrendo a imagem à direita até encontrar um pixel preto.



Fonte: (Autor, 2020).

Ao encontrar um pixel preto, o algoritmo procura um pixel branco acima ou abaixo, até 10 pixels em cada direção. Caso um pixel branco seja encontrado, deve-se retomar a varredura na imagem para a direita. Pode acontecer de a linha não estar totalmente conectada, caso isso ocorra, não será encontrado um pixel branco acima e nem abaixo para seguir adiante. Nesse caso, o algoritmo deve repetir o processo de procurar um pixel branco acima ou abaixo, porém, na coluna à direita. Esse processo deve ser repetido até que seja encontrado um pixel branco ou ter terminado de verificar todas as possíveis colunas e não encontrar nenhum pixel branco, pois a linha chegou ao fim.

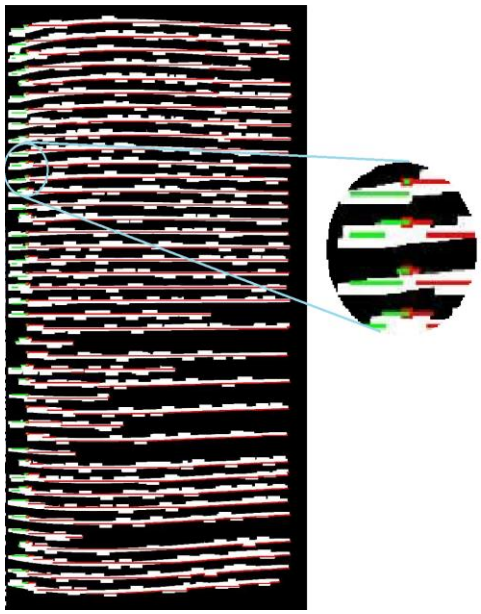
Figura 10. Exemplo de linha percorrida à direita a partir do ponto inicial.



Fonte: (Autor, 2020).

Todo esse processo de percorrer a linha é repetido para o lado esquerdo do ponto inicial. O resultado pode ser visualizado na Figura 11.

Figura 11. Processo de percorrer a linha, agora para o lado esquerdo.

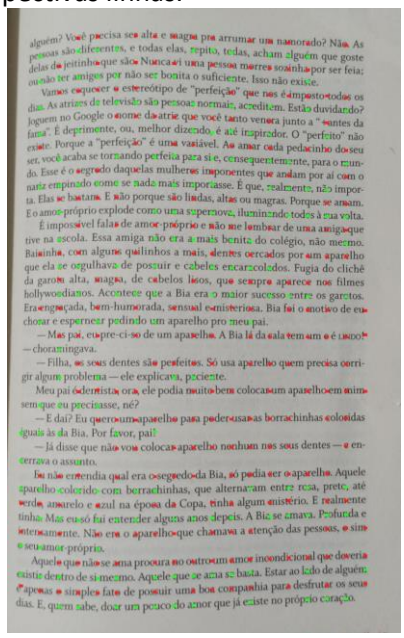


Fonte: (Autor, 2020).

Com o processo terminado, é possível determinar quais pontos pertencem a mesma

linha, como é apresentado na Figura 12. Pontos de cada linha foram pintados da mesma cor, sendo linhas com pontos na cor vermelha e linhas com pontos na cor verde. O resultado mostra que a classificação dos pontos em suas respectivas linhas do texto foi realizada de forma correta.

Figura 12. Pontos corretamente classificados nas suas respectivas linhas.


alguém? Você precisa ser alta e magra pra arrumar um namorado? Não. As pessoas são diferentes, e todas elas, repito, todas, acham alguém que goste delas do jeitinho que são. Nunca vi uma pessoa morrer sozinha por ser feia; ou não ter amigos por não ser bonita o suficiente. Isso não existe.
Vamos esquecer o estereótipo de "perfeição" que nos é imposto todos os dias. As atrizes de televisão são pessoas normais, acreditem. Estão duvidando? Joguem no Google o nome da atriz que você tanto venera junto a "antes da fama". É deprimente, ou, melhor dizendo, é até inspirador. O "perfeito" não existe. Porque a "perfeição" é uma vasilável. As amar cada pedacinho do seu ser, você acaba se tornando perfeita para si e, consequentemente, para o mundo. Esse é o segredo daquelas mulheres imponentes que andam por aí com o nariz empinado como se nada mais importasse. É que, realmente, não importa. Elas se bastam. E não porque são lindas, altas ou magras. Porque se amam. E o amor-próprio explode como uma supernova, iluminando todos à sua volta.
É impossível falar de amor-próprio e não me lembrar de uma amiga que tive na escola. Essa amiga não era a mais bonita do colégio, não mesmo. Baixinha, com alguns quilinhos a mais, dentes cercados por um aparelho que ela se orgulhava de possuir e cabelos encaracolados. Fugia do clichê da garota alta, magra, de cabelos lisos, que sempre aparece nos filmes hollywoodianos. Acontece que a Bia era o maior sucesso entre os garotos. Era engraçada, bem-humorada, sensual e misteriosa. Bia foi o motivo de eu chorar e espernear pedindo um aparelho pro meu pai.
— Mas pai, eu pre-ci-so de um aparelho. A Bia lá da sala tem um e é lindo! — choramingava.
— Filha, os seus dentes são perfeitos. Só usa aparelho quem precisa corrigir algum problema — ele explicava, paciente.
Meu pai é dentista e ora, ele podia muito bem colocar um aparelho em mim sem que eu precisasse, né?
— E daí? Eu quero um aparelho para poder usar as borrachinhas coloridas iguais às da Bia. Por favor, pai!
— Já disse que não vou colocar aparelho nenhum nos seus dentes — e encerrava o assunto.
Eu não entendia qual era o segredo da Bia, só pedia ter o aparelho. Aquele aparelho colorido com borrachinhas, que alternavam entre rosa, preto, até verde, amarelo e azul na época da Copa, tinha algum mistério. E realmente tinha. Mas eu só fui entender alguns anos depois. A Bia se amava. Profunda e intensamente. Não era o aparelho que chamava a atenção das pessoas, e sim o seu amor-próprio.
Aquele que não se ama procura no outro um amor incondicional que deveria existir dentro de si mesmo. Aquele que se ama se basta. Estar ao lado de alguém é apenas o simples fato de possuir uma boa companhia para desfrutar os seus dias. E, quem sabe, dosar um pouco do amor que já existe no próprio coração.

Fonte: (Autor, 2020).

Com os pontos devidamente separados para cada linha, é possível calcular os pontos de destino. As coordenadas *y* de destino de cada linha foram calculadas com a média de todas as coordenadas *y* dos pontos da mesma linha. Já as coordenadas *x* de destino são as mesmas que a as coordenadas *x* da origem, pois os pontos se mantêm na mesma coluna.

Com os pontos de origem e os de destino calculados, é possível fazer a retificação da imagem. Para isso foi utilizado o algoritmo *Thin-Plate Splines* (TPS) (BOOKSTEIN, 1989). Essa técnica é bastante utilizada e apropriada para a interpolação de superfícies sobre dados irregularmente espaçados. Sua formulação garante restrições de que a superfície interpolante apresente mínima energia de deformação e que seja suave (MAGNA JÚNIOR, 2012).

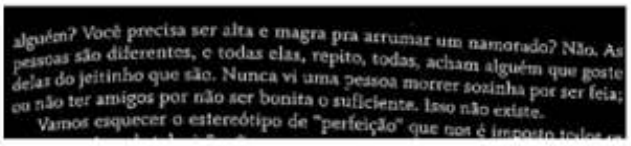
Segundo Bookstein (1989), a energia de curvatura em um ponto com coordenadas (*x*, *y*) é expressa pela Equação 9.

$$I_f = \iint_{R^2} \left(\left(\frac{\partial^2 z}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 z}{\partial x \partial y} \right)^2 + \left(\frac{\partial^2 z}{\partial y^2} \right)^2 \right) dxdy \quad (9)$$

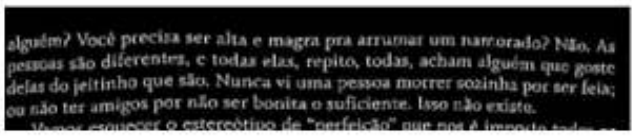
Com dois conjuntos de pontos homólogos (origem e destino) são geradas funções de mapeamento que relacionam pontos de um conjunto ao outro. A função de mapeamento da TPS pode ser decomposta em uma transformação linear e uma parcela representando as distorções não lineares (BOOKSTEIN, 1989).

O resultado da distorção aplicando o algoritmo TPS pode ser conferido na Figura 13.

Figura 13. Trecho contendo linhas da imagem da página de um livro. a) imagem original com deformação. b) após a retificação da imagem.


alguém? Você precisa ser alta e magra pra arrumar um namorado? Não. As pessoas são diferentes, e todas elas, repito, todas, acham alguém que goste delas do jeitinho que são. Nunca vi uma pessoa morrer sozinha por ser feia; ou não ter amigos por não ser bonita o suficiente. Isso não existe.
Vamos esquecer o estereótipo de "perfeição" que nos é imposto todos os

a)


alguém? Você precisa ser alta e magra pra arrumar um namorado? Não. As pessoas são diferentes, e todas elas, repito, todas, acham alguém que goste delas do jeitinho que são. Nunca vi uma pessoa morrer sozinha por ser feia; ou não ter amigos por não ser bonita o suficiente. Isso não existe.
Vamos esquecer o estereótipo de "perfeição" que nos é imposto todos os

b)

Fonte: (Autor, 2020).

2.4. Reconhecimento dos caracteres

Neste trabalho, com o intuito de realizar a identificação de caracteres contidos na imagem, foi utilizado o algoritmo Tesseract (SMITH, 2007).

O Tesseract é uma técnica OCR (*Optical Character Recognition*) codificada em C/C++ e desenvolvido pela HP (Hewlett-Packard) entre 1984 e 1994. Em 2005, a HP liberou Tesseract como um *software* de código aberto, e desde então o projeto vem sendo mantido pela Google.

O Tesseract utiliza um limiar global para a separação dos pixels de fundo e os de primeiro plano. O método de limiarização utilizado na binarização é o de Otsu (OTSU, 1979). Uma análise de componentes conexos é feita e em seguida são encontradas as regiões dos caracteres.

O processo de reconhecimento é realizado na palavra como um todo. Inicialmente, os caracteres são classificados, e caso a palavra for considerada insatisfatória, o Tesseract busca alternativas para os caracteres de baixa acurácia.

A cadeia de caracteres que possuir a melhor classificação é a que será considerada como resultado da palavra. Cada palavra aceita passa a fazer parte do treinamento para o classificador adaptativo. As palavras da imagem

são processadas duas vezes, na primeira vez, as que tiveram um bom reconhecimento são adicionadas como treinamento para o classificador adaptativo. No segundo momento, as palavras que não atingiram um bom percentual de reconhecimento na primeira vez, são processadas e reconhecidas novamente, pois agora o classificador adaptativo possui mais informações.

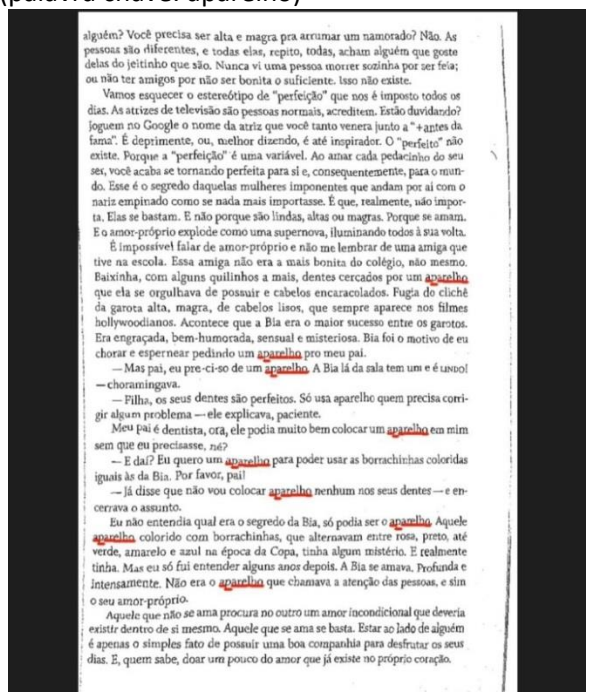
Dessa forma o Tesseract consegue reconhecer e classificar as palavras contidas na imagem e identificar não só as palavras, mas também as coordenadas que cada palavra ocupa na imagem.

2.5. Busca por palavras chave

Com as palavras devidamente identificadas, o processo de buscar a palavra desejada se torna bem simples. É feita uma busca sequencial, onde cada palavra contida na imagem é comparada com a que se está sendo buscada.

No fim do processo, todas as palavras encontradas que são idênticas a procurada, são grifadas como mostra a Figura 14.

Figura 14. Resultado após a conclusão da busca (palavra chave: aparelho)



alguém? Você precisa ser alta e magra pra arrumar um namorado? Não. As pessoas são diferentes, e todas elas, repito, todas, acham alguém que goste delas do jeitinho que são. Nunca vi uma pessoa morrer sozinha por ser feia; ou não ter amigos por não ser bonita o suficiente. Isso não existe.

Vamos esquecer o estereótipo de "perfeição" que nos é imposto todos os dias. As atrizes de televisão são pessoas normais, acreditem. Estão duvidando? Joguem no Google o nome da atriz que você tanto venera junto a "+antes da fama". É deprimente, ou, melhor dizendo, é até inspirador. O "perfeito" não existe. Porque a "perfeição" é uma variável. Ao amar cada pedacinho do seu ser, você acaba se tornando perfeita para si e, consequentemente, para o mundo. Esse é o segredo daquelas mulheres imponentes que andam por aí com o nariz empinado como se nada mais importasse. É que, realmente, não importa. Elas se bastam. E não porque são lindas, altas ou magras. Porque se amam. E o amor-próprio explode como uma supernova, iluminando todos à sua volta.

É impossível falar de amor-próprio e não me lembrar de uma amiga que tive na escola. Essa amiga não era a mais bonita do colégio, não mesmo. Baixinha, com alguns quilinhos a mais, dentes cercados por um aparelho que ela se orgulhava de possuir e cabelos encaracolados. Fugia do clichê da garota alta, magra, de cabelos lisos, que sempre aparece nos filmes hollywoodianos. Acontece que a Bia era o maior sucesso entre os garotos. Era engraçada, bem-humorada, sensual e misteriosa. Bia foi o motivo de eu chorar e espernear pedindo um aparelho pro meu pai.

— Mas pai, eu pre-ci-so de um aparelho. A Bia lá da sala tem um e é LINDO! — choramingava.

— Filha, os seus dentes são perfeitos. Só usa aparelho quem precisa corrigir algum problema — ele explicava, paciente.

Meu pai é dentista, ora, ele podia muito bem colocar um aparelho em mim sem que eu precisasse, né?

— E daí? Eu quero um aparelho para poder usar as borrachinhas coloridas iguais às da Bia. Por favor, pai!

— Já disse que não vou colocar aparelho nenhum nos seus dentes — e encerrava o assunto.

Eu não entendia qual era o segredo da Bia, só podia ser o aparelho. Aquele aparelho colorido com borrachinhas, que alternavam entre rosa, preto, até verde, amarelo e azul na época da Copa, tinha algum mistério. E realmente tinha. Mas eu só fui entender alguns anos depois. A Bia se amava. Profunda e intensamente. Não era o aparelho que chamava a atenção das pessoas, e sim o seu amor-próprio.

Aquele que não se ama procura no outro um amor incondicional que deveria existir dentro de si mesmo. Aquele que se ama se basta. Estar ao lado de alguém é apenas o simples fato de possuir uma boa companhia para desfrutar os seus dias. E, quem sabe, doar um pouco do amor que já existe no próprio coração.

Fonte: (Autor, 2020).

3. RESULTADOS

Para avaliar a eficiência do método proposto, foi utilizado um smartphone Xiaomi Redmi Note 8 Pro com processador MediaTek Helio G90T e 6 GB de RAM, onde foram realizados experimentos com 100 imagens capturadas utilizando o aplicativo desenvolvido. Foram selecionadas palavras com grande frequência nas imagens das páginas de livros para serem usadas como palavras chave de busca. A Tabela 1 apresenta os resultados obtidos.

Tabela 1. Resultados obtidos.

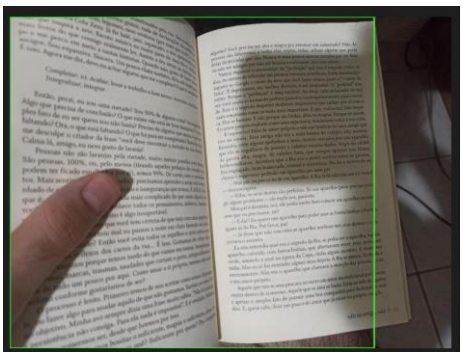
Resultados	
Fotos capturadas	100
Palavras buscadas em cada página	5
Total de ocorrências	1997
Palavras encontradas corretamente	1770
Palavras não encontradas	173
Palavras encontradas incorretamente	54
Porcentagem de acerto	88,6%

Fonte: (Autor, 2020).

O método proposto neste trabalho obteve resultados satisfatórios, embora apresente um tempo elevado de execução, em média 90 segundos, principalmente devido ao algoritmo *Thin-Plate Splines* ter que calcular a interpolação de muitos pontos. Os erros ocorreram quando a imagem capturada possui grande diferença de iluminação, em algumas regiões que estavam muito iluminadas e em outras que continham sombras. Essa diferença de iluminação atrapalhou a segmentação, como pode ser visualizado na Figura 15(a).

Imagens com pouca qualidade também dificultavam o processo de encontrar pontos nas linhas, produzindo erros na identificação de caracteres como mostrado na Figura 15(b). O mesmo acontece caso a página do livro possua figuras, tabelas ou qualquer outro elemento, além de texto. Nesses casos, o algoritmo tenta encontrar pontos sobre esses elementos, o que interfere negativamente em encontrar os pontos corretamente sobre as linhas, como é o caso da Figura 15(c), que possui títulos entres os parágrafos em caixa alta.

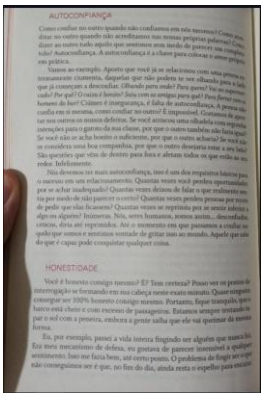
Figura 15. Exemplos de falhas ocorridas. a) Falha na segmentação. b) Falha na identificação dos pontos. c) Falha devido a elementos diferentes.



a)



b)



c)

Fonte: (Autor, 2020).

4. CONCLUSÕES

Este trabalho objetivou o desenvolvimento de uma metodologia para busca de palavras chave em imagens de páginas de livros. Os resultados apresentados mostram que foi obtido uma boa taxa de acerto, comprovando ser uma alternativa promissora para projetos futuros com este tema.

A busca de palavras foi realizada com imagens capturadas pelo smartphone e processada, levando um grande tempo de execução, em média 90 segundos, podendo variar dependendo da quantidade de linhas contidas na página. Seria interessante, em trabalhos futuros, que a metodologia seja aprimorada, para que possa ser executado em tempo real. Um bom caminho para se conseguir isso, seria alterar o *Thin-Plate Splines*, já que para todos os pontos obtidos na imagem é calculada a interpolação dos pixels vizinhos, mesmo que os pontos de origem e os de destino sejam muito próximos. Isso acarreta em um tempo de processamento que poderia ser evitado, pois a alteração desses pontos não faria diferença na imagem final.

Também seria interessante aumentar a abrangência das páginas de livros, podendo fazer a busca também em páginas que contenham figuras, isso necessitaria do desenvolvimento de novos algoritmos para tratar essa questão.

REFERÊNCIAS

AGRAWAL, N.; KAUR, A. An Algorithmic Approach for Text Recognition from Printed/Typed Text Images. *In*: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, DATA SCIENCE & ENGINEERING. 8., 2018. **Anais** […]. Noida, India, 2018. https://doi.org/10.1109/CONFLUENCE.2018.8442875

BOOKSTEIN, F. L. Principal warps: thin-plate splines and the decomposition of deformations. **IEEE Transaction on Pattern Analysis Machine Intelligence**, v. 11, n. 6, p.567–585, 1989. https://doi.org/10.1109/34.24792

BUNCH, J. R.; HOPCROFT, J. E. Triangular factorization and inversion by fast matrix multiplication. **Mathematic of Computation**, v. 28, n. 125, p 231-236, 1974. https://doi.org/10.1090/S0025-5718-1974-0331751-8

HARRIS, P.; STEPHENS, M. **A combined corner and edge detector.** Plessey Research Roke Manor, Reino Unido. 1988. https://doi.org/10.5244/C.2.23

KATZ, F. S. **Estudo de comportamento de consumo de livros digitais**. 2011. 95 f. TCC (Graduação) - Curso de Administração, UFRS, Rio Grande do Sul, 2011.

KHAOULA, E.; GARCIA, C.; MAMALET, F.; SÉBILLOT, P. Text Recognition in Multimedia Documents: A Study of two Neural-based OCRs Using and Avoiding Character Segmentation. **International Journal on Document Analysis and Recognition (IJDAR)**, v. 17 n. 1, p. 1-13, 2013. https://doi.org/10.1007/s10032-013-0202-7

KAW, A. K.; KALU, E. E.; NGUYEN, D. Numerical methods with applications: chapter 04.06 Gaussian Elimination. University of South Florida. 2018. Disponível em: http://mathforcollege.com/nm/mws/gen/04sle/mws_gen_sle_txt_gaussian.pdf. Acesso em: 13 dez 2020.

KUHN, D. M.; CERVI, C. R.; MANICA, E. Extração de elementos textuais em imagens capturadas por smartphones: análise da relação entre as

características das imagens e a eficácia da extração. *In:* ESCOLA REGIONAL DE BANCO DE DADOS (ERBD). **Anais [**...] Porto Alegre, Rio Grande do Sul, Sociedade Brasileira de Computação, 2018.

LIANG, J.; DOERMANN, D.; LI, H. Camera-based analysis of text and documents: a survey. **International Journal on Document Analysis and Recognition** (IJDAR), v. 7, n. 2-3, p. 84–104, 2005. https://doi.org/10.1007/s10032-004-0138-z

MAGNA JÚNIOR, J. P. **O uso de Thin-Plate Splines na transformação de coordenadas com modelagem de distorções entre realizações de referenciais geodésicos**. 2012. 117 f. Tese (Doutorado) - Universidade Estadual Paulista, Faculdade de Ciências e Tecnologia, 2012.

MIRANDA, R. A. R, SILVA, F. A, ARTERO, A. O., PITERI, M. A, Handwritten Character Recognition based on Frequency, Character-edge Distances and Densities. WORKSHOP DE VISÃO COMPUTACIONAL (WVC 2013), 9., 2013, Rio de Janeiro. **Anais** [...]. Rio de Janeiro, RJ, 2013.

OLIVEIRA, G. H.; SILVA, F. A.; PEREIRA, D. R.; ALMEIDA, L. L.; ARTERO, A. O. BONORA A. F.; ALBUQUERQUE, V. H. C. Automatic Detection and Recognition of Text-Based Traffic Signs from images. **IEEE Latin America Transactions**, v. 16, n. 12, p. 2947-2953, 2018. https://doi.org/10.1109/TLA.2018.8804261

OTSU, N. A Threshold Selection Method from Gray-Level Histograms. **IEEE Transactions on Systems, Man, And Cybernetics**, v. 9, n. 1, p. 62-66, 1979. https://doi.org/10.1109/TSMC.1979.4310076

REIS, B.; TEIXEIRA, J. M. X. N., TEICHRIEB, V.; KELNER, J. Perspective Correction Implementation for Embedded (Marker-Based) Augmented Reality. *In*: WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA – WRVA. 5., 2008, Bauru. **Anais** [...]., Baurú: Unesp, 2008.

SIDHWA, H.; KULSHRESTHA, S.; MALHOTRA, S.; VIRMANI, S. Text Extraction from Bills and Invoices. *In*: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTING, COMM UNICATION CONTROL AND NETWORKING (ICACCCN), Greater Noida (UP), India, 2018. https://doi.org/10.1109/ICACCCN.2018.8748309

SILVA, F. A.; ARTERO, A. O.; PAIVA, M. S. V.; BARBOSA, R. L. **Reconhecimento de Caracteres Baseado em Regras de Transições entre Pixels Vizinhos.** Avanços em Visão Computacional. Curitiba: Omnipax, 2012. https://doi.org/10.7436/2012.avc.14

SMITH, R. An Overview of the Tesseract OCR Engine. Ninth International Conference On Document Analysis And Recognition. *In*: NINTH INTERNATIONAL CONFERENCE ON DOCUMENT ANALYSIS AND RECOGNITION (ICDAR 2007), 2007. https://doi.org/10.1109/ICDAR.2007.4376991

SUZUKI, S.; ABE, K. Topological Structural Analysis of Digitized Binary Images by Border Following. COMPUTER VISION, GRAPHICS, AND IMAGE, **Processings...**, v. 30, n. 1, 1985. https://doi.org/10.1016/0734-189X(85)90016-7