Toth: Uma abordagem para extração de elementos textuais em imagens com linhas de texto inclinadas

Daniel M. Kuhn[1], Cristiano R. Cervi[1], Edimar Mânica[2]

[1]Instituto de Ciências Exatas e Geociências – UPF - Passo Fundo - RS

[2]Campus Ibirubá - IFRS - Ibirubá - RS

138714@upf.br, cervi@upf.br, edimar.manica@ibiruba.ifrs.edu.br

Abstract. *Optical character recognition software is designed to convert document textual elements into editable and searchable text. This task presents specific challenges when the textual elements are in images captured by smartphone cameras. One of these challenges is the skew of text lines that affects the effectiveness and efficiency of current recognition methods. This work presents an approach to extract textual elements in images with inclined text lines. The experiments demonstrated that the approach obtained a significant increase of effectiveness in relation to the baseline, at the moment that it also presented superior efficiency.*

Resumo. *Softwares de reconhecimento óptico de caracteres têm como propósito converter elementos textuais de documentos em texto editável e pesquisável. Essa tarefa apresenta desafios específicos quando os elementos textuais estão em imagens capturadas por câmeras de smartphones. Um desses desafios é a inclinação das linhas do texto que afeta a eficácia e eficiência dos métodos de reconhecimento atuais. Este trabalho apresenta uma abordagem para extrair elementos textuais em imagens com linhas de texto inclinadas. Os experimentos demonstram que a abordagem obteve um aumento de eficácia significativo em relação ao baseline, ao instante em que também apresentou eficiência superior.*

1. Introdução

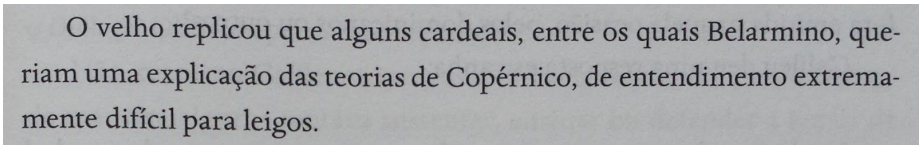
A análise de *Big Data* é um aspecto chave da sociedade moderna uma vez que permite criar conhecimento a partir de dados. Essa análise traz o conhecimento para o indivíduo de uma forma direta e facilitada permitindo a emancipação das pessoas e as habilitando a tomarem decisões com mais embasamento [Manica, Dorneles and Galante 2017]. Problemas de heterogeneidade, escalabilidade, complexidade e privacidade impedem o progresso de todos os estágios do *pipeline* que extrai valor a partir de dados [Labrinidis and Jagadish 2012]. Nesse contexto, os problemas iniciam durante a aquisição de dados porque muitos dados não estão nativamente em um formato estruturado e estruturar tal conteúdo para análise futura é o principal desafio [Agrawal et al 2012].

Um exemplo de dados relevantes em um formato não estruturado é observado em textos presentes em imagens postadas nas redes sociais. Estima-se que só no instagram - atualmente a maior rede social de fotografias - sejam postadas em média 52 milhões de fotografias todos os dias [Statistic Brain, 2017]. Um percentual dessas

imagens contém elementos textuais. Esse percentual, embora pequeno, representa um grande número de postagens, dada a escalabilidade da rede social. Extrair esses elementos textuais é útil para aprender mais sobre o usuário e fornecer recomendações mais precisas de produtos e serviços.

A extração de conteúdos textuais em imagens é realizada através do uso de softwares de Reconhecimento Óptico de Caracteres (OCR – *Optical Character Recognition*). O *OCR* é um processo de reconhecimento visual que converte documentos de texto em texto editável e pesquisável [Berchmans and Kumar 2014].

A Figura 1(a) ilustra uma imagem de um trecho de um livro capturada por um *smartphone* submetida a um software de *OCR* (entrada). A Figura 1(b) apresenta os elementos textuais extraídos da Figura 1(a).



(a)

O velho replicou que alguns cardeais, entre os quais Belarmino, queriam uma explicação das teorias de Copérnico, de entendimento extremamente difícil para leigos.

(a)

Figura 1. Exemplo de uma imagem de entrada e os elementos textuais resultantes do processo de extração.

Nos últimos trinta anos, um número substancial de pesquisas acerca de mecanismos de *OCR* foram realizadas [Islam and Noor 2016]. A grande maioria dos esforços destinou-se a solucionar problemas decorrentes da digitalização de documentos de texto através do uso de dispositivos de *scanner*, o que resultou na obtenção de altas taxas de precisão de extração em documentos dessa natureza [Asad et al 2016].

Entretanto, os métodos de pré-processamento de imagens aplicados em documentos escaneados são em diversos casos inapropriados ou insuficientes quando destinados a otimizar o reconhecimento de caracteres de imagens capturadas por câmeras de *smartphones*. Isso ocorre porque as características encontradas em imagens escaneadas são, em sua grande maioria, distintas das características presentes em arquivos de imagens obtidas através da câmera de *smartphones*. As imagens capturadas por câmeras podem apresentar variação de iluminação, linhas de texto inclinadas, baixa resolução, desfocagem e distorção de perspectiva, apresentando layouts complexos e interação entre o conteúdo e o plano de fundo [Liang, Doermann and Li 2005] [Kuhn, Cervi and Manica 2018].

Considerando o contexto apresentado, este trabalho tem como objetivo o desenvolvimento de uma abordagem denominada *Toth*, cujo propósito é extrair elementos textuais de imagens capturadas por *smartphones* que apresentam linhas de texto inclinadas. Como contribuição principal, este trabalho contempla a implementação de um pré-processamento para rotacionar imagens com linhas de texto inclinadas. Para avaliar as solução proposta, foram realizados experimentos com imagens sintéticas contendo trechos de livros, capturadas por *smartphones*. Comparando os resultados com um *baseline*, a abordagem proposta obteve um aumento de eficácia significativo, ao

instante que também foi capaz de concluir a extração em tempos significativamente inferiores quando comparado ao *baseline*.

Este artigo está organizado da seguinte forma. Na Seção 2 são discutidos os trabalhos relacionados. A Seção 3 descreve a abordagem proposta. Na Seção 4 são apresentados os experimentos realizados, bem como são discutidos seus resultados. Finalmente, a Seção 5 apresenta as considerações finais e os trabalhos futuros.

2. Trabalhos relacionados

Esta seção descreve cinco trabalhos que possuem relação ao contexto no qual este trabalho está inserido.

No trabalho [Asad et al 2016] apresenta-se um sistema de *OCR* baseado em redes LSTM (*Long Short Term Term*), capaz de reconhecer caracteres borrados decorrentes de movimentos indesejados. Redes LSTM, são um tipo especial de redes neurais recorrentes (RNN – *Recurrent neural network*) com capacidade de recordar informações por longos períodos de tempo [Olah 2015].

Em [Kil et al 2018] apresenta-se uma abordagem para corrigir adversidades referentes à perspectiva das imagens, tratando linhas que apresentam aspecto curvilíneo. A abordagem se baseia na premissa de que a maioria dos segmentos de linha das imagens são alinhados horizontalmente ou verticalmente, codificando essas propriedades em uma função de custo. Minimizando a função, obtém-se os parâmetros de transformação para posição da câmera, curva da página e comprimento focal da câmera, utilizando-os para realizar a correção de perspectiva das imagens. Esta abordagem apresenta resultados satisfatórios tanto para documentos de texto quanto imagens com fundo rico em detalhes, como rótulos de produtos.

Em [Smith 1987] foi proposto uma nova abordagem para reconhecimento de caracteres. Esse trabalho deu origem ao motor de *OCR Tesseract* [Tesseract 2015]. Em 2005, *Tesseract* passou a ser um projeto *Open Source* e desde 2006 vem sendo desenvolvido pela *Google Inc*[1]. *Tesseract* provê suporte a Unicode, capaz de reconhecer mais de 100 linguagens diferentes. Este motor de *OCR* é totalmente treinável, sendo possível adicionar novos símbolos e até mesmo novos idiomas inteiros. *Tesseract* regularmente é citado entre os *softwares* de *OCR Open Source* com maior acurácia [Gabasio 2013] [Dhiman 2013]. Considerando sua acurácia, o fato de ser *Open Source* e a possibilidade de aplicar processos de treinamento, entendemos que tais fatores são relevantes para o contexto da nossa proposta, o que foi determinante para sua adoção como *baseline* e também como motor de extração neste trabalho.

Em um trabalho prévio [Kuhn, Cervi and Manica 2018], foi avaliada a relação entre as características das imagens e a eficácia da extração de elementos textuais em imagens capturadas por *smartphones* submetidas ao *Tesseract*. O experimento avaliou o *Tesseract* com o modo de segmentação de página padrão para o cliente *Tesseract* para a interface de linha de comando (*PSM_AUTO*). Este trabalho identificou que as linhas de texto inclinadas e a variação de iluminação afetam a eficácia da extração.

Este trabalho diferencia-se dos trabalhos acima citados por implementar etapas de pré-processamento para tratar imagens com linhas de texto inclinadas.

[1]Disponível em: <https://www.google.com>. Acessado em: 10/10/2018.

3. Abordagem Toth

Esta seção apresenta a abordagem *Toth*, que visa a extração de elementos textuais em imagens capturadas por *smartphones*. A subseção 3.1 apresenta a visão geral da abordagem. Nas seções que seguem, são especificadas cada etapa da abordagem *Toth*.

3.1. Visão Geral

Este trabalho tem como objetivo o **desenvolvimento de uma abordagem para extração de elementos textuais em imagens capturadas por *smartphones***. A Figura 2 apresenta as principais etapas da abordagem *Toth*.

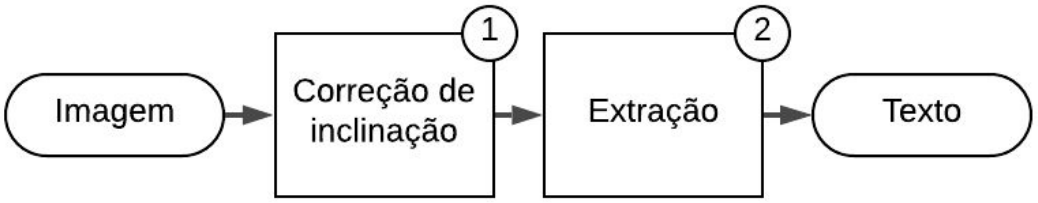


Figura 2. Fluxo de execução da abordagem *Toth*.

Toth é constituído das seguintes etapas: (1) *Correção de inclinação* - pré-processamento responsável por rotacionar imagens com linhas de texto inclinadas; e (2) *Extração* - obtenção dos elementos textuais contidos nas imagens. O processo inicia com a submissão de uma imagem, passa pelas etapas 1 e 2, resultando no conteúdo textual da imagem em formato editável.

3.2. Correção de inclinação

A correção da inclinação rotaciona as imagens que possuem linhas de texto inclinadas. Para realizar a rotação das imagens foi proposto o algoritmo 1.

Algorithm 1: RI (Rotation of Images - Rotação de Imagens

Input: I_i
Output: I_r
1 $I_i \leftarrow$ converte_escala_cinza(I_i);
2 $Angle \leftarrow$ PDAL(I_i);
3 $I_r \leftarrow$ PRI(I_j, $Angle$);
4 **return** I_r;

Como entrada, o algoritmo 1 recebe uma imagem e como saída, obtém-se a imagem rotacionada. Inicialmente, converte-se a imagem para escala de cinza (linha 1). Na linha 2, o algoritmo PDAL (Algoritmo 2) retorna o ângulo de rotação para a imagem. Por fim (linha 3), o algoritmo PRI (Algoritmo 3) realiza a rotação da imagem.

Algorithm 2: PDAL (Process to Detect the Angle of Lines - Processo para detectar o ângulo das Linhas)

Input: I_i, Min_line
Output: $Angle$

```
1 I_b ← binarizacao_adaptativa(I_i, THRESH_BINARY_INV);
2 I_b ← dilata(I_b);
3 I_b ← contrai(I_b);
4 L ← identifica_linhas(I_b, Min_lines);
5 sum ← 0;
6 for L_i ∈ L do
7     sum += calcula_angulo(L_i);
8 end
9 return media(sum / size(L));
```

Algorithm 3: PRI (Process to Rotates Image - Processo para Rotacionar Imagem)

Input: I_i,$Angle$
Output: I_r

```
1 Rad ← converte_radianos(Angle);
2 W_i ← calcula_nova_largura(Rad);
3 H_i ← calcula_nova_altura(Rad);
4 C_i ← obtem_centro_imagem(I_i);
5 M_i ← obtem_ratrix_rotacao(C_i, Angle);
6 M_i ← ajusta_centro(M_i, W_i, H_i, C_i);
7 I_r ← aplica_transformacao_afim (M_i, I_i);
8 return I_r;
```

O algoritmo 2 é responsável por identificar as linhas de texto e calcular o ângulo médio de inclinação. Como entrada, o algoritmo recebe uma imagem e o comprimento mínimo das linhas de texto que serão consideradas. Como saída, o algoritmo retorna o ângulo para rotacionar a imagem. Na linha 1, realiza-se a binarização adaptativa da imagem de entrada. Neste procedimento converte o texto para a cor branca e o fundo da imagem para a cor preta. Na linha 2, aplica-se uma dilatação para conectar no sentido horizontal os *pixels* que compõem as palavras que formam o texto. Na linha 3, realiza-se a contração parcial dos *pixels* dilatados na linha 2. O objetivo do tratamento realizado nas linhas 2 e 3 é conectar todas as palavras de uma determinada linha. Na linha 4, identifica-se as retas formadas dentro de cada linha de texto, respeitando o limiar de comprimento definido. Na linha 5, inicializa-se uma variável para armazenar o somatório dos ângulos da reta. No laço 6-8, realiza-se o somatório dos ângulos das retas. Por fim, calcula-se o ângulo médio das retas.

O algoritmo 3 é responsável por rotacionar as imagens. Como entrada, recebe a imagem a ser rotacionada e o ângulo da rotação desejada. O algoritmo retorna a imagem devidamente rotacionada. Na linha 1, converte-se o ângulo de rotação para radianos. Nas linhas 2 e 3, calcula-se, respectivamente, a nova largura e altura que a imagem terá após a rotação. A linha 4 obtém o ponto central da imagem de entrada. Na linha 5, obtém-se a *matriz de rotação 2D*[2] da imagem a ser rotacionada. Na linha 6 ajusta-se o centro da matriz de rotação. Por fim (linha 7), submete-se a imagem para realizar a *transformação afim*[3], conforme a matriz de rotação especificada.

[2] Documentação do OpenCv: matriz de rotação 2D. Disponível em: <https://docs.opencv.org/3.1.0/da/d54/group__imgproc__transform.html#gafbbc470ce83812914a70abfb604f4326> Acessado em: 25/08/2018.

[3] Documentação do OpenCv: transformação afim. Disponível em: <https://docs.opencv.org/3.1.0/da/d54/group__imgproc__transform.html#ga0203d9ee5fcd28d40dbc4a1ea4451983> Acessado em: 25/08/2018.

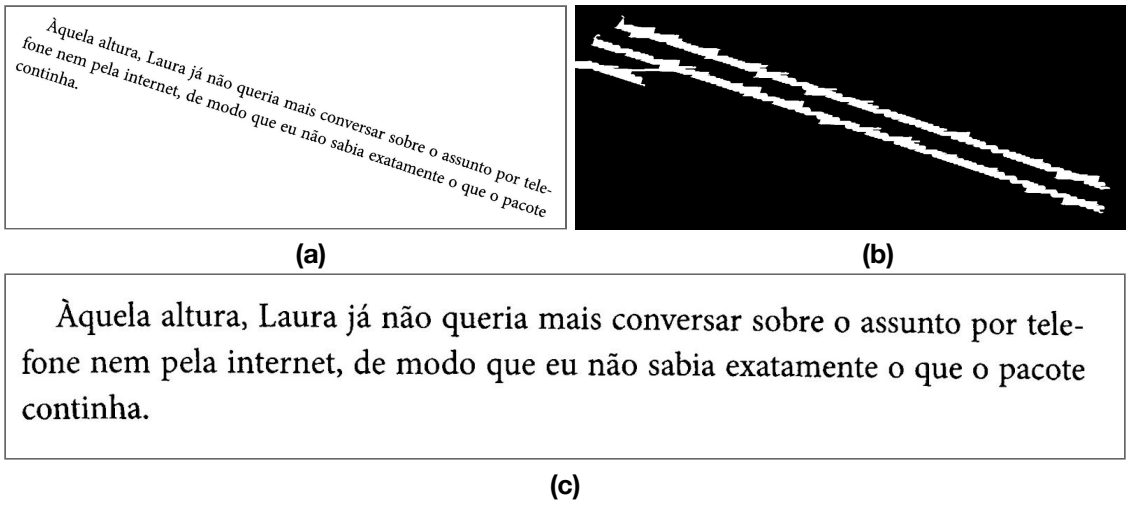


Figura 3. Eficácia da extração em relação às características de interesse.

A Figura 3(a) apresenta um exemplo de imagem submetida ao algoritmo 1. A Figura 3(b) apresenta a imagem resultante do processamento para identificar as linhas de texto do algoritmo 2. A Figura 3(c) apresenta a imagem resultante do algoritmo 1 sobre a imagem de entrada (Figura 2(a)). Observa-se que a imagem foi devidamente rotacionada, alinhando as linhas de texto no sentido horizontal.

3.3. Extração

Na etapa de *Extração*, os elementos textuais contidos nas imagens são extraídos. A abordagem *Toth* faz uso do motor *OCR Tesseract*. Este motor de *OCR* é regularmente citado entre os *softwares* de *OCR Open Source* com maior acurácia. Além disso, é possível aplicar processos de treinamento para identificar novos símbolos.

4. Avaliação experimental

Esta seção descreve os experimentos realizados com o intuito de avaliar a eficácia e a eficiência da extração da abordagem *Toth*. Esta seção está organizada da seguinte forma. A subseção 4.1 apresenta a base de dados utilizada. A seção 4.2 define as métricas utilizadas e o *baseline*. A subseção 4.3 apresenta as configurações do ambiente onde os experimentos foram conduzidos. A subseção 4.4 descreve os experimentos realizados e os resultados obtidos. Finalmente, a subseção 4.5 discute os casos de falha da abordagem *Toth*.

4.1. Bases de dados

Para os experimentos utilizou-se uma base de dados, a qual foi criada exclusivamente para este trabalho. A base de dados BD_ROTACIONA possui 100 imagens com trechos de livros capturadas por *smartphones* com linhas de texto inclinadas. Essa base de dados foi criada para avaliar o comportamento da abordagem *Toth* e do *Tesseract* em imagens com texto inclinado. Para compor a base de dados, foram capturadas imagens há aproximadamente 25 centímetros de distância em relação às páginas dos livros. Após, a área de interesse das imagens foi definida por um usuário especialista, não havendo presença de termos ou caracteres pertencentes aos parágrafos vizinhos. As imagens

foram binarizadas[4] e automaticamente rotacionadas por um algoritmo que atribuiu a cada uma das imagens uma rotação aleatória entre [-30, 30] graus. Portanto, essa é uma base sintética que simula a ocorrência de linhas inclinadas.

A base de dados possui os gabaritos que descrevem a extração ideal para cada imagem. Os gabaritos foram criados por um usuário especialista que transcreveu manualmente o conteúdo textual de cada imagem. O texto em formato digital foi então armazenado, mantendo a devida relação entre o arquivo da imagem e seu respectivo conteúdo textual.

4.2. Métricas e baseline

Para avaliar a eficácia da extração, foram adotadas as métricas tradicionais de recuperação de informação: precisão *(precision)*, revocação *(recall)* e F1 *(F-measure).* Para este trabalho, contextualizando a definição de Precisão e Revocação apresentada em [Baeza-Yates and Ribeiro-Neto 2013], assume-se que a precisão mede a fração dos termos extraídos que é relevante e a revocação mensura a fração dos termos relevantes que foi extraída. Portanto:

$$precisão = \frac{|\,Termos\ relevantes \cap Termos\ extraídos\,|}{|\,Termos\ extraídos|}, \tag{1}$$

$$revocação = \frac{|\,Termos\ relevantes \cap Termos\ extraídos\,|}{|\,Termos\ \ relevantes\,|}. \tag{2}$$

Um termo relevante é aquele que foi extraído de uma imagem corretamente e, portanto, está presente no gabarito. Por fim, o F1 consiste na média harmônica entre os índices de precisão e revocação, com o objetivo de fornecer um só índice de medida. Quanto maior a revocação, precisão e a F1, maior a eficácia da abordagem. A F1 é definida como segue:

$$F1 = 2 \cdot \frac{precisão \cdot revocação}{precisão + revocação}. \tag{3}$$

Ressalta-se que acentuações, distinção de letras maiúsculas e minúsculas, bem como, pontuações foram desconsideradas na análise.

Utilizou-se o tempo de extração como métrica de eficiência. O tempo de extração indica o tempo total gasto para extrair os elementos textuais de uma determinada imagem. No que se refere à abordagem *Toth,* o tempo de extração consiste no tempo de pré-processamento das imagens, acrescido do tempo demandado pelo extrator.

A significância estatística foi verificada através do teste T (Student's t-test) [FINN 1996]. O limiar de significância utilizado foi $\alpha = 0{,}05$ (padrão). Portanto, quando o valor de p_bicaudal calculado pelo Teste T for menor que α, confirma-se com uma confiança de 95% que há diferença estatisticamente significante entre as abordagens comparadas.

A abordagem Toth foi comparada com o *Tesseract*. Os fatores determinantes para a adoção deste motor *OCR* como *baseline* foram: (i) sua acurácia; (ii) o fato de ser um software Open Source; (iii) a possibilidade de aplicar processos de treinamento.

[4] Documentação do OpenCv: binarização. Disponível em: <https://docs.opencv.org/3.4/d7/d4d/tutorial_py_thresholding.html> Acessado em: 14/02/2019.

4.3. Ambiente experimental

O extrator adotado no experimento foi o *Tesseract* na versão 3.05. Para este trabalho, utilizou-se uma versão pré-compilada do *Tesseract* para *smartphones* com o *sistema operacional Android (Tess-two)*[5].

A abordagem *Toth* faz uso das funções de processamento de imagem da biblioteca *OpenCv* [OpenCv 2018], portanto, utilizou-se uma versão pré compilada (Versão 3.1.0) para para *smartphones* com o *sistema operacional Android*.

O ambiente de execução do experimento consiste em um *smartphone Galaxy S4 - I9505*, *Android 6.0 Lollipop*, com 2 GB de *RAM* e 16 *GB* de armazenamento.

O modo de extração definido no *Tesseract* foi *OEM_DEFAULT*, utilizando os arquivos padrão de dados do idioma Português Brasil e Inglês. Após realizada a configuração, as imagens foram isoladamente submetidas ao extrator. Foram registrados os tempos necessários para a extração. Registrou-se também, os tempos de processamento demandados para as etapas de *Correção de inclinação*.

O parâmetro do algoritmo 2 – que seleciona apenas as linhas com comprimento mínimo entre o ponto de início e fim da linha – foi definido empiricamente para 70% da largura da imagem a ser rotacionada.

4.4. Experimentos e resultados

Esta subseção apresenta os experimentos realizados, bem como seus respectivos resultados. Na subseção 4.4.3, verifica-se a melhor configuração de parâmetros (modo de segmentação) para o *Tesseract* e *Toth*. A subseção 4.4.4 analisa qual o método mais eficaz para imagens rotacionadas. A subseção 4.4.5 analisa os tempos de processamento para cada um dos métodos propostos.

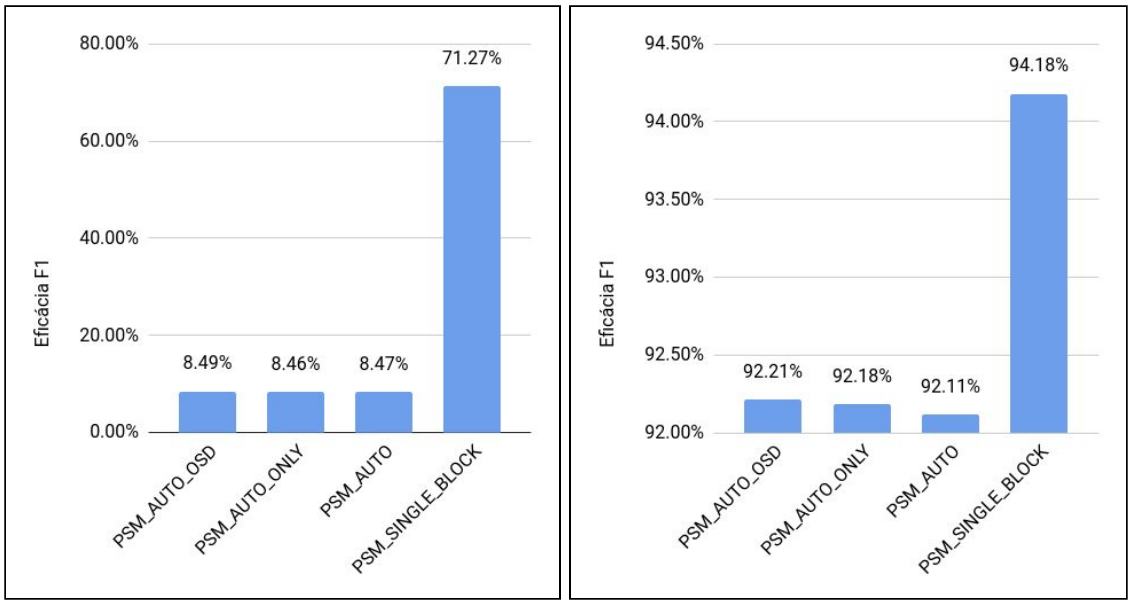
4.4.3. Modo de segmentação de página

O objetivo desse experimento foi responder a seguinte questão: **qual o modo de segmentação de página com maior eficácia?** Para responder essa questão, foram comparados diferentes modos de segmentação de página do *Tesseract*.

Foram comparados quatro modos de segmentações de páginas disponíveis no *Tesseract*: *PSM_AUTO_OSD* - Segmentação automática de páginas com orientação e detecção de linguagem (*OSD*); (ii) *PSM_AUTO_ONLY* - Segmentação de página automática, mas sem *OSD*; (iii) *PSM_AUTO* - Segmentação de página automática mas sem OSD; (iv) *PSM_SINGLE_BLOCK* - Assume um único bloco uniforme de texto.

Foram desconsiderados, por estarem fora do escopo deste trabalho, os demais modos de segmentação de páginas disponíveis no *Tesseract*, por se tratarem de configurações específicas para (i) extração de linhas, termos ou caracteres isolados; (ii) extrações para texto com orientação vertical e extração de elementos textuais dispostos em círculos; e (iii) extrações de textos esparsos ou que não levam em consideração a ordem dos elementos textuais da imagem.

[5] Repositório oficial do projeto Tess-two. Disponível em: <https://github.com/rmtheis/tess-two> Acessado em: 08/07/2018.



(a) Tesseract: BD_ROTACIONA **(b) Toth: BD_ROTACIONA**

Figura 4. Eficácia da extração em relação às características de interesse.

A Figura 4 apresenta comparativos entre os quatro modos de segmentação de página. A Figura 4(a) apresenta a F1 para o *Tesseract* enquanto a Figura 4(b) apresenta a F1 para o *Toth.* Conforme os resultados obtidos, observa-se que o modo de segmentação de página com maior eficácia é o *PSM_SINGLE_BLOCK*. Isto ocorre para os dois métodos. Os próximos experimentos utilizam esse modo de segmentação de página, tanto para o *Tesseract* quanto para o *Toth*, referindo-se a eles apenas como *Tesseract e Toth,* respectivamente.

4.4.4. Método para imagens rotacionadas

O objetivo desse experimento foi responder a seguinte questão: **qual o método com maior eficácia para imagens rotacionadas?** Para responder essa questão, comparou-se o *Tesseract* com a abordagem *Toth*.

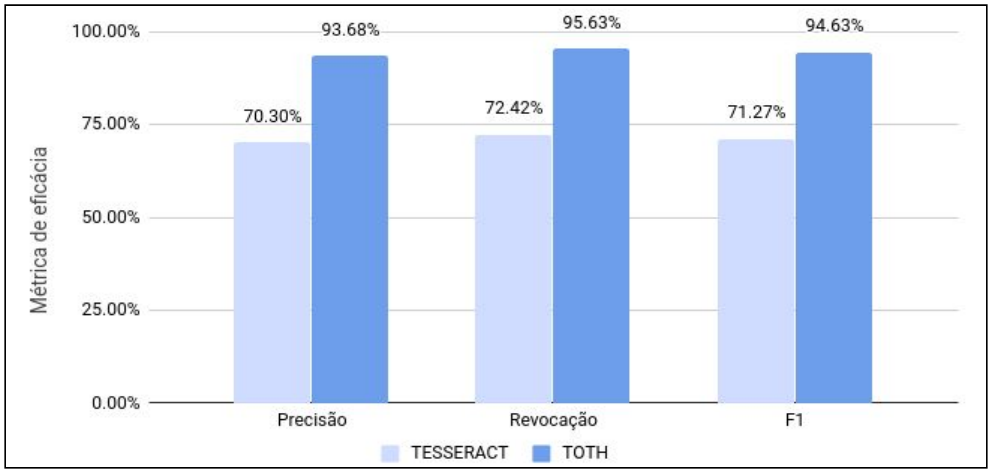


Figura 5. Eficácia da extração para imagens com linhas de texto inclinadas.

A Figura 5 apresenta os resultados de precisão, revocação e F1. Conforme pode ser observado, a abordagem *Toth* obteve maior eficácia, apresentando maior precisão

(93,68%) em relação ao *Tesseract* (70,30%) e maior revocação (95,63%), contra 72,42% obtido pelo *Tesseract*. Dessa forma, a abordagem *Toth* obteve uma F1 de 94,63%, contra 71,27% obtida pelo *Tesseract*. Isso se deve pelo fato de que *Toth* rotaciona as imagens antes de submeter ao motor de extração. O Teste T mostra que esse ganho é estatisticamente significante visto que o valor do p_bicaudal (1,27E-12) é menor que o coeficiente de significância adotado (α = 0,05).

Portanto, conforme os resultados obtidos, é possível constatar que a abordagem *Toth* obtém eficácia significativamente superior ao *Tesseract.*

4.4.5. Tempo

Esse experimento tem por objetivo responder a seguinte questão: **Qual o tempo de processamento para os métodos propostos?** Para responder essa questão, foram registrados os tempos demandados para realizar as extrações.

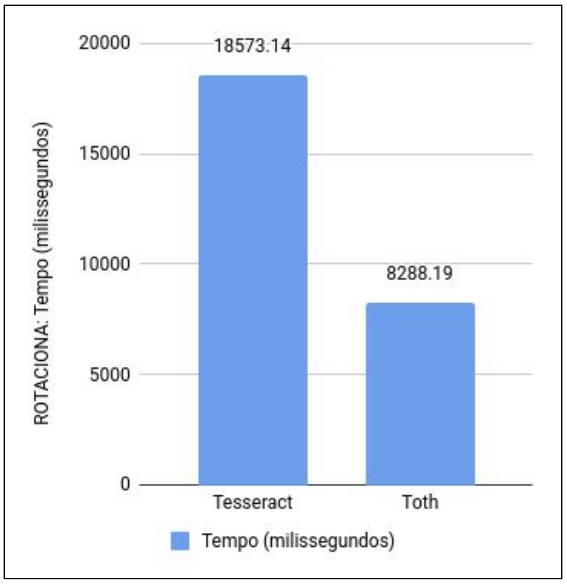


Figura 6. Tempos necessários para a extração demandado pelo Tesseract e Toth.

A Figura 6 apresenta os tempos demandados pelo *Tesseract* e *Toth* para realizar as extrações das imagens. Ressalta-se que para os tempos da abordagem *Toth*, considerou-se o tempo de pré-processamento, acrescido do tempo de extração. Dessa forma, para os tempos demandados pela abordagem *Toth*, soma-se o tempo demandado pelo Algoritmo 2 com o tempo demandado pelo extrator. Conforme os resultados obtidos, observa-se que a abordagem *Toth* realiza a extração em média 2,24 vezes mais rápida. Isso ocorre porque a correção das linhas de texto inclinadas reduz o esforço demandado pelo extrator.

O ganho de eficiência obtido pela abordagem *Toth* é estatisticamente significante, visto que o valor do p_bicaudal (1,08E−09) é menor que o coeficiente de significância adotado (α = 0,05).

4.5. Casos de falha e limitações

Esta seção descreve os principais casos de falhas da abordagem proposta. Essa análise é útil para os desenvolvedores que possam vir a dar continuidade nessa abordagem, bem

como, para aqueles que possam vir a desenvolver novas abordagens para extração de texto em imagens capturadas por *smartphones*.

Do total de 100 imagens da base de dados, 14 extrações realizadas pela abordagem *Toth* obtiveram menor eficácia e, ainda, 17 extrações obtiveram a mesma eficácia apresentada pelo *Tesseract*. Ao analisar empiricamente estas imagens, observou-se que tanto nos 14 casos que obtiveram menor eficácia, quanto nas 17 extrações que obtiveram a mesma eficácia, as imagens apresentavam linhas com poucos graus de inclinação. Já as imagens em que *Toth* obteve maior eficácia, apresentavam maior graus de inclinação.

Outro aspecto a ser observado, diz respeito ao parâmetro do algoritmo 2 – que seleciona apenas as linhas com comprimento mínimo entre o ponto de início e fim da linha – foi definido empiricamente para 70% da largura da imagem. Portanto, o conteúdo textual de imagens capturadas a uma distância maior podem não ser identificados, visto que o comprimento das linhas de texto ficará abaixo do limiar definido.

5. Considerações finais

Este trabalho apresentou uma abordagem para extração de imagens com linhas de texto inclinadas, denominada *Toth*. A abordagem supera o *baseline* em termos de revocação, precisão e F1, realizando as extrações com maior eficácia. Dentre as contribuições realizadas durante o desenvolvimento desse trabalho, destacam-se:

1. Elaboração de uma base de dados com 100 imagens rotacionadas e seus respectivos gabaritos para avaliar métodos de *OCRs* em imagens com linhas de texto inclinadas;
2. Implementação da abordagem *Toth* com etapas para correção de imagens com linhas de texto inclinadas. A abordagem obteve um ganho de eficácia de 23,36 pontos percentuais em relação ao *Tesseract*. Em relação aos tempos de extração, *Toth* viabilizou extrações em média 2,24 vezes mais rápidas.

No decorrer das análises dos resultados obtidos nos experimentos, foram identificadas possíveis ampliações dos experimentos, os quais podem ser desenvolvidos em trabalhos futuros: (i) ampliar o número de características a serem analisadas; (ii) verificar a relação da eficácia da extração por intervalos de graus de inclinação; (iii) analisar os casos específicos em que *Toth* obteve menor eficácia; e (iv) ampliar o número de experimentos, avaliando a abordagem em bases de dados maiores.

Referências

E. Manica; C. F. Dorneles; R. Galante. (2017). R-Extractor: a method for data extraction from template-based entity-pages. In *Computer Software and Applications Conference (COMPSAC)*, *IEEE 41st Annual*. IEEE. p. 778-787.

A. Labrinidis, H. V. Jagadish. (2012). Challenges and opportunities with big data, *Proceedings of VLDB Endowment*, v. 5, n.12, pp. 2032-2033.

D. Agrawal, P. Bernstein, E. Bertino, et. al. (2012). *Challenges and Opportunities with Big Data - A community white paper developed by leading researchers across the United States*.

Statistic Brain. (2017). Instagram Company Statistics. Disponível em: brain https://www.statisticbrain.com/instagram-company-statistics. Acessado em: 15 de Janeiro de 2018.

D. Berchmans; S. S. Kumar. (2014). Optical character recognition: An overview and an insight. In *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, Kanyakumari, pp. 1361-1365.

N. Islam; Z. Islam; N. Noor. (2016). A Survey on Optical Character Recognition System. Journal of Information & Communication Technology-JICT Vol. 10 Issue.2.

F. Asad et al. (2016) High Performance OCR for Camera-Captured Blurred Documents with LSTM Networks. In *Document Analysis Systems (DAS)*, 2016 12th IAPR Workshop on. IEEE. p. 7-12.

J. Liang, D. Doermann, and H. Li. (2005). Camera-based analysis of text and documents: a survey. *International Journal on Document Analysis and Recognition (IJDAR)*, v. 7, n. 2-3, pp. 84–104.

D. M. Kuhn; C. R. Cervi; E. Manica (2018). Extração de elementos textuais em imagens capturadas por smartphones: análise da relação entre as características das imagens e a eficácia da extração. *Escola Regional de Banco de Dados (ERBD)*, [S.l.], v. 14, n. 1/2018.

R. C. Gonzalez; R. E. Woods. (2000). Processamento de imagens digitais. Edgard Blucher.

C. Olah. (2015). Understanding LSTM. Disponível em: http://colah.github.io/posts/2015-08-Understanding-LSTMs/. Acesso em: novembro de 2017.

T. Kil; W. Seo; H. I. Koo and N. I. Cho. (2017). Robust Document Image Dewarping Method Using Text-Lines and Line Segments. *14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Kyoto, 2017, pp. 865-870.

R. W. Smith. (2017). The Extraction and Recognition of Text from Multimedia Document Images, PhD Thesis, University of Bristol, November 1987.

Tesseract (2015). Tesseract. Disponível em: https://github.com/tesseract-ocr/tesseract. Acesso em: novembro de 2017.

A. Gabasio. (2013). Comparasion of Optical Character Recognition (OCR) Software. Department of Computer Science, Faculty of Engineering, LTH, Lund University, 2013.

S. Dhiman; A. Singh. Tesseract vs gocr a comparative study. International Journal of Recent Technology and Engineering, v. 2, n. 4, p. 80, 2013.

OpenCv. (2018). OpenCv Disponível em: https://opencv.org. Acessado em: novembro de 2017.

R. Baeza-Yates; B. Ribeiro Neto. (2013). Recuperação de Informação: Conceitos e Tecnologia das Máquinas de Busca. Porto Alegre: Bookman Editora.