Tratamento de vestígios digitais impressos através de adaptações da tecnologia de OCR

Daniel A. Miranda, Leandro L. Pozzebon

Abstract—**It is common for forensic analysts to receive in printed form data that is usually produced, stored and used in digital form. In certain occasions, difficulty in obtaining the data in it's original format and the amount of printed material are enough to motivate research of an automated way to translate the information back from the paper to a digital format.**

This article presents an approach to leveraging OCR technology to automate tasks such as reassembling complex spreadsheets from printed documents. Two experiments carried on in real cases of a Brazilian Federal Police forensics unit are presented which demonstrate the use of free software and commercial-of-the-shelf software, the Python programming language, pattern recognition and image processing algorithms to achieve productivity increase in analyzing financial data.

Resumo—**Não raro peritos criminais, auditores da receita e técnicos de outros órgãos recebem em formato impresso dados que são normalmente produzidos, armazenados e utilizados em formato digital. Em alguns casos a dificuldade de se obter os dados no formato original e a quantidade de material impresso são suficientes para justificar a pesquisa de uma forma automatizada para traduzir a informação do papel de volta para um formato digital.**

Este artigo apresenta uma abordagem para alavancar a tecnologia de OCR para a automatização de tarefas como reconstrução de tabelas complexas a partir de documentos impressos. São apresentados dois experimentos realizados em casos reais de um setor de perícias da Polícia Federal que demonstram o uso de software livre e software comercial, da linguagem de programação Python e de algoritmos de processamento de imagens e reconhecimento de padrões para obter aumento de produtividade na análise de dados financeiros.

Termos de busca— **OCR automatizado, Python, script**

I. Introdução

O número de operações policiais deflagradas no intuito de desbaratar organizações criminosas especializadas na prática de crimes contra a Sistema Financeiro, contra a ordem Tributária e contra a Administração Pública cresce a cada dia no Brasil. Diretamente em todos esses crimes e, indiretamente em outros, o objetivo principal dos criminosos é a obtenção de recursos de forma ilícita e a posterior lavagem do dinheiro.

Diante deste cenário, para bem elucidar a materialidade e autoria dos crimes, se faz necessário o rastreamento dos bens e direitos subtraídos de forma ilegal pelos criminosos, principalmente dos valores financeiros. Para isso, o principal exame contábil utilizado pela perícia é o exame de movimentação financeira, onde o vestígio a ser examinado compõe-se, normalmente, da documentação bancária formada por cadastros, extratos e comprovantes das transações financeiras.

É nesse ponto que reside um dos principais problemas enfrentados pelos peritos na execução do exame: a forma de remessa da documentação bancária pelas Instituições Financeiras.

No geral, até o ano 2002, a maior parte dos dados relativos a movimentação financeira de um suposto criminoso era encaminha em meio físico, através de papel. Esse tipo de suporte (papel) requer, dependendo do período e do perfil do suposto criminoso, uma demanda muito grande de tempo e de mão-de-obra para a digitação dos dados contidos na documentação, para posterior análise, fatores escassos no âmbito da Criminalística, resultando em demoras nas confecções dos laudos, muitas vezes prejudiciais ao bom andamento do inquérito policial ou processo judicial.

Essa demora dos exames, aliada ao aumento dos inquéritos policiais relacionados a esses tipos de crimes, fez com que, a partir de então, as autoridades gestionassem junto aos bancos o envio dos dados da movimentação financeira em meio digital. A falta, porém, de uma regulamentação sobre o assunto, resultou no atendimento parcial de tal requisição. Alguns bancos continuam a enviar os dados em papel, outros enviam parte em papel e parte em meio digital e, em muitos casos, a parte encaminhada em meio digital é em formato de imagem ou PDF, dificultando o manejo dos dados em planilhas de cálculo e em bancos de dados.

O Banco Central do Brasil, com base na Lei 9.613/98 (Lei da Lavagem de Dinheiro) [1] expediu a Circular 3.290/2005 [2] e a Carta Circular 2.254/2006 regulamentando junto aos bancos o tipo de informação, o meio de armazenamento e o leiaute de envio de tais dados às autoridades requisitantes, regulamentação que ainda não é cumprida por boa parte das instituições demandadas.

· Manuscrito recebido em 30 de junho de 2008. Este trabalho foi realizado com o apoio do Departamento de Polícia Federal do Brasil

Daniel A. Miranda é Perito Criminal Federal, na especialidade Computação Científica (e-mail: miranda.dam@dpf.gov.br).

Leandro L. Pozzebon é Perito Criminal Federal, na especialidade Contabilidade (e-mail: leandro.llp@dpf.gov.br).

Ambos os autores são lotados Setor Técnico-Científico da Superintendência de Polícia Federal no Rio Grande doSul - SETEC/SR/DPF/RS, Av. Ipiranga 1365, Bairro Azenha, Porto Alegre – RS; Tel.:+55 (51) 3235 9066

Atualmente, Poder Judiciário, Ministério Público, Polícias Judiciárias, Institutos de Criminalística, COAF, Banco Central e a Receita Federal do Brasil, através de Seminários e encontros de trabalho buscam somar esforços a fim de que a remessa dos dados da movimentação financeira seja encaminhada pelos bancos de forma eficaz, eficiente e seguindo um único padrão de leiaute, totalmente integrado ao manejo em bancos de dados e planilhas de cálculo.

Enquanto as autoridades brasileiras buscam um consenso interno em torno do assunto, outro problema reside na documentação bancária de instituições financeiras localizadas no exterior e que não estão sujeitas a legislação brasileira, necessitando, para tal, da utilização da cooperação internacional entre países.

No Brasil, o Departamento de Recuperação de Ativos e Cooperação Jurídica Internacional – DRCI, subordinado ao Ministério da Justiça, é o órgão encarregado de realizar a intermediação entre as demandas brasileiras junto a outros países que mantém tratados de cooperação jurídica.

No tocante ao recebimento de documentação bancária no exterior, o que se observa é uma situação se não igual, pior que a dos bancos brasileiros, somente com a remessa de dados em papel ou em meio digital, mas em formato de imagem ou PDF. Segundo o próprio DRCI, normalmente não há nenhuma legislação externa obrigando as instituições financeiras a seguirem um determinado padrão de leiaute dos dados, estando tudo baseado na boa vontade das partes.

É em meio a esse contexto que o perito se depara com um dilema: executar o trabalho hercúleo com base no que foi enviado (dados em meio papel ou em formato de imagem e PDF) ou aguardar a boa vontade das instituições na remessa dos dados em formato manejável em bancos de dados e planilhas de cálculo.

Uma das soluções para este problema está na construção de ferramentas tecnológicas, desenvolvidas para aplicações em casos específicos como os casos práticos mostrados adiante.

II. Tecnologias Disponíveis

A análise da informação que foi impressa começa com a geração de uma representação adequada da mesma em meio digital, podendo essa representação ser mais simples ou mais complexa dependendo do tipo de análise desejada. Nos casos mais simples, pode ser suficiente digitalizar os documentos e gerar um arquivo de texto simples para cada página digitalizada; no entanto, análises mais complexas podem exigir que seja obtida uma planilha ou mesmo um banco de dados a partir dos documentos digitalizados. Cada uma dessas representações possui um grau de dificuldade inerente para ser construída e precisa de um certo grau de "inteligência" por parte do sistema que a produzirá.

A primeira representação a ser obtida é a imagem dos documentos. Estão disponíveis no mercado scanners de mesa com alimentação automática capazes de digitalizar mais de 50 páginas por minuto. A diferença entre a informação digitada e as imagens geradas pelo scanner é que as imagens não podem ser pesquisadas nem correlacionadas diretamente: é uma representação dos dados inadequada para o tipo de análise desejada. Se a representação adequada puder ser obtida de forma automatizada, obteremos o desejado ganho de produtividade.

Para interpretar as imagens e gerar documentos úteis contendo texto ou planilhas, utiliza-se a tecnologia de OCR, sigla em inglês para Reconhecimento Óptico de Caracteres [3].

III. Comparação dos Aplicativos de OCR

Foram utilizados três aplicativos para OCR na tentativa de obter planilhas ou bancos de dados suficientemente fiéis para a análise: o aplicativo Tesseract versão 2.01 [4], disponível gratuitamente, o software fornecido com o scanner HP ScanJet 8270 e a versão demonstrativa do aplicativo Abbyy FineReader 9.0 [5].

Apesar de o objetivo deste trabalho não ser a avaliação dos aplicativos de OCR disponíveis nem a busca pelo melhor aplicativo, buscou-se comparar uma solução baseada em software livre e uma baseada em um aplicativo comercial.

O sofware fornecido com o scanner possui capacidade de gerar documentos em diversos formatos, incluindo PDFs pesquisáveis, RTF, HTML e TXT, mas apenas a partir de imagens obtidas diretamente do scanner, não permitindo o processamento de imagens a partir de arquivos. Essa limitação motivou a busca por outro software comercial, que pudesse processar arquivos de imagens.

O software livre utilizado foi o aplicativo Tesseract, que se originou na empresa HP e teve seu código aberto em 2005. Não possui interface gráfica. Aceita arquivos de imagem no formato TIFF e gera um arquivo TXT com o texto reconhecido. Existe um procedimento [13] para "treinar" o aplicativo para reconhecer novas fontes ou novos idiomas, no entanto os autores não conseguiram reproduzi-lo com sucesso.

O software comercial utilizado foi o aplicativo FineReader, desenvolvido pela empresa Abbyy, de origem russa. Possui interface gráfica elaborada, aceita diversos formatos de arquivos de imagem, no entanto não possui interface por linha de comando. A versão de demonstração utilizada estava limitada a 15 dias de uso ou 50 páginas convertidas.

A qualidade do algoritmo de reconhecimento de cada aplicativo é difícil de quantificar de forma objetiva, uma vez que o texto original dos documentos não é conhecido. Foram produzidas imagens no formato TIFF de três documentos com texto conhecido, em seguida as imagens foram processadas pelos aplicativos e os resultados foram comparados com o texto esperado para cada um dos documentos.

O primeiro documento foi um curto texto do qual constavam 26 letras e 10 algarismos em três fontes diferentes. O segundo documento foi um texto em português, com formatação simples, sem figuras, sobre o arquiteto brasileiro Oscar Niemeyer. O terceiro documento foi um texto também com formatalção simples, sem figuras, em inglês sobre o arquiteto norte-americano Frank Lloyd Wright.

Programa	**FineReader**	**Tesseract**
Formatos de arquivos aceitos	Jpg, gif, tiff, bmp e outros.	tiff
Formatos de arquivos gerados	pdf, pdf/a, html, doc/docx, rtf, xls/xlsx, ppt, dbf, csv, txt, lit.	txt
Treinamento do algoritmo	Sim	Sim *
Interface gráfica	Sim	Não
Interface em linha de comando	Não	Sim

Tab. 1: Quadro comparativo entre dois aplicativos de OCR utilizados nos estudos de caso. *Os autores não conseguiram realizar com sucesso o procedimento de treinamento.

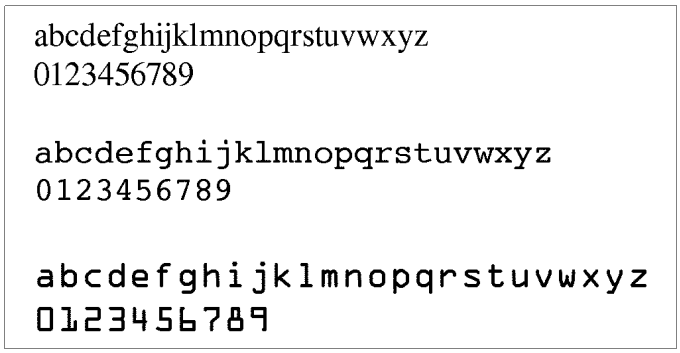


Fig. 1: Primeiro documento, apresentando 26 letras e 10 algarismos em três fontes diferentes.

```
abcdcfghijklmnopqrstuvwxyz
0123456789
abcdefghijklmnopqxstuvwxyz
0 1 2 3 4 5 6 7 8 9
abcdafghijklmnopqrstuvwxyz
CI L E 3 *4 5 E 7 B cl
```

Tex. 1: Texto reconhecido pelo aplicativo **Tesseract** para o primeiro documento.

```
abcdefghijklmnopqrstuvwxyz 0123456789
abcdefghijklmnopqrstuvwxyz 0123456789
abcdefghijklmnopqrstuvwxyz
```

Tex. 2: Texto reconhecido pelo aplicativo **FineReader** para o primeiro documento (os últimos dez algarismos não foram reconhecidos).

Documento	**Alfabeto e algarismos**	**Texto em português**	**Texto em inglês**
Número de caracteres no critério 1	114	23.080	31.163
Número de caracteres no critério 2	108	19.341	26.154
Erro critério 1 Tesseract	30	73 (0,32%)	23 (0,074%)
Erro critério 2 Tesseract	12	73 (0,38%)	23 (0,088%)
Erro critério 1 FineReader	12*	33 (0,14%)	8 (0,026%)
Erro critério 2 FineReader	10*	27 (0,12%)	3 (0.0096%)

Tab. 2: Comparação do erro no reconhecimento de documentos para os aplicativos Tesseract e FineReader utilizando dois critérios diferentes. *A ausência dos dez últimos algarismos implica um erro de, no mínimo, 10 edições.

No critério 1, os documentos são comparados sem levar em consideração as tabulações, quebras de linha e espaços múltiplos. Tabulações e quebras de linha são transformadas em espaços e, por fim, todos os espaços múltiplos são transformados em espaços simples. O número de erros é o número de edições necessárias (substituições, inserções ou exclusões) para tranformar um texto no outro (distância de Levenshtein [14]).

O critério 2 é idêntico ao critério 1 exceto por desconsiderar todos os espaços, não apenas os duplos.

Os documentos acima foram produzidos digitalmente, e não apresentam manchas, distorções ou outros defeitos que possam prejudicar o reconhecimento. Nos documentos reais, a diferença entre o número de erros dos dois algoritmos é maior, favorecendo o aplicativo FineReader, no entanto não foi feita a contagem manual dos erros para fornecer um resultado numérico.

IV. Automação da Inteligência

Os softwares de OCR testados demonstraram capacidade para gerar texto simples a partir de imagens, no entanto demonstraram inabilidade ou sérias deficiências na organização automática dos textos em planilhas, como relatado nos estudos de caso no item IV.

Utilizando apenas os recursos disponíveis comercialmente foi possível evitar a digitação do texto, mas o procedimento envolvia a conversão das imagens dos documentos em “PDFs pesquisáveis”[6] ou outros arquivos que exigiam a cópia manual do conteúdo das células e a organização, também manual, em uma planilha eletrônica.

Os dois casos analisados representam paradigmas diferentes para a análise dos documentos: No primeiro caso os documentos apresentam os dados em formato extremamente irregular, com registros truncados, folhas perfuradas, fotocópias de folhas dobradas, carimbos sobrepostos aos registros, lançamentos a mão e outras dificuldades; já no segundo caso os registros aparecem integralmente e são bem legíveis; o desafio passa a ser organizar dados de diferentes naturezas que aparecem intercalados, alguns em forma de planilha e outros não.

A solução encontrada nos dois casos foi automatizar a parte "inteligente" do processo através da construção de pequenos *scripts* (programas) escritos na linguagem Python. Esses *scripts* são específicos para o leiaute de cada documento e são responsáveis por organizar as informações espalhadas pela página em planilhas e verificar a sua consistência. Não foi possível descartar completamente a interferência humana no processo, mas a mesma ficou reduzida a uma revisão na etapa final, em que eram checados os totalizadores, corrigidas falhas de reconhecimento de OCR e supridas lacunas que os *scripts* não foram capazes de preencher.

V. Estudos de Caso

Foram analisados dois casos reais em que instituições bancárias do exterior forneceram dados impressos em papel referentes a quebra de sigilo bancário.

A. Israel Discount Bank (New York Branch)

O material analisado consistia de cerca de 1700 páginas contendo fotocópias de dados de transações bancárias em contas do Israel Discount Bank. As informações estavam divididas em onze seções terminadas por totalizadores informando quantas transações aquela seção continha e a quantidade de dinheiro movimentada nessas transações. Cada registro estava separado dos demais por uma linha com asteriscos.

Após testes preliminares com os três aplicativos de OCR, verificou-se que o material era complexo demais para qualquer deles, e o leiaute afetava a precisão do reconhecimento. Optou-se por dividir os documentos em pequenas partes e foi feito um *script* que recortava a página em pequenos retângulos contendo o conteúdo de cada célula, apresentando-os individualmente ao OCR.

A base do *script* é o algoritmo de rotulação [7], utilizado na área de processamento de imagens, e consiste em numerar todas as regiões brancas contíguas de uma imagem binária (uma imagem preto-e-branca, sem tons de cinza). A imagem foi transformada em negativo e rotulada. As molduras dos registros foram identificadas através das regiões rotuladas que possuíam as dimensões corretas. Como havia várias células na área perfurada e várias em que a borda esquerda foi omitida, foi necessário que o *script* completasse a borda esquerda e a inferior de cada um dos registros, para que cada célula ficasse contida em um retângulo fechado. Uma vez que cada célula ficou em um retângulo fechado, o algoritmo de rotulação foi aplicado novamente na imagem original binarizada, não invertida, de forma que as regiões brancas foram rotuladas. As que possuíam as dimensões corretas e estavam em posições coerentes eram as regiões que continham o texto.

Na primeira vez que os documentos foram escaneado as perfurações apareceram como círculos pretos. Verificou-se que o algoritmo fica mais simples quando a perfuração aparece em branco, da mesma cor da página, portanto os documentos foram escaneados novamente.

Estima-se que tenham sido gastas cerca de 120 horas de trabalho para desenvolver os *scripts*, que totalizam 993 linhas de código. Quando os *scripts* ficaram prontos, o processamento de todo o lote demorou dois dias em uma estação de trabalho Intel Core2 Quad@2.66GHz com 4GB de memória RAM. As páginas foram digitalizadas em 600 DPI.

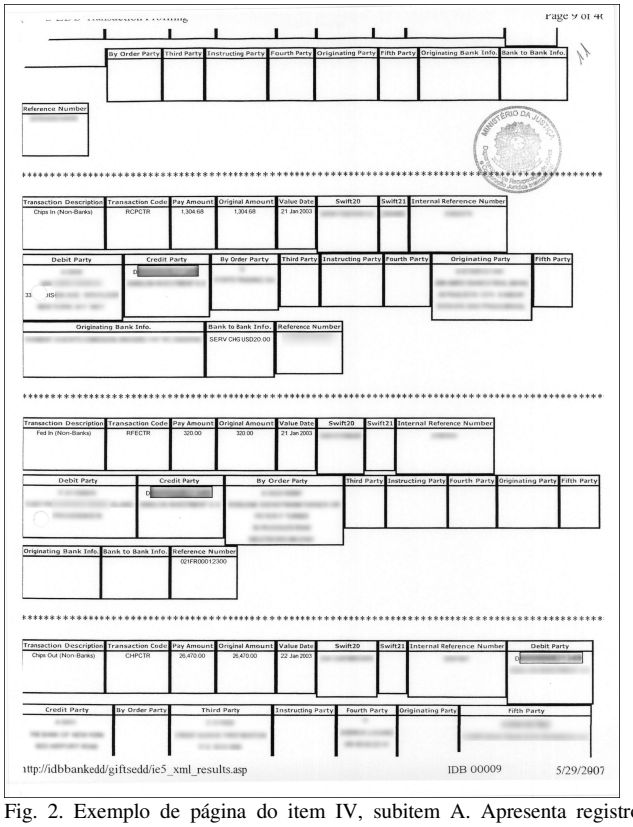


Fig. 2. Exemplo de página do item IV, subitem A. Apresenta registros truncados, carimbo, perfurações da encadernação em duas células sendo uma delas em cima do texto, canto superior esquerdo dobrado. Dados confidenciais foram tornados ilegíveis por motivo de segredo de justiça.

Foi utilizado o sistema operacional Debian GNU/Linux para realizar o desenvolvimento e o processamento, o interpretador Python [8] para executar os *scripts*, as bibliotecas scipy [9], PIL [10] e numpy[11] para a linguagem Python, o kit ImageMagick [12] para fazer a conversão das

imagens e foi utilizado o software Tesseract versão 2.01 [4] como software de OCR. Todos os softwares utilizados estão disponíveis gratuitamente na Internet.

O algoritmo pode ser resumido nas seguintes etapas:

-Alinhamento da imagem

-Identificação das molduras

-Fechamento das molduras

-Identificação dos campos e cabeçalhos

-Recorte e apresentação para o OCR

-Organização da informação, identificação de lacunas e testes de consistência

-Preenchimento das planilhas

Ao fim do processamento, os *scripts* geraram três planilhas no formato ".csv". A primeira continha o nome reconhecido pelo OCR no cabeçalho de cada um dos campos, a segunda continha os valores lidos dentro de cada um dos campos e a terceira continha o nome do arquivo de imagem de onde aquela informação havia sido lida.

Isso permitiu que a consistência dos dados fosse verificada de diversas formas. Entre os critérios utilizados, o primeiro foi verificar se os campos "Pay Amount" e "Original Amount" haviam sido lidos com o mesmo valor, uma vez que os mesmos eram idênticos em quase todos os registros. Outro critério foi verificar se o número de transações reconhecidas pelos *scripts* era o mesmo informado nos totalizadores. Por fim, foi verificado se a soma dos valores das transações era igual ao previsto ao fim de cada uma das 11 seções.

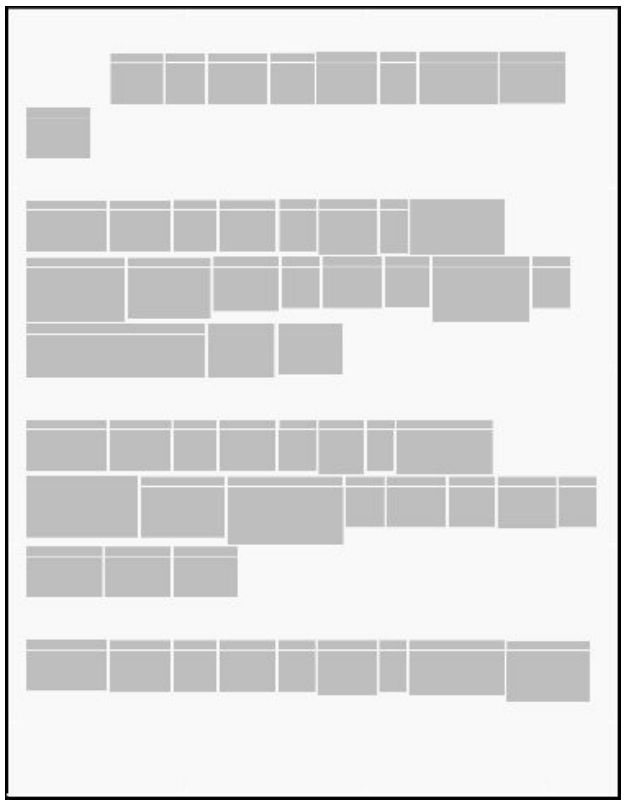


Fig. 3. Retângulos gerados automaticamente mostrando os segmentos que foram apresentados individualmente para o programa de OCR. O *script* desprezou as células com molduras incompletas ao fim da página.

Com essas três planilhas em mãos e os valores dos totalizadores, o perito utilizou um aplicativo de planilha eletrônica para realizar as correções necessárias, principalmente os valores de transações que foram deixados em branco pelos *scripts* devido a células truncadas ao fim das páginas, reconhecidas incorretamente pelo OCR, sobrepostas pelo carimbo, ocultas por dobras ou ilegíveis por motivos diversos. Observou-se mediante contagem manual que o totalizador ao fim da seção 8, que acusava 546 transações, estava errado, e a contagem de 200 detectada pelo *script* estava correta. Desconsiderando esse caso, observou-se que não foram identificados apenas três dos 5395 registros restantes e que 85% dos registros detectados foi lido corretamente. Os valores e transações que o *script* não conseguiu detectar e acusou como inconsistentes compunham 15% do total de transações e do montante movimentado, agilizando muito o trabalho de análise.

Seção	registros	consistentes	soma
1	147	125	2,660,519.44 USD
2	399	346	4,701,142.47 USD
3	530	455	4,980,086.81 USD
4	606	523	12,562,464.49 USD
5	609	513	10,925,639.25 USD
6	827	701	17,361,917.80 USD
7	594	489	12,004,085.06 USD
8	200	173	8,899,622.65 USD
9	596	505	11,141,758.42 USD
10	476	405	7,003,080.93 USD
11	608	529	10,513,993.04 USD
Total	5592	4764	102,754,310.36 USD

Tab. 3. Total de registros detectados pelos *scripts* em cada seção, estimativa de quantos são consistentes e soma das transações consistentes.

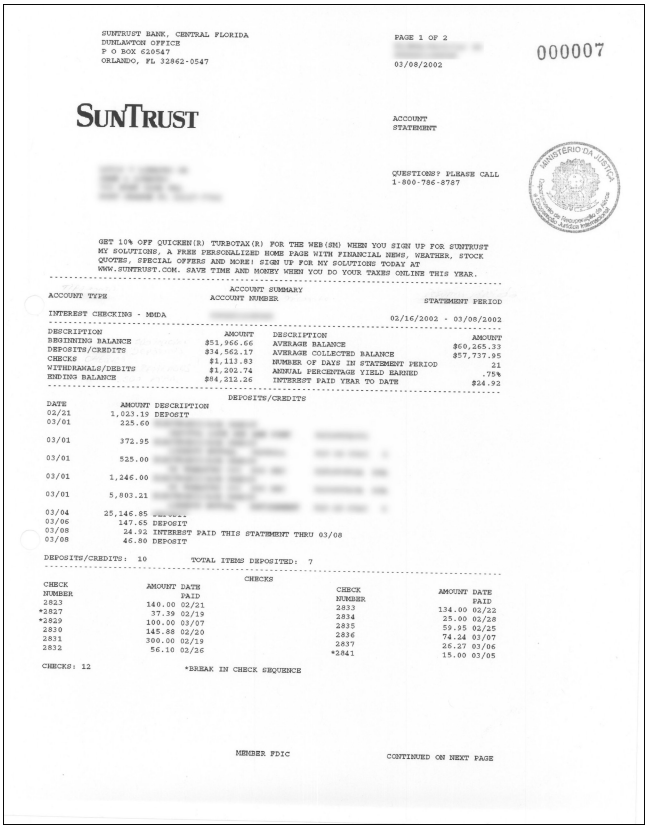
Seção	registros	soma	diferença
1	147	3,111,216.21 USD	450,696.77 USD
2	399	5,209,403.10 USD	508,260.63 USD
3	530	5,765,162.13 USD	785,075.32 USD
4	606	14,410,979.01 USD	1,848,514.52 USD
5	609	13,144,616.13 USD	2,218,976.88 USD
6	827	20,033,280.62 USD	2,671,362.82 USD
7	594	14,411,488.41 USD	2,407,403.35 USD
8	546	11,805,063.59 USD	2,905,440.94 USD
9	599	12,965,564.63 USD	1,823,806.21 USD
10	476	8,043,247.83 USD	1,040,166.90 USD
11	608	11,657,107.34 USD	1,143,114.30 USD
Total	5941	120,557,129.00 USD	17,802,818.64 USD

Tab. 4. Total de registros que constava nos totalizadores, soma que constava nos totalizadores e diferença entre a soma ideal e a soma obtida na tabela 1.

B.SunTrust Banks (Dunlawton Office, Orlando, FL)

O material analisado consistia de 125 páginas contendo extratos bancários de contas do banco SunTrust. As informações estavam organizadas em conjuntos de duas ou três páginas abrangendo períodos de um mês. Os documentos possuíam boa qualidade de impressão e apresentavam leiaute regular e previsível. Apesar de existirem carimbos e perfurações, nenhum desses elementos se sobrepôs ao texto. Normalmente os documentos recebidos possuem qualidade similar a esse caso.

Nesse caso, a abordagem utilizada foi mais simples e não envolveu conhecimentos de processamento de imagem. Para cada página escaneada foram selecionadas manualmente na tela do aplicativo Abbyy Finereader 9.0 apenas duas regiões retangulares para serem processadas: uma contendo o número da página e as datas do extrato e outra contendo os dados do extrato. O aplicativo de OCR produziu arquivos de texto corrido, que foram lidos pelos *scripts* para montar as planilhas.



SUNTRUST BANK, CENTRAL FLORIDA
DUNLAWTON OFFICE
P O BOX 620547
ORLANDO, FL 32862-0547

PAGE 1 OF 2
[redacted]
03/08/2002

000007

SUNTRUST

ACCOUNT
STATEMENT

QUESTIONS? PLEASE CALL
1-800-786-8787

GET 10% OFF QUICKEN(R) TURBOTAX(R) FOR THE WEB(SM) WHEN YOU SIGN UP FOR SUNTRUST MY SOLUTIONS, A FREE PERSONALIZED HOME PAGE WITH FINANCIAL NEWS, WEATHER, STOCK QUOTES, SPECIAL OFFERS AND MORE! SIGN UP FOR MY SOLUTIONS TODAY AT WWW.SUNTRUST.COM. SAVE TIME AND MONEY WHEN YOU DO YOUR TAXES ONLINE THIS YEAR.

ACCOUNT SUMMARY

ACCOUNT TYPE	ACCOUNT NUMBER	STATEMENT PERIOD
INTEREST CHECKING - MMDA	[redacted]	02/16/2002 - 03/08/2002

DESCRIPTION	AMOUNT	DESCRIPTION	AMOUNT
BEGINNING BALANCE	$51,966.66	AVERAGE BALANCE	$60,265.33
DEPOSITS/CREDITS	$34,562.17	AVERAGE COLLECTED BALANCE	$57,737.95
CHECKS	$1,113.83	NUMBER OF DAYS IN STATEMENT PERIOD	21
WITHDRAWALS/DEBITS	$1,202.74	ANNUAL PERCENTAGE YIELD EARNED	.75%
ENDING BALANCE	$84,212.26	INTEREST PAID YEAR TO DATE	$24.92

DEPOSITS/CREDITS

DATE	AMOUNT	DESCRIPTION
02/21	1,023.19	DEPOSIT
03/01	225.60	[redacted]
03/01	372.95	[redacted]
03/01	525.00	[redacted]
03/01	1,246.00	[redacted]
03/01	5,803.21	[redacted]
03/04	25,146.85	[redacted]
03/06	147.65	DEPOSIT
03/08	24.92	INTEREST PAID THIS STATEMENT THRU 03/08
03/08	46.80	DEPOSIT

DEPOSITS/CREDITS: 10 TOTAL ITEMS DEPOSITED: 7

CHECKS

CHECK NUMBER	AMOUNT	DATE PAID	CHECK NUMBER	AMOUNT	DATE PAID
2823	140.00	02/21	2833	134.00	02/22
*2827	37.39	02/19	2834	25.00	03/08
*2829	100.00	03/07	2835	59.95	02/25
2830	145.88	02/20	2836	74.24	03/07
2831	300.00	02/19	2837	26.27	03/06
2832	56.10	02/26	*2841	15.00	03/05

CHECKS: 12 *BREAK IN CHECK SEQUENCE

MEMBER FDIC CONTINUED ON NEXT PAGE

Fig. 4. Exemplo de página do item IV, subitem B. Apresenta carimbo e perfurações, mas os mesmos não se sobrepõem ao texto. Dados confidenciais foram tornados ilegíveis por motivo de segredo de justiça.

Estima-se que tenham sido gastas cerca de 16 horas de trabalho para desenvolver os *scripts*, que totalizam 519 linhas de código. As páginas digitalizadas com 600 DPI.

Foi utilizado o sistema operacional Ubuntu GNU/Linux para realizar o desenvolvimento e o processamento, o interpretador Python para executar os *scripts* e o Microsoft Windows para executar a versão de demonstração do aplicativo Finereader 9.

O algoritmo pode ser resumido nas seguintes etapas:

- O OCR é executado com seleção manual das áreas a serem processadas.

- O primeiro *script* renomeia os arquivos de acordo com a data do extrato e o número da página.

- O segundo *script* junta as páginas de um mesmo extrato.

- O terceiro *script* processa o texto de cada extrato e coloca os valores nas planilhas de depósitos, cheques, retiradas e saldo.

A consistência dos dados foi verificada através dos totalizadores contidos em cada extrato.

Esse material não pôde ser processado na sua integridade devido a limitações na versão de demonstração do aplicativo de OCR, no entanto foi possível aproveitar 100% dos dados das páginas que foram processadas.

VI. Conclusões

Foi demonstrada uma abordagem mais eficiente que a digitação manual para analisar grande quantidade de documentos contábeis utilizando-se ferramentas de OCR e programação. É viável adotar essa abordagem utilizando tanto software livre como software proprietário.

A abordagem adotada neste artigo pode ser adaptada para a utilização em outros tipos de documentos, e é recomendada caso os documentos a serem processados sejam uniformes e de boa qualidade, pois o desenvolvimento é rápido e exige apenas conhecimentos básicos de programação.

Documentos complexos exigem maior esforço para desenvolver os *scripts* de automatização. O investimento de tempo e mão-de-obra especializada é justificável quando há uma quantidade maior de material a ser processado ou forem necessárias poucas adaptações dos *scripts* existentes.

A qualidade dos resultados obtidos neste trabalho pode ser melhorada através da melhoria da qualidade do reconhecimento e da facilidade de uso dos aplicativos livres, bem como da adoção de uma interface em linha de comando para os aplicativos comerciais, de forma a integrá-los melhor com os scripts.

Referências

[1] *Lei nº 9.613, de 3 de Março de 1998* (2008, 30 de junho) Disponível online em: http://www.planalto.gov.br/ccivil/LEIS/L9613.htm

[2] *Circular 3290 de 31 de agosto de 2005* (2008, 30 de junho) Disponível online em: http://www5.bcb.gov.br/normativos/detalhamentocorreio.asp?N=105223255&C=3290&ASS=CIRCULAR+3.290

[3] *Optical character recognition - Wikipedia, the free encyclopedia* (2008, 30 de junho) Disponível online em: http://en.wikipedia.org/wiki/Optical_character_recognition

[4] *tesseract-ocr - Google Code* (2008, 30 de junho) Disponível online em: http://code.google.com/p/tesseract-ocr/

[5] *ABBYY FineReader - Professional OCR Software for Document and PDF Conversion Application.* (2008, 30 de junho) Disponível online em: http://finereader.abbyy.com/

[6] *Acrobat for Legal Professionals: Is that PDF Searchable?* (2008, 30 de junho) Disponível online em: http://blogs.adobe.com/acrolaw/2007/02/is_that_pdf_sea.html

[7] Hanan Samet, *Connected Component Labeling Using Quadtrees*, Journal of the ACM (JACM), v.28 n.3, p.487-501, July 1981

[8] *Python Programming Language -- Official Website* (2008, 30 de junho) Disponível online em http://www.python.org

[9] *Scipy: Scientific Tools for Python* (2008, 30 de junho) Disponível online em http://www.scipy.org/

[10] *Python Imaging Library (PIL)* (2008, 30 de junho) Disponível online em: http://www.pythonware.com/products/pil/

[11] *Numpy* (2008, 30 de junho) Disponível online em: http://numpy.scipy.org

[12] *ImageMagick: Convert, Edit, and Compose Images* (2008, 30 de junho) Disponível online em: http://www.imagemagick.org

[13] *Training Tesseract* (2008, 17 de agosto) Disponível online em: http://code.google.com/p/tesseract-ocr/wiki/TrainingTesseract

[14] В.И. Левенштейн (1965) Двоичные коды с исправлением выпадений, вставок и замещений символов. Доклады Академий Наук СССР 163.4:845–848. Appeared in English as: V. I. Levenshtein, Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10 (1966):707–710.