

Um Sistema Android para Reconhecimento Automático do Número de Cartão de Crédito usando Template Matching

Lincoln S. Rodrigues Júnior¹, Karin S. Komati¹

¹Coordenação de Informática – Instituto Federal do Espírito Santo (Ifes) Campus Serra
Rodovia ES-010 – Km 6,5 – Manguinhos – CEP: 91.501-970 – Serra – ES – Brazil

lincollnjr1994@gmail.com, kkomati@ifes.edu.br

Abstract. *There was an increase in purchases via credit card via smartphones with the growth of electronic commerce. It becomes more convenient and faster to just scan the image of the credit card than to enter the data of the card. In this work, a mobile system is presented that digitizes the credit card number and generates the recognized digits. The system, when scanning the data printed on the face of the credit card, applies digital image processing techniques for image enhancement and performs image digit recognition through the Template Matching technique. Using a test database containing 50 different credit card templates, a 75% hit rate was achieved on recognition.*

Resumo. *Com o crescimento dos comércio eletrônico, ocorreu um aumento nas compras via cartão de crédito via smartphones. Se torna mais cômodo e mais rápido apenas digitalizar a imagem da face do cartão do que digitar os dados do cartão. Neste trabalho, apresenta-se um sistema mobile que digitaliza o número do cartão de crédito e gera os dígitos reconhecidos. O sistema, ao digitalizar os dados impressos na face do cartão de crédito, aplica técnicas de processamento digital de imagens para melhoraria da imagem e realiza o reconhecimento dos dígitos imagem, através da técnica Template Matching. Utilizando uma base de testes contendo 50 modelos de cartões de crédito diferentes, atingiu-se um índice de acerto de 75% no reconhecimento.*

1. Introdução

Com a acessibilidade de tecnologias para acesso à internet sem precisar estar conectado a um cabo, as compras em comércios eletrônicos através de dispositivos móveis aumentaram significativamente. Na pesquisa feita pelo E-Bit, WebShoppers, referente ao primeiro semestre de 2015 [E-BIT, 2015], mostra que o volume de acesso via dispositivos móveis continua aumentando e estima que mais de 37% dos acessos aos maiores sites de comércio eletrônico já são originados por *smartphones* ou *tablets*; e que, entre os e-consumidores que declararam possuir por *smartphones* ou *tablet*, quase a metade fez pelo menos uma compra por meio desses dispositivos nos últimos seis meses.

As compras em comércios eletrônicos podem ser pagas através de boletos bancários, depósitos, cartões de crédito e outras formas de pagamento. Segundo a empresa brasileira de soluções para pagamentos, Moip, no período de 2013 a 2014, 73,4% das compras online utilizaram o cartão de crédito para o pagamento, tendo um crescimento de 3% em relação ao ano anterior da pesquisa [EcommerceNews, 2015]. Se

forem contabilizados todos os pagamentos das compras *online* mais os pagamentos nas lojas físicas, os pagamentos por cartão de crédito no ano de 2013 somaram mais de R\$ 853 bilhões e em 2014, foram mais de R\$ 1 trilhão segundo a Abecs [Felitti; Salgado, 2015].

O pagamento via cartões de crédito passou por uma evolução, desde a versão manual que copiava os dados em papel-carbono, pela máquina específica cuja única funcionalidade é autorizar ou não a compra até as versões que transformam o *smartphone* num terminal de pagamento de cartões. Usando o *smartphone*, pode-se acoplar leitores ou usar sensores do celular (por exemplo NFC - *Near Field Communication*) para fazer a leitura dos dados do cartão de crédito (via Tarja, via Chip e Senha e/ou via Apple Pay).

Este trabalho se foca na versão via APP, que dispensa os acessórios para leitura de cartões. No entanto, é uma opção que demanda a digitação dos dados do cartão de crédito, e alguns usuários consideram esta tarefa muito trabalhosa. Por questões de performance do dispositivo móvel ou pela questão do tamanho da tela, a tarefa se torna cansativa devido à quantidade de rolagens ou de aumento de tela.

Uma solução para melhorar a experiência do cliente em comércio eletrônico adaptado para dispositivos moveis é reduzir a quantidade de formulários, gravando os dados do cliente e os dados dos cartões dos clientes, assim evitando a digitação de dados (como no Apple Pay). No entanto, há muitos riscos quanto à segurança dos dados quando o mesmo é armazenado no banco de dados da empresa. Assim, uma outra proposta é agilizar a digitação dos dados para pagamentos através de cartão de crédito utilizando a câmera do dispositivo móvel para interpretar os dados impressos no cartão e preencher os campos com os dados capturados, sem que exista o armazenamento no banco de dados da empresa.

Todavia, a leitura das informações do cartão de crédito por um sistema automático não é uma tarefa simples. Veja alguns exemplos de imagens de cartões de crédito apresentadas na Figura 1. Na imagem em a), o último conjunto de 4 dígitos está sobreposto a uma imagem com reflexo, dificultando a leitura, até mesmo para um ser humano. Já na imagem em b), o brasão colorido também dificulta a leitura do segundo conjunto de 4 dígitos do número do cartão. Em c), o fundo do cartão possui informações textuais que atrapalham a leitura e em d), as cores similares do fundo do cartão e a dos dados exigem que o leitor consiga reconhecer o alto-relevo das informações.

A proposta deste trabalho é desenvolver um sistema *mobile* de reconhecimento do número do cartão de crédito através de imagens capturadas pela câmera do *smartphones* usando a abordagem *Template Matching*. Das várias plataformas *mobile*, optou-se pelo Android por ser a mais utilizada no mundo, segundo dados do IDC (*International Data Corporation*), o Android está presente em quase 85% dos *smartphones*, em mais de 190 países [Florenzano, 2015].



Figura 1. Exemplos de cartões que apresentam dificuldades na leitura.

2. Trabalhos Correlatos

Nesta sessão serão apresentados os trabalhos correlatos com o sistema desenvolvido nesse trabalho.

Desenvolvido pela empresa JUMIO, o BAN CHECKOUT é um sistema para Android e IOS que realiza a leitura da face do cartão de crédito e gera um arquivo com os dados para agilizar o preenchimento de formulários [JUMIO, 2016]. O objetivo do BAN CHECKOUT é agilizar o pagamento feito através de dispositivos móveis e reduzir as fraudes relacionadas a compra com cartão de crédito. Segundo a JUMIO, a digitalização dos dados pode aumentar a taxa de conclusão de transações em até 10%, eliminando cerca de 75 teclas e quase 2 minutos de tempo de transação. Outra utilidade da aplicação é evitar erros de entrada na base de cliente.

Além da leitura de cartão de crédito, nos EUA, a ferramenta também compara os dados do cartão de crédito com os da carteira de motorista, diminuindo os casos de estornos relacionados a fraudes. A JUMIO também garante que todas as ações ocorram dentro do dispositivo e não há transmissão ou armazenamento de imagens ou dados. A Figura 2 mostra a tela de cadastro do cartão de crédito para pagamento e cadastro dos dados pessoais do cliente. O usuário tem a opção de digitar através do teclado do dispositivo ou clicar no botão “Scan Data” para que seja digitalizado o cartão de crédito (e a carteira de motorista, caso seja emitida nos EUA). Caso contrário, o usuário terá que digitar apenas seus dados pessoais. O processo de extração dos dados através da câmera até a tela com os dados preenchidos dura cerca de 10 segundos. Se os mesmos fossem digitados o tempo seria em média 20 vezes maior.

O Card.io é uma ferramenta SDK (em português, Kit de Desenvolvimento de Sistema) disponível para várias tecnologias, permitindo o reconhecimento do número do cartão de crédito em outros aplicativos terceiros, bastando que seja desenvolvido um módulo para integração com o *software* do Card.io [Gama, 2012].



Figura 2. Telas na seguinte ordem: cadastro de dados, digitalização dos dados do cartão de crédito, digitalização da carteira de motorista e formulário de pagamento preenchido com dados digitalizados. Retirado de [JUMIO, 2016].

Há mais de 160 aplicativos integrados com a solução Card.io, incluindo Venmo, Qthru, Newegg, Spotze, EventDay, Clinkle, TaskRabbit, Ambur, BeagleApp, Floktu, WillCal e outros. No Espírito Santo, um dos aplicativos que apresentam integração com o Card.io é o PicPay [PICPAY, 2016], startup de soluções de pagamento. Segundo a empresa, os dados reconhecidos pelo aplicativo não são salvos, sendo somente usados para a transação.

A Figura 3 apresenta o fluxo de reconhecimento do número do cartão no sistema Card.io. Na imagem à esquerda, o sistema exibe a tela aguardando para iniciar o reconhecimento do número do cartão de crédito, na imagem do meio, o retângulo de cor verde indica que o sistema conseguiu enquadrar o cartão de crédito para começar o reconhecimento dos dígitos. A última parte, apresenta o resultado do reconhecimento dos números do cartão.



Figura 3. Telas de digitalização dos dados do cartão do Card.io, retirada de [ENGADGET, 2011].

A nova versão do Safari permite escanear um cartão de crédito com a câmera do dispositivo, em vez de digitar as informações dele. Após escanear o cartão, o navegador retorna número do cartão de crédito para um formulário [GIZMODO, 2014]. A Figura 4 apresenta o fluxo de processos para realizar o reconhecimento dos dígitos do número do cartão de crédito. No entanto, este sistema grava a imagem do cartão, que é um dado vulnerável para usos desonestos.

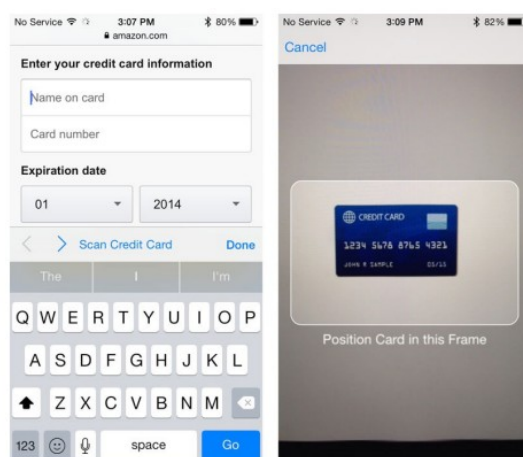


Figura 4. Fluxo de processos de reconhecimento do número do cartão de crédito do Safari do IOS 8, retirada de [MACRUMORS, 2014].

A Unipay é uma start-up brasileira dedicada a pagamentos móveis, que transforma o smartphone de pequenos lojistas em uma máquina de POS (do inglês: *Point of Sale* ou *Point of Service*, em português Ponto de Venda ou Ponto de Serviço, são as máquinas de cartão de crédito), utilizando a câmera do *smarthphone* para realizar a leitura do número, nome e data de validade do cartão de crédito do consumidor de forma automática [Paiva, 2013]. Desta forma, o consumidor precisará apenas do código de verificação, que se localiza na parte de trás do cartão.

3. O Sistema de Reconhecimento de Caracteres

De forma simplificada, um sistema OCR (*Optical Character Recognition*) é uma tecnologia para reconhecer caracteres a partir de uma imagem sejam eles escaneados, escritos a mão, datilografados ou impressos [Singh; Bacchuwar; Bhasin, 2012] [Trier, Jain, Taxt, 1996].

Um OCR pode ser definida em algumas etapas, como mostrada a arquitetura da Figura 5, na qual é exibida a utilização do OCR no problema de reconhecimento de número de cartão de crédito. A imagem de entrada, do cartão de crédito, passa por um processo de extração do ROI (Região de Interesse ou do inglês *Region Of Interest*), depois é feito um pré-processamento no qual a imagem é convertida para escala de cinza e é melhorada quanto à iluminação e ruído. Depois cada caracter do número é recortado ou segmentado para ser avaliado por uma técnica de reconhecimento de caracteres, chegando finalmente aos dígitos do cartão.

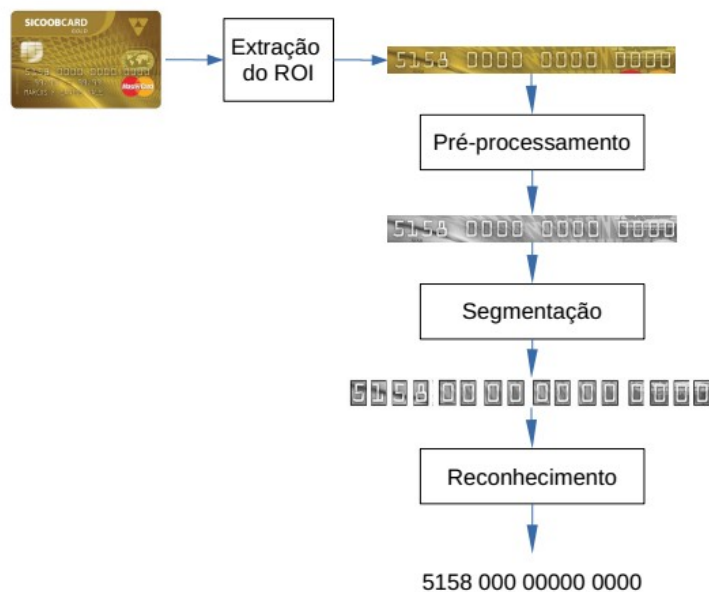


Figura 5. Arquitetura simplificada do sistema OCR.

3.1. Extração do ROI

O ROI é a região definida automaticamente a partir de parâmetros obtidos na própria imagem (ou por um usuário) onde o processamento estará concentrado. No caso deste trabalho, é a parte da imagem que possui os números do cartão de crédito. Tira-se proveito de que a imagem é capturada em tempo real, e que o usuário poderá posicionar o cartão onde o sistema indica através de um retângulo, tal como nos sistemas dos trabalhos correlatos, e portanto não é necessária a etapa de tratamento de correção de rotação, como é feito por Berclaz e colegas (2010).

O cartão de crédito é um cartão de plástico que pode conter ou não um chip e apresenta na frente, no mínimo, o nome do portador, número do cartão e data de validade e, no verso, um campo para assinatura do cliente, o número de segurança (CVV) e a tarja magnética (geralmente preta). Os elementos a serem detectados no cartão de crédito precisam seguir um padrão definido pela ISO 7816-1 ff, que tem a finalidade de regulamentar as dimensões de cartões inteligentes, cartões com chip e tarja magnética [Gorferay Card Service Co., 2011].

Analisando somente o a parte da frente do cartão de crédito, o padrão estabelece que a região referente ao número do cartão de crédito deve ficar a 0,25 milímetro da borda direita do cartão e a 5,2 milímetros da borda inferior. A segunda região que contém outras informações como o nome, data de validade e endereço do cliente está a 0,18 milímetro da borda esquerda e acima 0,03 milímetro da borda inferior. A Figura 6 apresenta o posicionamento dessas duas regiões. Apesar da região 1 (*region 1*) poder apresentar até 19 (dezenove) dígitos no número do cartão, para este trabalho somente será analisado cartões que apresentam 16 (dezesesseis) dígitos em seu número.

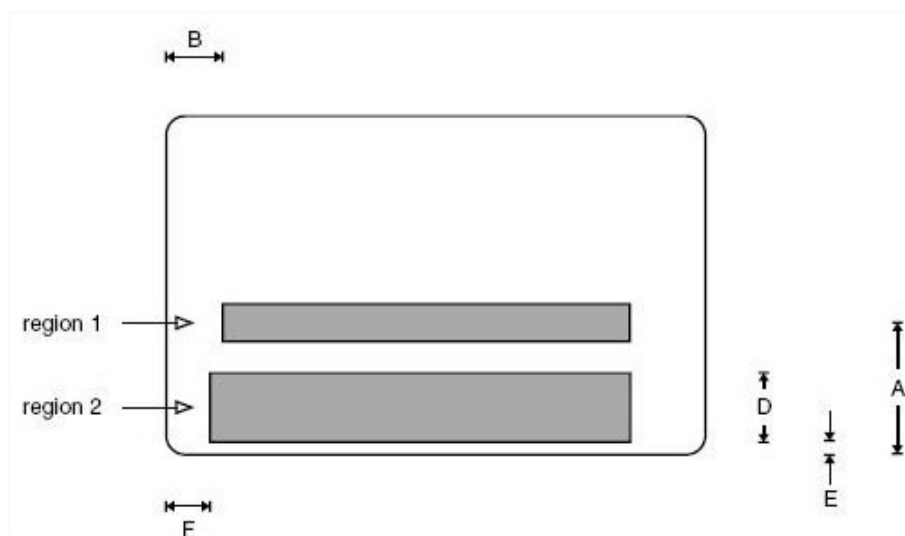


Figura 6. Regulamentação sobre posição das regiões da parte da frente do cartão conforme a ISO 7816-1 ff. Valores: $A = 0,52\text{mm}$, $B = 0,25\text{mm}$, $D = 14,53\text{mm}$, $E = 0,03\text{mm}$ e $F = 0,18\text{mm}$, retirado de [Gorferay Card Service Co., 2011].

3.2. Pré-processamento

Por Processamento Digital de Imagens (PDI) entende-se a manipulação de uma imagem por computador de modo que a entrada e a saída do processo sejam imagens. Utilizado em vários tipos de aplicações, desde correções de imagens de satélites, edições de fotografias e até reconhecimento de caracteres, PDI oferece uma série ampla de algoritmos para sistemas de diversas áreas [Filho; Neto, 1999].

A etapa de pré-processamento visa melhorar a qualidade da imagem, podendo alterar a iluminação, retirada de ruídos, alterações de cor, entre outras técnicas de PDI. Neste trabalho, utilizamos técnicas clássicas, bem consagradas e que já se encontram implementadas e disponíveis em várias bibliotecas de PDI. Inicialmente, aplica-se a conversão de imagem colorida para imagem em escala de cinza, pois, o sistema de reconhecimento usa apenas a forma e não a cor para identificar os dígitos. Na sequência, aplica-se um algoritmo para equalizar o histograma, isto é, corrigir as variações de iluminação da imagem. Por fim, o processo de limiarização converte a imagem em uma imagem em preto e branco, tal como mostrado na Figura 7.

3.3. Segmentação

Segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões (conjunto de pixels) ou objetos, com o objetivo de simplificar e/ou mudar a representação de uma imagem para facilitar a sua análise.

Neste trabalho, o objetivo da segmentação é dividir cada dígito do número do cartão de crédito. Logo após a limiarização, o sistema retira todos os pixels das bordas que não pertencem aos dígitos, retirando as linhas na cor preta dos quatro lados da imagem. Este passo é necessário, pois o corte inicial feito para extração do ROI segue o tamanho definido pela ISO 7816-1 ff, que é um pouco maior que o tamanho dos dígitos.

Depois é feito uma divisão em quatro blocos dos grupamentos do número do cartão, e finalmente divide-se cada dígito de cada bloco, originando 16 (dezesesseis) imagens pequenas, conforme Figura 7.

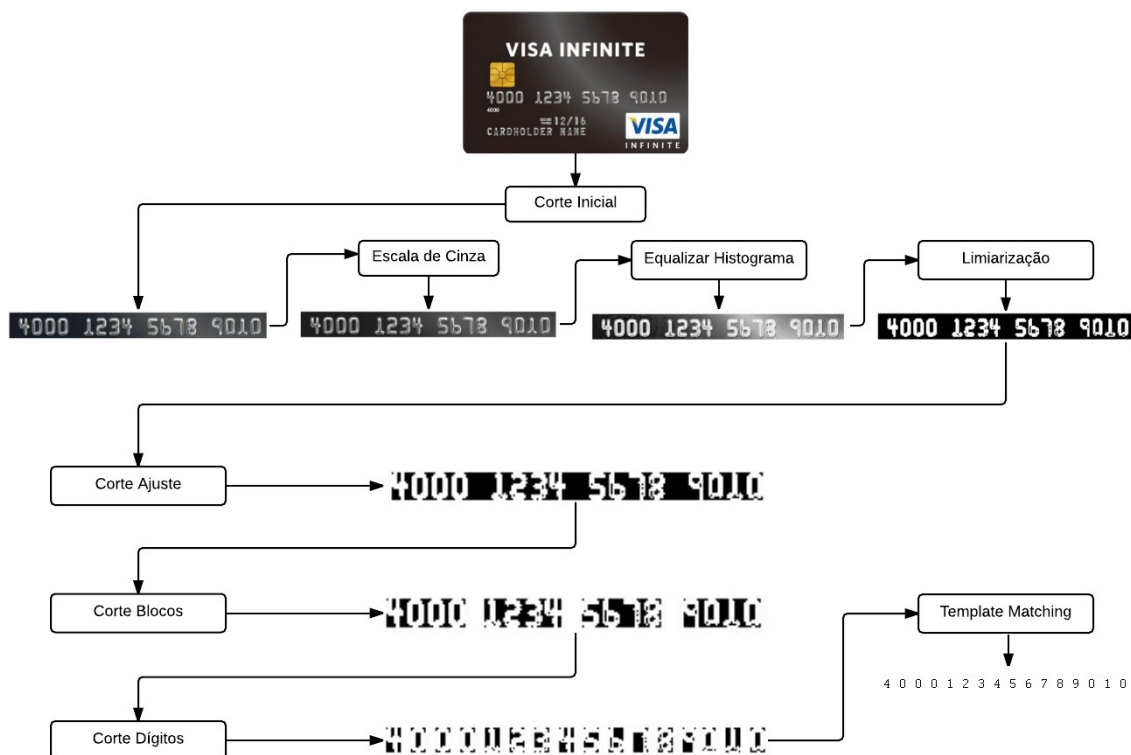


Figura 7. Arquitetura completa do Sistema.

3.4. Reconhecimento

Já a etapa de reconhecimento converte as imagens em texto. A técnica usada neste trabalho é o *Template Matching*, que é uma técnica de PDI que tem como objetivo encontrar pequenas partes de uma imagem que correspondem aproximadamente a uma máscara ou *Template* [Brunelli, 2009].

Essa técnica de reconhecimento de *Templates* basicamente percorre toda a imagem comparando se cada conjunto de pixels tem o mesmo valor que o do *Template*. Quando se encontra um número suficiente aceitável para o *algoritmo* desenvolvido, é dito que foi encontrado um *Match*, ou seja, foi encontrado o *Template* na imagem dada com entrada. A Figura 8 mostra um exemplo num sistema de reconhecimento de placas de trânsito usando esta técnica, numa imagem inicial verificam-se os vários *templates* (trinta no total) correspondentes às várias imagens de placas de trânsito. Ao final, tem-se a indicação da localização de um dos *templates* na imagem de entrada, bem como a identificação de qual foi o *template* usado.



Figura 8. Exemplo de uso de Template Matching, retirado de [Ornellas; Komati, 2013].

Neste trabalho, os dígitos já estão segmentados, e o esforço do reconhecimento é bem menor, pois basicamente não é necessário se fazer uma varredura da imagem, e sim, uma comparação via template de cada dígito com os 16 (dezesseis) recortes.

4. Materiais e Métodos

O sistema CardTess é um aplicativo *mobile* para Android que reconhece os dígitos do número do cartão de crédito. Para o desenvolvimento do aplicativo, foi utilizado o Android Studio, versão 2.1.

O Android é um conjunto de softwares para dispositivos móveis que inclui um sistema operacional, *middleware* e aplicações [Caelum, 2010]. Os desenvolvedores para esse sistema operacional conseguem ter acesso completo as mesmas API's que os aplicativos principais, isso se deve a sua arquitetura ser projetada para reutilização de componente, sendo assim, qualquer aplicativo publicado pode usar outro aplicativo, sujeito as restrições de segurança do aplicativo utilizado. Um exemplo é uma aplicação de agenda que guarda o endereço da pessoa. Essa aplicação pode acessar um aplicativo de mapas passando como parâmetro os dados para buscar o endereço. Para utilizar componentes do dispositivo móvel, basta solicitar acesso ao componente e sobrescrever os métodos do componente. Para utilizar a câmera captura de imagens basta solicitar permissão para acessar a câmera e sobrescrever os métodos de captura.

O OPENCV é uma biblioteca de PDI de código aberto, que fornece uma estrutura de simples utilização para desenvolvedores para construção de sofisticadas aplicações de forma fácil [Bradski; Kaehler, 2008]. A biblioteca foi desenvolvida na linguagem C e C++, e está disponível para Windows, Linux e Mac. O OPENCV também oferece interfaces para várias outras linguagens, uma delas é a linguagem Java no sistema Android.

A linguagem de programação utilizada para desenvolvimento de aplicativos é o Java. Java é uma linguagem de programação orientada a objeto, que foi desenvolvida pela Sun Microsystems na década de 90, e atualmente, pertence a Oracle. A característica mais marcante dessa linguagem é que programas criados nela não são compilados em código nativo da plataforma. Programas em Java são compilados para um *bytecode*, que é executado por uma máquina virtual, o que permite aos desenvolvedores criarem um programa uma única vez e depois executar este em qualquer uma das plataformas suportadas pela tecnologia. Por ser multiplataforma, a linguagem ganhou espaço em diferentes dispositivos [TECHTUDO, 2014].

No entanto, a plataforma Android não roda *bytecodes* no padrão JVM. O *plugin*

do Android para Eclipse, Android Studio, ou o Android Developer Tools realizam uma conversão do *bytecode* padrão JVM para o formato Android, permitindo que se possa usar o IDE sem nos preocupar com detalhes do *bytecode* [Caelum, 2010].

A Figura 9 mostra as telas do CardTess, mostrando a tela inicial para leitura do cartão, que já possui o formato do cartão e o dado a ser lido, e depois a tela de escaneamento, no qual o usuário aponta para o cartão a ser lido.

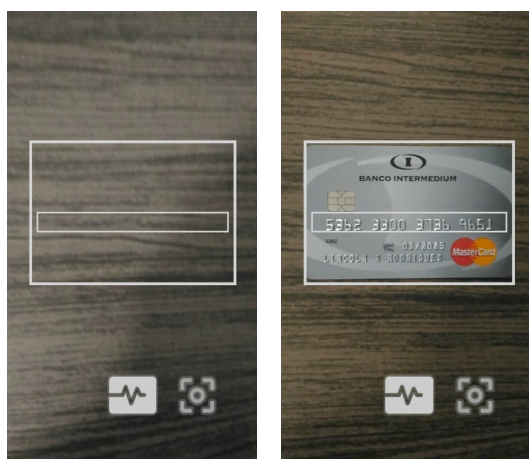


Figura 9. Telas do CardTess.

As imagens de testes foram coletadas em páginas de propaganda, num total de cinquenta imagens, e todas são apresentadas no Anexo I deste artigo.

5. Resultados

Os resultados foram contabilizados como acurácia, tal qual trabalho de Krejcar (2012). A acurácia é dada pela γ_{acc} , e é o percentual dado pela diferença entre todos os símbolos que deveria reconhecer e o número de erros (símbolos reconhecidos incorretamente), como na Equação (1).

$$\gamma_{acc} = \frac{\omega - \omega_{err}}{\omega} 100 \quad [\%]$$

Equação (1)

O resultado final foi de 75% de acurácia. Analizando os erros existentes, verificou-se que o problema não reside na técnica de reconhecimento de caracteres e sim, no resultado após a limiarização. Conforme explicado no início deste artigo, na Figura 1 há vários exemplos que dificultam a leitura dos números, de forma proposital.

Um exemplo que gera erros no reconhecimento é mostrada na Figura 10, com a imagem do cartão e o resultado após a limiarização. Veja que o logo por trás do último bloco de dígitos faz com que o resultado da limiarização gere um resultado que impossibilita o reconhecimento dos dígitos.



Figura 10. Exemplo de resultado com erros de reconhecimento.

O resultado de 75% foi compatível com o resultado do OCR Ocrad em um HTC do trabalho de Krejcar (2012). Usando o Tesseract OCR [Smith, 2007], em seu melhor resultado obteve 26,17% de acurácia em imagens coloridas com as dimensões originais. Tentou-se usar o pré-processamento e as segmentações, mas todas estas manipulações pioram o resultado do Tesseract OCR.

Embora o tempo de processamento não tenha sido medido, o tempo de resposta dos testes foram compatíveis com o sistema BAN CHECKOUT, da empresa JUMIO.

6. Considerações Finais

O cartão de crédito tem se tornado ao longo dos anos o meio de pagamento mais utilizado para compras em comércios eletrônicos, e tem adotado melhorias para facilitar o preenchimento dos dados do cartão, de forma que o usuário não precise digitar todos os dados para finalizar a compra.

O sistema desenvolvido neste trabalho teve como objetivo desenvolver uma aplicação Android a fim de reconhecer o número do cartão de crédito utilizando a técnica Template Matching. Para tal, foi implementada a extração do ROI, o pré-processamento de imagens e várias fases de segmentação.

O CardTess, nome do sistema desenvolvido neste trabalho, apresentou ser eficiente para reconhecer caracteres utilizados, onde obteve sucesso no reconhecimento de 75,00% dos dígitos do número do cartão, dado que teve os ajustes laterais e verticais com sucesso.

Visto os resultados de reconhecimento dos dígitos do número do cartão de crédito no sistema desenvolvido neste trabalho, pode ser possível melhorá-lo para trabalhos futuros, tais como utilizar engines para reconhecimento dos dígitos do número do cartão, utilizar técnicas mais elaboradas no pré-processamento, tal como a equalização de histograma adaptativo ao invés da equalização do histograma e utilizar dados já conhecidos, tais como os dígitos iniciais do cartão que definem a instituição que emitiu o cartão, e possuem poucas possibilidades de dígitos.

Referências

Berclaz, J., Bhatti, N., Simske, S. J., Schettino, J. C. (2010), "Image-Based Mobile Service: Automatic Text Extraction and Translation", Em: Proceedings of SPIE, Vol.

7542, Article Number: 754204.

- Bradski, G.; Kaehler, A. (2008), Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc.
- Brunelli, R. (2009), Front Matter. Wiley Online Library.
- Caelum. (2010), "Google Android: uma nova plataforma Java para celulares". Disponível em: <<http://blog.caelum.com.br/google-android-uma-nova-plataforma-java-para-celulares>>. Acessado em: 30/09/2016.
- E-BIT. (2015), "WEBSHOPPERS". Disponível em: <http://img.ebit.com.br/webshoppers/pdf/32_webshoppers.pdf>. Acessado em: 30/09/2016.
- EcommerceNews. (2015), "Cartão de crédito é meio de pagamento usado em 73% das compras no comércio eletrônico". Disponível em: <<http://ecommercenews.com.br/noticias/pesquisas-noticias/cartao-de-credito-e-meio-de-pagamento-usado-em-73-das-compras-no-comercio-eletronico>>. Acessado em: 30/10/2016.
- Felitti, G.; Salgado, R. (2015), "Gigantes da Tecnologia e Startups Avançam para Sistemas de Pagamento Digital". Disponível em: <<http://epocanegocios.globo.com/Informacao/Visao/noticia/2015/01/credito-debito-ou-digital.html>>. Acessado em: 30/10/2016.
- Filho, O. M.; Neto, H. V. (1999), Processamento digital de imagens. Brasport. Disponível em: <<http://www.ogemarques.com/wp-content/uploads/2014/11/pdi99.pdf>>
- Florenzano, C. (2015), "Porque você deveria desenvolver aplicativos Android". Disponível em: <<http://www.cbsi.net.br/2015/11/porque-voce-deveria-desenvolver-aplicativos-android.html>>. Acessado em: 30/10/2016.
- Gama, H. C. F. (2012), "No More Swiping: Card.io lança App, Ferramentas de desenvolvimento que "vêm" seu cartão de crédito". Disponível em: <<https://curiosidadesnainternet.com/2012/01/24/no-more-swiping-card-io-lanca-app-ferramentas-de-desenvolvimento-que-veem-seu-cartao-de-credito/>>. Acessado em: 30/10/2016.
- GIZMODO. (2014), "Safari do iOS 8 detecta seu cartão de crédito com a câmera". Disponível em: <<http://gizmodo.uol.com.br/safari-do-ios-8-detecta-seu-cartao-de-credito-com-a-camera/>>. Acessado em: 30/10/2016.
- Gorferay Card Service Co. (2011) "Types of Cards". Disponível em: <<http://www.gorferay.com/types-of-cards/>>. Acessado em: 30/10/2016.
- JUMIO. (2016), "Learn how Jumio's BAM Checkout and M-commerce can work for you". Disponível em: <<https://www.jumio.com/bam-checkout/how-it-works/>>. Acessado em: 30/10/2016.
- Krejcar, O. (2012), "Smart implementation of text recognition (OCR) for smart mobile devices". Em: INTELLI, The First International Conference on Intelligent Systems and Applications. pág. 19-24.
- Ornellas, T. S.; Komati, K. S. (2013). "Reconhecimento de placas de sinalização de regulamentação por um método versátil de template matching". Em: Anais do Simpósio de Engenharia, Computação e Sistemas (SECS 2013), Vitória.

- Paiva, Fernando. (2013), "Brasileiros criam leitor de cartão de crédito com câmera do celular". Disponível em: <<http://www.mobiletime.com.br/24/05/2013/brasileiros-criam-leitor-de-cartao-de-credito-com-camera-do-celular/342018/news.aspx>>. Acessado em: 30/10/2016.
- PICPAY. (2016). "Envie e receba dinheiro pelo smartphone, para qualquer pessoa". Disponível em: <<https://www.picpay.com/br/>>. Acessado em: 30/10/2016.
- Singh, A.; Bacchuwar, K.; Bhasin, A. (2012), A survey of OCR Applications. Em: *International Journal of Machine Learning and Computing*, IACSIT Press, v. 2, n. 3, p. 314-318.
- Smith, R. (2007). "An overview of the Tesseract OCR engine". In: IEEE International Conference on Document Analysis and Recognition, p. 629–633.
- TECHTUDO. (2014), "Java: Entenda para que serve o software e os problemas da sua ausência". Disponível em: <<http://www.techtudo.com.br/dicas-e-utoriais/noticia/2014/11/java-entenda-para-que-serve-o-software-e-os-problemas-da-sua-ausencia.html>>. Acessado em: 30/10/2016.
- Trier, Ø. D.; Jain, A. K.; Taxt, T. (1996). Feature extraction methods for character recognition-a survey. Em: *Pattern recognition*, Elsevier, v. 29, n. 4, p. 641–662.

Anexo I





Cartão 21



Cartão 22



Cartão 31



Cartão 32



Cartão 23



Cartão 24



Cartão 33



Cartão 34



Cartão 25



Cartão 26



Cartão 35



Cartão 36



Cartão 27



Cartão 28



Cartão 37



Cartão 38



Cartão 29



Cartão 30



Cartão 39



Cartão 40



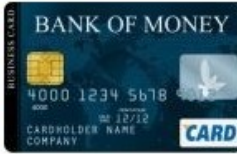
Cartão 41



Cartão 42



Cartão 47



Cartão 48



Cartão 43



Cartão 44



Cartão 49



Cartão 50



Cartão 45



Cartão 46