

Identificador de Caracteres em Carteiras de Identidades

Sistema identificador de caracteres para o auxílio de cadastros

Thiago Gomes de Sousa Bezerra
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
thiagotnd@hotmail.com

Diogo Gomes de Sousa Bezerra
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
diogogsb@hotmail.com

Resumo - Através de uma raspberry pi 3 e uma câmera, desenvolver um sistema capaz de detectar caracteres.

Palavras chave – Raspberry pi 3, detectar.

I. JUSTIFICATIVA

Na sociedade contemporânea é comum o cadastro de documentos e de dados pessoais em diversas ocasiões, como na entrada de um hospital por exemplo, no qual para que seja feito o devido atendimento é necessário realizar o cadastro do paciente. Em muitas vezes, como no exemplo citado, o cadastro é realizado de maneira manual, onde os dados do paciente são transcritos para uma ficha técnica. Esse procedimento na maioria das vezes ocorre de maneira lenta, no qual ocasiona um certo congestionamento. Percebe-se que é possível citar vários outros locais que sofrem do mesmo problema, como em banco por exemplo.

Hoje no Brasil existe uma lei de municípios que limita o tempo máximo de espera em filas para atendimento, no qual tal limite não pode ultrapassar o tempo máximo de 15 min, ou até 30 min em horário de pico, para os bancos. Estudos mostram que infelizmente este tempo não é respeitado na maioria das cidades analisadas. [1]

Visando minimizar e otimizar o tempo gasto para os variados cadastros, observou-se a necessidade de desenvolvimento de um dispositivo capaz de realizar, de forma automática, o cadastramento de documentos pessoais.

II. OBJETIVO

Desenvolver um sistema capaz de detectar os principais dados (nome, RG e CPF) de uma carteira nacional de identidade, através de processamento de imagem, a fim de minimizar e otimizar o tempo gasto para o cadastro de documentos e dados pessoais.

III. BENEFÍCIOS E REQUISITOS

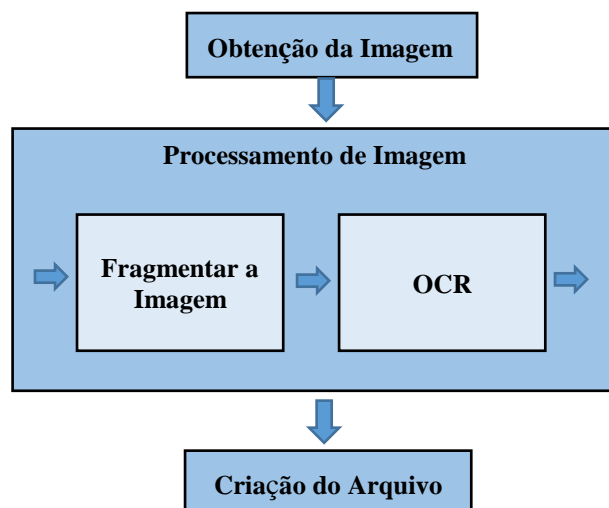
A partir das situações apresentadas, propõem-se desenvolver um sistema capaz de identificar em uma

carteira de identidade (por meio de uma câmera), dados como nome completo, RG, CPF e data de nascimento de um determinado paciente, além de criar um arquivo texto, que irá conter os dados identificados no processo. Esse sistema será capaz de diminuir o tempo de espera em filas de cadastros e prontuários, assim, tornando possível otimizar esse processo e consequentemente agilizar os procedimentos seguintes.

Com requisitos mínimos necessários para o desenvolvimento desse projeto, temos a captação de fotos por meio do webcam e da raspberry pi 3, no qual realizará o processamento de imagem com a utilização da biblioteca opencv, que nos dá a possibilidade de realizar o OCR ou reconhecimento óptico de caracteres. Assim, pode-se transformar os dados obtidos pela câmera em um arquivo no formato .txt, sendo eles salvos na própria raspberry pi 3, através de um cartão de memória.

IV. METODOLOGIA

O fluxograma do projeto proposto está presente abaixo:



- Obtenção da Imagem:

Através da WebCam presente na raspberry pi, as carteiras de identidade serão fotografadas e arquivadas.

- **Processamento de Imagens:**

Em posse da imagem (foto da identidade) obtida, fragmentar a imagem em regiões, que são locais específicos que contém dados a serem utilizados, tais como nome, data de nascimento, nomes dos pais, RG e CPF. Após a “fragmentação”, utilizar o OCR a partir do tesseract e o software criado utilizando o OpenCV, afim de obter-se tais dados na forma de strings.

- **Criar arquivo para o cadastro:**

Com as strings obtidas, criar um arquivo no formato nome.txt, o qual irá conter todos os dados obtidos durante o processamento de imagens.

V. DESCRIÇÃO DE HARDWARE

O hardware utilizado para o desenvolvimento do projeto pode ser observado na tabela 1, assim como os devidos preços. Observa-se que em termos de componentes, o projeto se apresenta bastante simplificado.

	Componente	Fabricante	Preço
01	Raspberry pi 3B	Raspberry	\$230
01	Web Cam	Sony Playstation Eye	\$50
01	Teclado/mouse		\$50
01	Cartão de memória 8gb	SanDisk	\$20
01	Display/Monitor		\$100
01	Fonte de alimentação 3.1A 5v.	Inova	\$15
01	Conversor HDMI-VGA		\$10
TOTAL			\$425,00

Tabela 1 – Componentes utilizados.

A câmera, junto ao teclado e mouse, será conectada por meio das portas USB da Raspberry pi 3. A raspberry será conectada a um monitor através de um cabo HDMI. A sua alimentação será dada por meio de uma fonte de alimentação de 5V e 3.1A.

A fim de otimizar a captura de dados, foi desenvolvido uma estrutura para fixar o documento analisado. Tal estrutura foi construída com medidas fixas, assim o processo de aquisição de dados é realizado de forma igual para qualquer usuário. Na figura 1 pode ser observada o protótipo desenvolvido para a fixação do documento.



Figura 1 – Protótipo desenvolvido para a fixação do documento.

VI. DESCRIÇÃO DE SOFTWARE

O foco deste projeto é identificação de caracteres a partir de determinadas imagens, sendo elas obtidas por uma câmera. Para tal tarefa, é necessário a utilização do Tesseract, um software livre de OCR (Optical Character Recognition) ou reconhecimento óptico de caracteres, que é capaz de identificar textos em imagens convertendo-os em strings. Visto que o Tesseract não é capaz de converter com exatidão todas as fontes possíveis, é necessário utilizar-se uma biblioteca capaz de auxiliar no aprimoramento deste tipo de reconhecimento de imagem. A biblioteca utilizada para essa tarefa é chamada OpenCV.

OpenCV é uma biblioteca multiplataforma, totalmente livre ao uso acadêmico e comercial, para o desenvolvimento de aplicativos na área de visão computacional, bastando seguir o modelo de licença da BSD Intel. O OpenCV possui módulos de Processamento de Imagens e Video I/O, Estrutura de dados, álgebra linear, GUI (Interface Gráfica do Usuário) básica com sistema de janelas independentes, controle de mouse e teclado, além de mais de 350 algoritmos de visão computacional como: Filtros de imagem, calibração de câmera, reconhecimento de objetos, análise estrutural e outros. Esta biblioteca foi desenvolvida nas linguagens de programação C/C++. Também, dá suporte a programadores que utilizem Python e Visual Basic e desejam incorporar a biblioteca a seus aplicativos.

O programa desenvolvido no projeto, é constituído praticamente por duas etapas, a aquisição e processamento da imagem, e a organização dos caracteres em formato de documento.

A primeira etapa, consiste na captura de imagem, utilizando o software fswebcam, e o processamento do mesmo, buscando-se uma melhor qualidade a fim de otimizar a conversão de caracteres. Para isto, o primeiro

passo foi converter a imagem em tons de cinza, utilizando a biblioteca “opencv2/core” do openCV. Utilizou-se também a biblioteca “opencv2/imgcodecs” do software, no qual nos permite recortar a imagem em pontos específicos, 5 como já mencionado, sendo eles especificados na própria função. Após a identificação destas regiões, aplicou-se em cada uma, a biblioteca tesseract, retornando 5 arquivos distintos uma para cada região.

Na segunda etapa, a preocupação é criação do arquivo com o nome obtido na etapa passada, além da junção de forma organizada dos caracteres lidos através do OCR nesse arquivo. Para isso, foi necessário criar um programa capaz de realizar praticamente a mesma tarefa que a função *cat* do Linux, onde nos tornou possível ler todas as strings e caracteres presentes em cada arquivo com extensão .txt (cinco ao total, provenientes do OCR), tornando-as em uma única variável tipo *char*, após a obtenção dessas strings, cria-se um arquivo chamado nome.txt a partir da variável nome obtida, em seguida acrescenta-se de maneira organizada as variáveis obtidas, como RG, CPF e também nome.

VII. RESULTADOS

Dado o prosseguimento do projeto, foi possível, a partir da biblioteca fswebcam, tirar fotos e definir o nome da imagem criada. Já com a utilização do tesseract, software utilizado para OCR (Optical Character Recognition, em português reconhecimento óptico de caracteres) foi possível transformar as palavras contidas nas imagens em strings, entretanto, em alguns casos essas transformações não retornaram as strings desejadas.

Realizando testes com identidades de diferentes usuários, percebe-se que o programa consegue, com uma certa precisão, reconhecer os dados presentes nos documentos. Utilizando a imagem capturada pela webcam mostrada na figura 2.1, pode-se ter uma ideia na capacidade do programa em identificar caracteres. O resultado pode ser observado na figura 2.2.

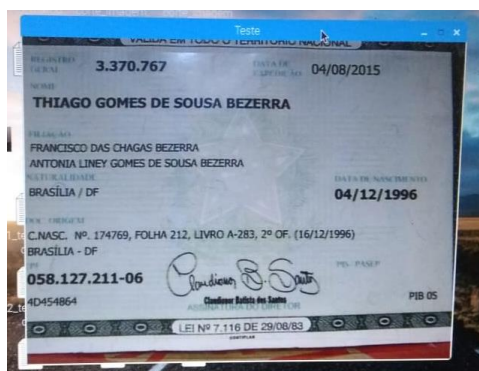


Figura 2.1 – Foto do documento a ser analisado pelo programa.

```
pi@raspberrypi:~/Desktop $ ./corte5
Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica
Warning. Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 212
3.370.767
Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica
Warning. Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 169
THIAGO GOMES DE SOUSA BEZERRA
Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica
Warning. Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 253
FRANCISCO DAS CHAGAS BEZERRA
ANTONIA LINEY GOMES DE SOUSA BEZERRA
04/12/1996
Tesseract Open Source OCR Engine v4.0.0-beta.1 with Leptonica
Warning. Invalid resolution 0 dpi. Using 70 instead.
Estimating resolution as 213
058.127.211-06
```

Figura 2.2 – Teste realizado com o documento de identidade.

Com o reconhecimento de caracteres realizado com êxito, temos em seguida a criação do arquivo nome_usuario.txt que contém os dados obtidos na etapa de reconhecimento. Como o tesseract retorna os caracteres lidos em um arquivo .txt, foi necessário ler todas as strings de todos os arquivos de maneira organizada, afim de obtermos variáveis do tipo *char* para serem usadas tanto na criação do arquivo com o nome do usuário, quanto na escrita desses dados nesse arquivo. A imagem abaixo mostra os resultados obtidos nesse processo.



Figura 3.1: Executando o programa.

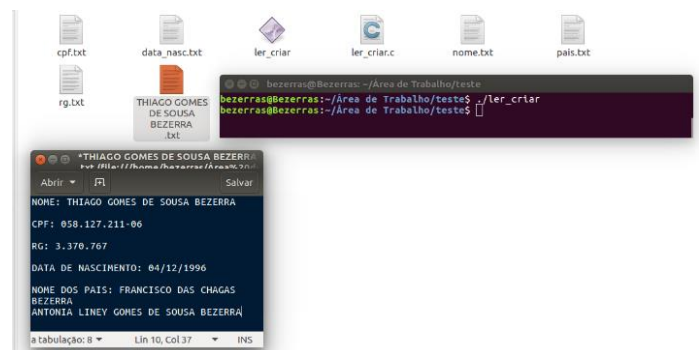


Figura 3.2: Resultado obtido após a execução do programa.

Nota-se que após a execução do programa, foi criado um arquivo texto com o nome lido pelo OCR, o qual contém todos os outros dados lidos pelo mesmo.

VIII. CONCLUSÕES

A partir dos dados obtidos com a implementação do projeto, pode-se verificar que os resultados obtidos foram satisfatórios dentro de suas limitações, visto que o sistema foi capaz de cadastrar, a partir de uma foto da carteira de identidade, os dados de um usuário em um pequeno espaço de tempo.

Uma das limitações encontradas para a utilização desse sistema é a versão da carteira de identidade utilizada, já que modelos mais antigos diferem dos modelos mais novos (utilizados neste projeto) tanto na distribuição dos dados que são necessários para o sistema, quanto na fonte utilizada. Além disso, verificou-se que a qualidade da carteira de identidade é também responsável na obtenção correta dos dados, no qual documentos manchados, dobrados ou danificados de alguma forma, atrapalham na obtenção dos resultados.

IX. ANEXOS

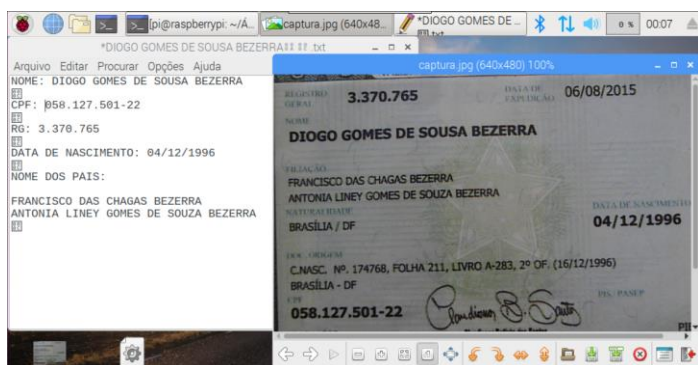


Figura 4 – Exemplo utilizando o projeto desenvolvido em um documento de identidade.

REFERENCIAS BIBLIOGRÁFICAS

1. Jusbrasil. Tempo de espera no banco. Disponível em: <<https://alestrazzi.jusbrasil.com.br/artigos/173949669/tempo-de-espera-no-banco-o-que-fazer-em-caso-de-demora>>
2. Code Experts learning, disponível em: <<https://blog.codeexpertslarning.com.br/lendo-imagens-uma-abordagem-%C3%A0-ocr-com-google-tesseract-e-python-ee8e8009f2ab>>