Reconhecimento de placas veiculares

Acesso ao Cine Drive-in e Comanda Eletrônica

Brenda Medeiros Santos
Faculdade UnB Gama
Gama-DF, Brasil()
brenda.eng.unb@gmail.com

Gabriela Conceição dos Santos
Faculdade UnB Gama
Gama-DF, Brasil
gabrielacsantos.engunb@gmail.com

Resumo – Este relatório apresenta uma proposta de Projeto para a disciplina Sistemas Embarcados. O Projeto consiste na implementação de um sistema de reconhecimento de placas de veículos, usando o Raspberry Pi, para controle de comanda no Cine Drive-in de Brasília.

Palavras-chave – *Raspberry Pi*, reconhecimento de placas, *OpenCV*, *Tesseract*, Cine Drive-in, comanda eletrônica.

I. INTRODUÇÃO

O monitoramento de veículos tem se tornado cada vez mais necessário diante do aumento da frota e do fluxo destes nos centros urbanos. De acordo com a Agência Internacional de Energia, espera-se que em 2035 haja cerca de 1.7 bilhões de automóveis, o que representa aproximadamente o dobro da frota atual [1].

Os avanços tecnológicos têm possibilitado maior controle desse fluxo e monitoramento de veículos, nesse sentido destaca-se a identificação e reconhecimento das placas de carros que podem ter diversas funcionalidades, como: detecção de infratores, carros roubados, estudo de tráfegos e, principalmente, para permitir o acesso de veículos a determinados locais [2].

1.1 **Cine Drive-in**

Inaugurado em agosto de 1973, o Cine Drive-in de Brasília (Figura 1) é o último do país em funcionamento. Possui 15 mil metros quadrados de área asfaltada, capaz de acomodar 400 veículos em seu estacionamento, uma tela de concreto medindo 312 metros quadrados, sendo sua projeção feita com moderno projetor Digital Barco [3].

No Cine Drive-in, a transmissão do áudio do filme para o rádio dos carros é feita com um transmissor de FM. Caso o carro do usuário não possua som, basta acender o farolete de seu veículo e solicitar orientação do atendente [3].

Além disso, para o maior conforto e comodidade de seus usuários, o Cine Drive-in disponibiliza também um atendimento de lanchonete realizado no veículo por um garçom. Desse modo, ao ingressar no cinema o usuário recebe um cardápio com todos os alimentos e bebidas oferecidos e usa o farolete para solicitar o atendente [3].

Figura 1: Cine Drive-in de Brasília.

Sendo assim, perante a necessidade de aperfeiçoamento e automatização do serviço de atendimento prestado pela lanchonete do Cine Drive-in e com o intuito de se evitar eventuais equívocos nos pedidos realizados por seus usuários, propõe-se a construção e implementação de um sistema de reconhecimento de placas veiculares para comandas individualizadas.

Para cada carro será criada uma comanda que será controlada pelo usuário através do aplicativo que deverá ser instalado no celular. Tal aplicativo contabilizará os pedidos e somará os gastos para que o cliente possa efetuar o pagamento na saída, cuja identificação de comanda será feita através da leitura da placa na saída, momento em que a placa será retirada da lista de carros presentes no local.

II. REVISÃO BIBLIOGRÁFICA

2.1 **Sistemas comerciais:**

Aborda-se o cenário atual das tecnologias de reconhecimento de placas de automóveis e de comandas eletrônicas, serão apresentados alguns produtos desses segmentos, que são oferecidos por empresas privadas, com descrição de seu funcionamento e preço, caso essas informações sejam de domínio público.

2.1.1 **Reconhecimento de Placas:**

2.1.1.1 **ALPR da Motorola:** trata-se de um dispositivo móvel que fica acoplado a um servidor situado no interior do veículo, realizando o processamento de dados adquiridos e avisando somente quando uma placa, em específico, é reconhecida. Segundo a Motorola, o sistema está apto para reconhecer as placas mesmo em condições climáticas e temporais adversas. De acordo com dados do fabricante, a implementação do sistema tem variação entre 12.250 a 18.700 dólares nos EUA, esse valor pode variar em virtude da quantidade de câmeras a serem utilizadas [4].

2.1.1.2 **3M:** em termos de requisitos, o dispositivo para a captura de placas da **3M** é bastante semelhante ao da Motorola. Um diferencial desse produto é possuir a tecnologia *Triple Flash*, cuja a função é suprimir os efeitos causados pelos faróis de outros carros ou do pôr do sol na captura da imagem da placa [5].

2.1.1.3 - **Coban:** construído com o intuito de realizar a captura de placas de veículos, a distinção entre o produto desenvolvido pela empresa Coban e os demais apresentados está no fato desse produto poder atualizar seu banco de dados via conexão *wireless* e, além disso, ser capaz de identificar a posição geográfica da placa de interesse. De acordo com o fabricante, a Coban, o seu produto está avaliado a partir de 9.500 dólares [6].

2.1.2. **Comanda Eletrônica:**

2.1.2.1 - **Altec:** a comanda eletrônica foi desenvolvida para ser utilizada, em bares e restaurantes, por garçons por meio de tablets, durante o atendimento aos usuários desses estabelecimentos. Segundo a empresa desenvolvedora, os diferenciais da comanda Altec com relação a outras oferecidas no mercado são: possuir uma interface amigável e de fácil aprendizado, apresentar funcionalidades em uma única tela, navegação por ícones, multi-funções em qualquer etapa do pedido, acessos rápidos às categorias e pratos, grupo de produtos customizáveis: cores e ícones para fácil visualização [7].

2.1.2.2 - **Pocket Cheff:** é um sistema de comanda eletrônica composto por um software completo de gerenciamento de restaurante que proporciona maior agilidade no atendimento, aumento da rotatividade de mesas e redução de falhas em pedidos, melhorando a rentabilidade do estabelecimento [8].

O Pocket Cheff foi desenvolvido para estabelecimentos de todos os tamanhos: de bares, lanchonetes, redes de fast food e restaurantes. Através da solução de comanda eletrônica, o garçom captura os pedidos que são enviados automaticamente para a cozinha por meio de iPods ou terminais touch screen. Este sistema para restaurantes oferece uma visão geral do status dos pedidos, controle diário do caixa e divisão de conta por pessoa [8].

2.1.2.3 - **Atendi:** foi criado para simplificar e agilizar o atendimento em estabelecimentos como bares, restaurantes e lanchonetes. A proposta desse sistema é trazer os melhores recursos tecnológicos em uma nova forma de apresentar e vender produtos. E termos de atendimento, o Atendi se destaca por: gerar comanda eletrônica, realizar o controle de caixa, apresentar o cardápio de forma digital, fazer pedidos através de tablets ou celulares, ou seja, trata-se de um frente de caixa completo com Sistema de Gestão Integrada (ERP) online [9].

2.2 **Ferramentas para o processamento digital de imagens:**

2.2.1 ***OpenCV***

O OpenCV (Open Computer Vision) é uma biblioteca multiplataforma usada no desenvolvimento de aplicativos para as áreas de Processamento de Imagens e Visão Computacional. Dispõe de vários algoritmos, onde pode-se destacar os de: segmentação, reconhecimento de faces, aprendizado de máquinas, filtragem de imagens, dentre outros [10].

A biblioteca detém interfaces com linguagens como C, C++, Python e Java, além de suportar sistemas operacionais Windows, Linux, Mac OS e Android. Além disso, ela dispunha de ferramentas de processamento de imagens e vídeos como filtros de imagem, calibração de câmera, reconhecimentos de objetos e análise estrutural tornando o processo de programação mais factível [10].

2.2.2 ***Tesseract***

O Tesseract foi originalmente desenvolvido na Hewlett-Packard Laboratories Bristol e na Hewlett-Packard Co, Greeley Colorado, entre os anos de 1985 a 1994, com mais algumas mudanças, foi portado para Windows em 1996, além de alguns “C++zing” (upgrades) em 1998. Em 2005 foi liberado a comunidade pela HP e desde 2006 é então desenvolvido pela Google [11].

Tesseract-OCR (Optical Character Recognition) trata-se uma biblioteca de código aberto e seu objetivo é a leitura de textos e caracteres de uma imagem. Ela tem a capacidade de transformar a imagem de um texto em um arquivo .txt do mesmo. Com os avanços no âmbito da programação, essa biblioteca foi modificada, permitindo o funcionamento da mesma em programação Python. Por esse motivo, o nome desta biblioteca para a linguagem Python tornou-se “PyTesseract” [11].

O funcionamento do PyTesseract se deve a diversos arquivos externos, chamados de dicionários. Nestes arquivos há diferentes tipos de fontes de letras e diferentes combinações de palavras. Assim é possível que o programa faça a leitura de qualquer frase ou caractere que se encaixe nesses arquivos [11].

III. REQUISITOS

- Raspberry Pi 3 Model B;
- Câmera ou Webcam;
- Captura de imagem;
- Detecção de placa veicular;

- Reconhecimento de placa veicular;
- Cadastro da placa no sistema;
- Descadastro da placa no sistema mediante efetivação do pagamento;
- Aplicativo com descrição de cardápio para pedidos.

IV. DESENVOLVIMENTO

Diante das informações apresentadas, o objetivo do presente projeto é a identificação da placa do veículo e o armazenamento de seus caracteres para a criação de comandas individualizadas a serem acessadas através de um aplicativo no celular.

O veículo irá se posicionar na entrada, de modo que a câmera conseguirá visualizar a placa. Será feita a captura da imagem, a identificação dos caracteres e o reconhecimento de quais letras e algarismos estão na placa, para posterior cadastro da placa no sistema.

A ideia inicial é trabalhar com a identificação de veículos particulares brasileiros (com placas de fundo cinza claro e caracteres pretos). Podendo ser ampliada com modificações no sistema para outros modelos de placas de veículos.

De maneira resumida, as atividades do sistema serão a captura da imagem para identificação da placa do veículo, em seguida, deve ser feita a identificação dos 7 caracteres presentes na placa, que serão reconhecidos como letras e algarismos. Por fim, a placa lida pelo sistema será adicionada à lista de placas presentes no Cine Drive-in, que permite o acesso ao cardápio e às compras.

4.1 **Descrição de Hardware**

O projeto será confeccionado através do uso de um Raspberry Pi 3, uma câmera apropriada (Webcam conectada via USB). Para o posicionamento apropriado da câmara será feita uma visita ao local.

4.2 **Descrição de Software**

Para o software, as atividades são a identificação da placa, o reconhecimento dos caracteres e o aplicativo que permitirá o acesso ao cardápio e aos pedidos. O processo de captura de imagens é feito através da função “fswebcam”, cada imagem é salva, analisada e em seguida substituída pela posterior, para não haver acúmulo de imagens desnecessárias na memória.A Figura 2 e 3 apresentam duas das imagens capturadas com o uso do *Raspberry Pi*.

Figura 2: placa 1 capturada pelo sistema.

Figura 3: placa 2 capturada pelo sistema.

Para uma melhor identificação dos caracteres da placa, foi utilizado o Código 1 em anexo, escrito em *Python*, o código realiza um processo de filtragem da imagem, aumentando seu o brilho, contraste e possibilitando uma binarização da mesma. A medida de se utilizar o código citado foi tomada, pois este facilitou o uso do Tesseract para a conversão da imagem em texto. As Figuras 4 e 5, apresentam as imagens captadas pelo sistema após o tratamento do filtro e binarização.

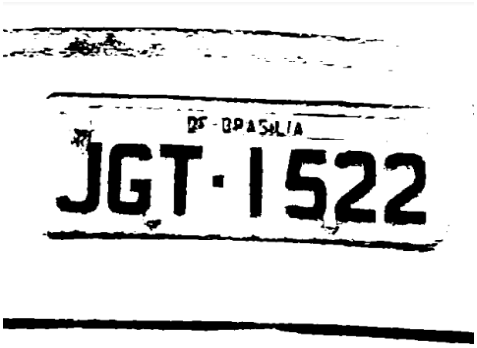

Figura 4: imagem da placa 1 captada pelo sistema após filtragem e binarização.

Figura 5: imagem da placa 2 captada pelo sistema após filtragem e binarização.

Enquanto as Figuras 6 e 7 mostram a leitura realizada pelo *Tesserect* dessas placas, respectivamente.

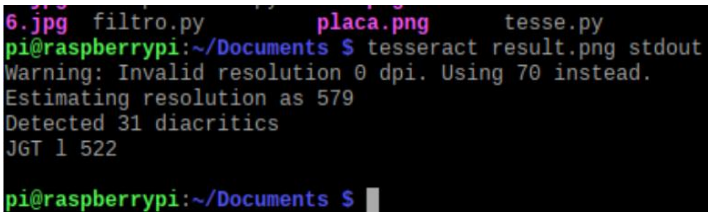

Figura 6: leitura feita pelo *Tesserect* da imagem da placa 1.

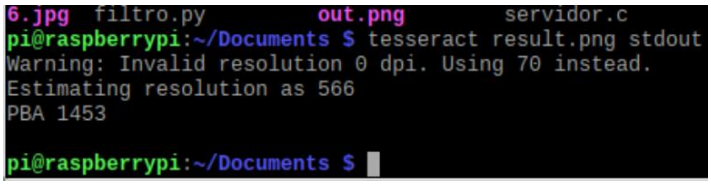

Figura 7: leitura feita pelo *Tesserect* da imagem da placa 2.

4.2.1. **Aplicativo**

O aplicativo que possibilitará a realização dos pedidos, trata-se de um aplicativo desenvolvido no APP INVENTOR[12], que apresenta ao usuário em sua tela inicial duas abas com o cardápio do Cine Drive-In, sendo uma primeira aba à esquerda com toda a disponibilidade de lanches (Figura 8) e uma segunda à direita contendo as bebidas disponíveis (Figura 9), ambas as abas apresentam os preços atualizados.

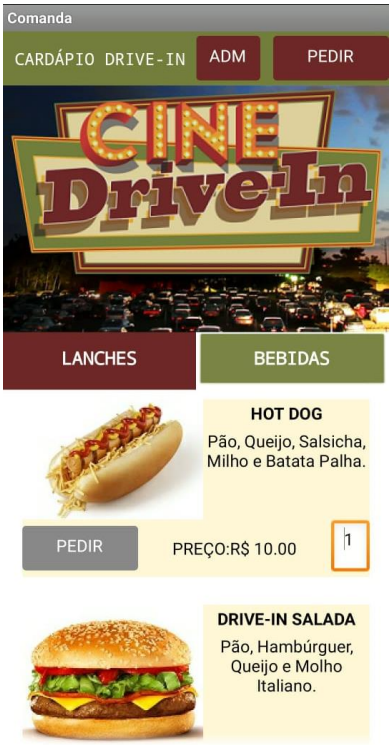

Figura 8: Tela inicial aba com a categoria “LANCHES”.

Figura 9: Tela inicial aba com a categoria “BEBIDAS”.

Figura 10: Tela secundária apresenta lista de pedidos, total a pagar e placas que foram captadas pelo sistema..

Nas duas categorias apresentadas na tela inicial do aplicativo, “LANCHES” e “BEBIDAS”, o usuário tem a opção de escolher a quantidade de cada pedido que deseja. Após escolher a quantidade, para adicionar o pedido à lista destes, ele precisará apertar no ícone “PEDIR”. Feito isso, esse ícone passará a ser apresentado como “NA LISTA”, significando que o pedido já foi adicionado a essa.

A tela inicial do aplicativo ainda conta com um ícone “ADM” para uso exclusivo dos administradores do Cine Drive-in, o acesso a esse ícone só é permitido mediante uma senha e nele consta uma lista com todos os pedidos realizados pelos usuários do cinema ao longo de cada sessão, bem como localização do veículo de cada usuário e características do modelo do carro. À medida que os usuários vão finalizando seus pedidos a lista desse ícone vai sendo atualizada com eles.

Para visualizar seus pedidos, o usuário terá que ir até o ícone “PEDIR” localizado no canto superior da tela inicial. Desse modo, uma segunda tela (Figura 10) lhe será apresentada, nela constará todos os pedidos feitos por ele e o valor total dos pedidos. Caso deseje alterar o pedido, basta ele apertar o ícone “VOLTAR” adicionar ou retirar algum pedido e repetir todos os procedimentos descritos anteriormente.

Ainda nessa segunda tela, constará os “DADOS DO VEÍCULO”, onde ele escolherá a placa respectiva ao seu veículo, terá a opção de descrever a localização e as características do seu carro a fim de facilitar a entrega, ou caso o usuário considere necessário fazer algum comentário acerca de sua localização.

Conforme foi dito anteriormente, após a finalização do pedido, este será enviado para uma planilha, através de um código executado no Google Scripts, que atualizará a lista com pedidos realizados e todas as suas respectivas informações, além da tela de ADM, o que permitirá que um pedido entregue seja apagado, assim como maior controle da ordem de pedidos realizados.

Acerca das placas atualizadas no componente de seleção na tela de Pedidos, esta também é realizada através de um código executado no Google Scripts que permite a inserção após a leitura de novas placas à lista. Além disso, gera também uma URL com o texto da coluna referente às placas adicionadas. Essa função é realizada utilizando recursos da Google API que possibilita leitura de colunas e torna mais fácil a inserção dessas *strings* no aplicativo(Figura 11).

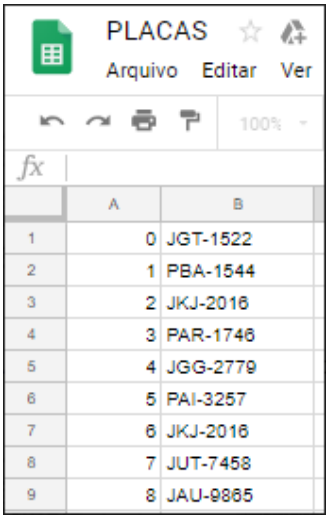

Figura 11: planilha de placas inseridas.

V. RESULTADOS

O aplicativo apresentou conexão com o sistema de reconhecimento de placas veiculares, o que possibilitou o preenchimento das placas na planilha que atualizava-se a cada nova leitura de placa do sistema e, consequentemente, atualizava o componente do aplicativo.

Além disso, o sistema de reconhecimento de placas apresentou excelentes resultados quando comparados aos trabalhos anteriores, pois uma mesma configuração de brilho, contraste e corte foi suficiente para o reconhecimento correto de placas em situações distintas de luminosidade.

Portanto, o projeto apresenta correto funcionamento, com exceção da retirada da placa da lista, que ainda não foi implementado. A lista de pedidos foi enviada corretamente pela planilha e na tabela de administrador, assim como as placas foram utilizadas para preencher o componente de seleção do aplicativo.

Como melhoria, pretende-se enviar o valor total de cada veículo para uma tabela que permita a verificação de pagamento, para que sua saída seja liberada. Além disso, pretende-se implementar a retirada das placas da lista, de acordo com cada nova leitura da placa que ocorrerá na saída do local.

VI. REFERÊNCIAS BIBLIOGRÁFICAS

[1] Nascimento, Jean Dias do. "Detecção e reconhecimento de placa automotiva com baixo custo." (2012).

[2] Leite, B.B., "Localização Automática de Placas de Veículos Automotores Particulares em Imagens Digitalizadas", Projeto Final, DEL/UFRJ, junho, 2003.

[3] "Cine Drive-in Brasília - Um cinema fora de Série". Disponível em: http://cinedrivein.com/. Acesso em: 21 de outubro de 2018.4

[4] "ALPR - Motorola". Disponível em: http://motorolasolutions.com . Acesso em: 21 de outubro de 2018.

[5] WEBER, Henrique; "Um protótipo móvel para detecção automática de placas veiculares brasileiras", UFRGS, 2013.

[6] ALPR - CobanTech"; Disponível em: http://www.cobantech.com/www/ALPR.html . Acesso em: 21 de outubro de 2018.

[7] "Comanda Eletrônica ALTEC". Disponível em: https://www.altec.ws/solucoes/comanda-eletronica/ . Acesso em: 22 de outubro de 2018.

[8] "Pocket Cheff". Disponível em: http://www.cheffsolutions.com.br/produtos/pocket-cheff/. Acesso em: 22 de outubro de 2018.

[9] "Atendi". Disponível em: http://www.atendi.com.br/solucoes#atendimento. Acesso em: 22 de outubro de 2018.

[10] Gary Bradski,Adrian Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library, Ed. 39; Reilly Media, Inc.

[11] Smith, Ray. "An overview of the Tesseract OCR engine." Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on. Vol. 2. IEEE, 2007.

[12] App Inventor; Disponível em: http://ai2.appinventor.mit.edu/ . Acesso em: 22 de outubro de 2018.

ANEXOS

→ Código 1: tratar_foto.py

```
import                                          cv2
import            numpy            as            np
from           PIL           import           Image

# Open a typical 24 bit color image. For this kind
of            image            there            are
#   8   bits   (0   to   255)   per   color   channel
img = cv2.imread('placa.jpg')
#   mandrill   reference   image   from   USC   SIPI

s                       =                       640
img = cv2.resize(img, (s,s), 0, 0, cv2.INTER_AREA)

def apply_brightness_contrast(input_img, brightness
=         0,         contrast         =         0):

    if          brightness          !=          0:
        if          brightness          >          0:
            shadow           =           brightness
            highlight             =             255
        else:
            shadow               =                0
            highlight    =    255    +    brightness
        alpha_b    =    (highlight    -    shadow)/255
        gamma_b               =               shadow

        buf  =  cv2.addWeighted(input_img,  alpha_b,
input_img,               0,               gamma_b)
    else:
        buf            =            input_img.copy()

    if           contrast           !=           0:
        f    =    131*(contrast    +    127)/(127*(131-
contrast))
        alpha_c                 =                 f
        gamma_c             =             127*(1-f)

        buf = cv2.addWeighted(buf, alpha_c, buf, 0,
gamma_c)

    return                                      buf

b                        =                        60
c                        =                       130

out    =    apply_brightness_contrast(img,    b,    c)

cv2.imwrite('out.png',   out)   #aumentar   brilho   e
contraste

img            =            cv2.imread('out.png',0)
ret,thresh1                                      =
cv2.threshold(img,72,255,cv2.THRESH_BINARY) #preto e
branco

cv2.imwrite('result.png',  thresh1)  #resultado  para
tesseract
```