Acionamento de portão eletrônico através da leitura de placas de veículos

Julie Delchova Rabelo
Engenharia Eletrônica
Universidade de Brasília
Brasília, DF
Email:juliedelchova@aluno.unb.br

Marcelo dos Santos Júnior
Engenharia Eletrônica
Universidade de Brasília
Brasília, DF
Email:marcelo.junior@aluno.unb.br

***Abstract*—Proposta de projeto da disciplina Sistemas Operacionais Embarcados. Acionamento de portão eletrônico a partir da detecção de placas de veículos cadastrados em banco de dados.**

I. Introdução

Visão computacional é o campo científico que estuda formas e técnicas de fazer com que computadores compreendam ou extraiam informações de imagens digitais e de vídeos.

A implementação de sistemas que utilizam Visão Computacional é uma alternativa eficiente para aplicações que envolvam análise de dados. A partir de determinada imagem, é possível extrair uma gama de dados, sendo assim, viabiliza-se pontuar a sua aplicabilidade no monitoramento de placas veiculares de identificação, o que visa principalmente a segurança. Por este motivo, além de proporcionar um maior conforto e automação de processo, o presente projeto propõe a costrução de um sistema de acionamento de portão eletrônico a partir da reconhecimento de placas veiculares. As placas terão de ser cadastradas no banco de dados para habilitar a abertura dos portões.

Para o funcionamento do sistema proposto, os principais conceitos a serem abordados são visão computacional, o hardware selecionado e biblioteca OpenCV.

A. Visão Computacional

A Visão Computacional é a área da Inteligência artificial que estuda processamento de imagens do mundo real por computador. Os métodos incluem a aquisição, o processamento e a análise e compreensão destas imagens, ou seja, a extração dos dados do mundo real que produza informações numéricas ou simbólicas compreeensíveis ao computador, que ele utiliza para tomar decisões. Ou seja, tem o objetivo de obter algoritmos capazes de interpretação visual de imagens, extraindo seus dados. (PEREIRA, 2016)

B. Raspberry Pi

Rasberry Pi se trata de um computador do tamanho de um cartão de crédito e de baixo custo. É um hardware capaz de navegar na internet, reproduzir vídeo, manipular planilhas, processar texto, compilar programas e jogar jogos, tendo a capacidade de interagir com o mundo exterior através da conexão GPIO (General Purpose Input/Output).

C. OpenCV

De acordo com Pereira (2016), o Open Source Computer Vision Library (OpenCV) é uma biblioteca de visão computacional open-source multiplataforma que foi desenvolvida pela Intel. Esta reúne os recursos necessários para se realizar projetos com Visão Computacional. Ela abrange: aquisição/obtenção de imagens de câmeras digitais, tratamento e processamento de vídeos, entre outros.

D. Tensorflow Lite

Tensorflow Lite é um software *open source* que fornece um framework para o desenvolvimento de redes neurais. Ele converte um modelo de rede pré-treinado por Tensorflow para um formato especial capaz de ser otimizado em velocidade e capacidade de armazenamento.

Este formato pode ser executado por dispositivos como telefones celulares com sistema operacional Android ou IOS e também por dispositivos que funcionam com o sistema Linux, como é o caso da Raspberry Pi.

II. Requisitos

- O sistema irá detectar e ler as placas dos carros a partir do módulo de câmera da placa *Raspberry Pi 3B+*.

- Validar a placa analisada por meio de busca de resgistros existentes em um banco de dados.

- O acionamento do portão será feito através da adaptação do circuito do controle de usuário.

- Contará com sistema de refrigeração a partir de um *cooler*.

- O sistema principal vai ser acionado por meio de um botão.

III. HARDWARE

O microprocessador utilizado é a *Raspberry Pi 3B+*. A partir das suas especificações é possível trabalhar com projetos que envolvem Visão Computacional. Isso se dá por conta de seu alto poder de processamento. Para melhor performance do microprocessador, o sistema contará com um mecanismo de refrigeração: um *cooler* acoplado na case que guardará a placa em questão. O módulo da câmera de vídeo será conectado ao microprocessador por meio de um cabo *flat*, registrando até 30 quadros por segundo. O acionamento se dá a partir da adaptação do circuito de controle do usuário, no qual o módulo relé de 1 canal será o responsável por mandar o sinal em nível logíco alto ou baixo. O sistema contará também com uma fonte de alimentação de 5V para energizar a *RaspBerry Pi 3B+*, além de um botão de acionamento do sistema principal, para a economia de energia e de processamento da placa. Este é também um recurso financeiramente mais atrativo.

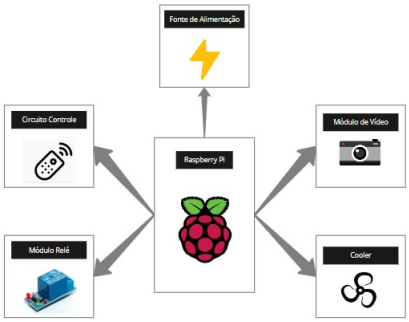

Fig. 1: Esquemático de demonstração do hardware do projeto

A. Tabela de custos

Componentes	Quantidade	Preço/un (R$)	Preço total (R$)
Raspberry Pi 3B+	1	273,60	273,60
Cartão SD 64GB	1	79,90	79,90
Câmera para RP3	1	39,50	39,50
Relé de 1 canal	1	8,30	8,30
Jumpers fêmea-fêmea	40	0,46	18,40
Controle Rossi TX 433 MHz	2	36,90	73,80
TOTAL	-	-	493,50

TABLE I: Componentes e seus respectivos preços .

B. Testagem dos Componentes Eletrônicos

Para a utilização da placa de desenvolvimento *Raspberry Pi 3B+* (RP3) , a mesma foi iniciada com o sistema operacional *Rasbian* que constituído por um *Debian* próprio para a RP3. Para realizar o *boot* do sistema operacional na placa, foi utilizado o software ”Raspberry Pi Imager”, responsável por montar o sistema em questão no cartão SD de 64 GB e classe 10.

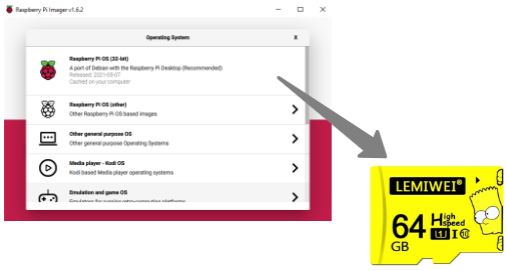

Fig. 2: Software ”Raspberry Pi Imager” e cartão SD utilizao no projeto.

Para o uso do módulo de câmera foi preciso a habilitação desta por meio do terminal. Os testes de funcionamento se deram por meio de código em Python. Certificou-se, desta forma, de que o módulo de câmera funciona corretamente para a realização do projeto.

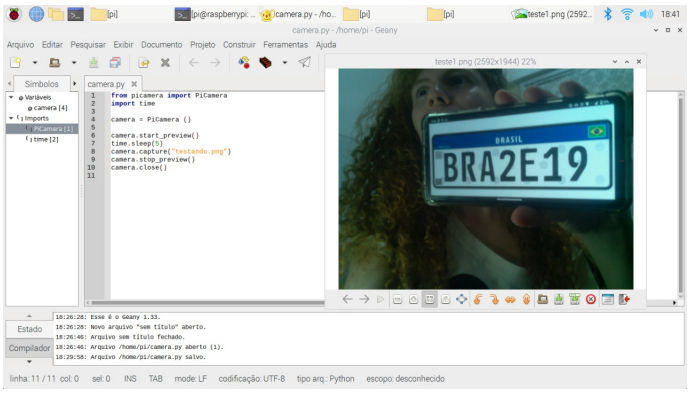

Fig. 3: Teste do módulo câmera da RP3.

Para o teste de funcionalidade do módulo relé de 1 canal, optou-se por um teste simples em Python. Para este teste foi realizada a seguinte montagem:

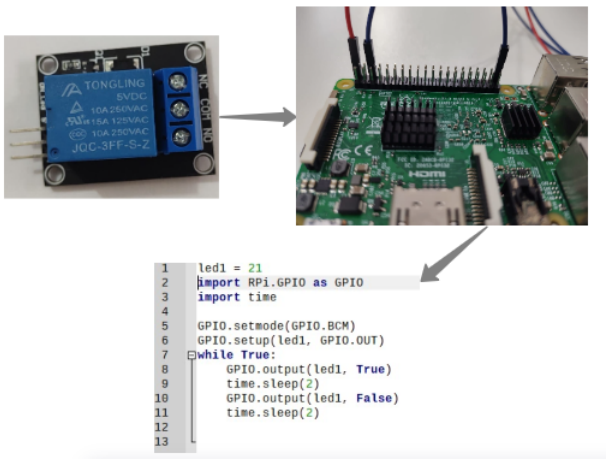

Fig. 4: Teste do módulo relé da RP3.

As especificações das ligações da montagem do circuito

de acionamento do módulo relé são:

- Pino 2 para o V_{CC} (+) ;
- Pino 6 para o GND (-) ;
- GPIO 21 para o canal de controle (S).

Para acionar o circuito do controle *Rossi*, o canal usado foi o *NO*, pois este se encontra normalmente aberto e não tem passagem de corrente enquanto o relé não for acionado.

Assim sendo, verificou-se que o módulo relé de 1 canal adquirido apresenta-se funcional e apto para uso.

C. Adaptação do circuito de controle

A partir a conexão do módulo relé anteriormente descrita, energizado no pino 2, com GND no pino 6 e com o canal de controle no pino 21, ocorrerá o acionamento do circuito do controle de usuário. Os jumpers soldados nos terminais de alguns dos botões do controle, como observado na figura 5, são conectados no canal *NO* (Normalmente Aberto) e no canal *C* (Comum).

(a) Circuito de controle (parte anterior).

(b) Circuito de controle (parte posterior)

Fig. 5: Circuito do controle do usuário.

IV. Metodologia

Para o reconhecimento da placa do carro e posterior abertura do portão onde no qual o sistema será instalado, são necessárias as 7 etapas ilustradas na figura 6.

Fig. 6: Diagrama de Blocos para o reconhecimento de placa .

A primeira etapa, obtenção da imagem, se refere à captura da imagem do veículo ao se aproximar do portão. Para isso, a câmera deve ser posicionada estrategicamente de maneira a enquadrar ao máximo a placa do veículo e reduzir ruídos que possam atrapalhar a identificação da placa. Entende-se como ruído aqui a ausência de iluminação, objetos semelhantes às placas automotivas, ângulo da imagem, dentre outros. Para o projeto, devido ao espaço limitado, a câmera foi posicionada na diagonal em relação a posição do veículo. Na figura 7 podemos visualizar a imagem obtida pela câmera do sistema.

Fig. 7: Imagem capturada pela câmera do sistema.

A segunda etapa do processo consistiu na localização da placa do veículo na imagem capturada pela câmera. Como podemos observar, a imagem bruta possui alguns elementos que dificultam a identificação da placa, como a vegetação no canto esquerdo. Esse obstáculo faz com que o método de extração aplicando contornos, do módulo OpenCV, se torne inviável para esse projeto. Para essa situação, a técnica de localização que mais se adaptou foi treinar uma rede neural convolucional para reconhecer placas automotivas. Utilizou-se, então, o TensorFlow, um sistema para criação e treinamento de redes neurais, para detectar e decifrar padrões e correlações criados pelo Google Brain.

A inteligência artificial treinada obteve resultados satisfatórios. A figura 8 traz a imagem da câmera já com a placa do veículo destacada pela IA.

Fig. 8: Imagem da placa localizada.

Com a localização da placa, é possível extraí-la da imagem bruta e salvá-la, temporariamente, em um arquivo ".png", separadamente. Essa é a terceira etapa do processo, possibilitado pela utilização do módulo OpenCV. A figura 9 traz a imagem 'Placa.png' resultante dessa etapa do processo.

Fig. 9: imagem "Placa.png"

Nem sempre os caracteres podem ser extraídos diretamente da imagem "Placa.png", sendo necessário, então, uma etapa de tratamento para reduzir, novamente, os ruídos e intensificar o contraste das letras em relação ao ambiente em volta. Isso é realizado na etapa 4.

O primeiro tratamento da etapa 4 é a transformação da imagem em escala de cinza. O tratamento se faz necessário pois diversas funções do OpenCV esperam receber uma imagem em tons de cinza. O segundo tratamento é a suavização da imagem, que tem a finalidade de eliminar traços ruidosos que podem ser confundidos, posteriormente, com caracteres. A figura 10 traz a imagem "Placa.png" suavizada.

Fig. 10: Imagem em tons de cinza e suavizada

O terceiro e último tratamento é a Limiarização da imagem. Esse último consiste em deixar a imagem preta e branca com a finalidade de realçar os caracteres. Existem diversas técnicas de limiarização cujos resultados podem ser observados na figura 10. A limiarização que melhor se adaptou a essa aplicação, por diferenciar de maneira efetiva a letra "A" do número "4", foi utilizando a técnica Thresh Binary com Limiar de 120.

Agora, com a imagem da placa já tratada, foi possível realizar a leitura dos caracteres, sendo essa a quinta etapa do processo. Para essa tarefa, utilizou-se o módulo Tesseract, ferramenta que transforma textos de imagens em strings. Mesmo após o tratamento da etapa 4, ainda é possível obter alguns caracteres indesejados, como símbolos e letras minúsculas. Caracteres estes que não fazem parte do texto das placas brasileiras. Portanto, ainda na etapa 5, a fim de minimizar estas imprecisões, ocorre uma filtragem na string obtida.

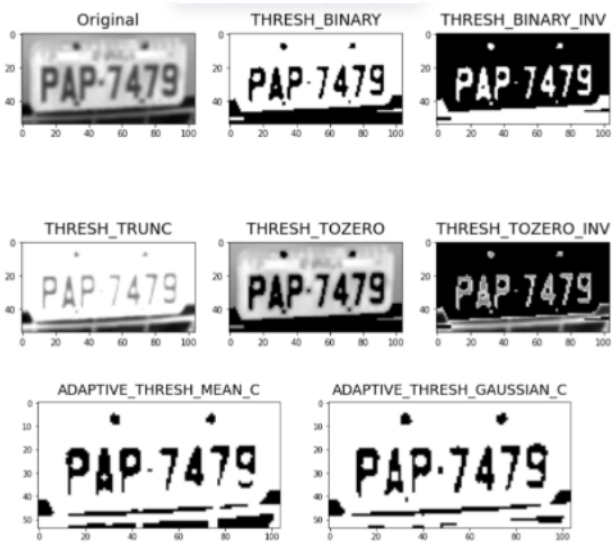

Fig. 11: Diversas técnicas de limiarização.

A etapa 6 representa a consulta as placas cadastradas e a verificação da *string* obtida. Caso ela seja equivalente a placas com acesso autorizado, o portão é acionado. As placas permitidas foram cadastradas previamente em um banco de dados SQLite chamado "Garagem.db", de modo que, com uma consulta simples utilizando a função SELECT, é possível autenticar a string obtida na etapa 5 se houver o retorno de algum registro. Caso não retorne registros, a etapa 1 se inicia novamente. O procedimento poderá se repetir por até vinte vezes, caso não seja possível identificar a placa do carro. Se, após as vinte tentativas, a string não for encontrada no banco de dados, o programa não realizará novas tentativas.

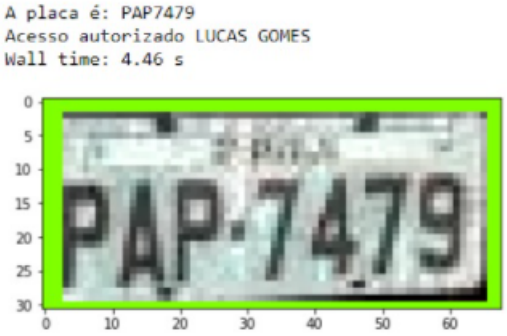

Fig. 12: Autenticação de placa e liberação do acesso.

Por fim, com a placa já validada, a Raspberry Pi aciona um relé que está adaptado a um circuito de controle transmissor. O transmissor, então, aciona o motor do portão eletrônico e aguarda 20 segundos para enviar um segundo comando para fechá-lo.

V. Integração: Hardware e Software

A. Instalação das bibliotecas necessárias

As bibliotecas *OpenCV*, *TensorFlow Lite* e *OCR-Tersseract* foram fundamentais para a implementação dos códigos na RaspberryPi 3B+.

B. Interface com usuário

A interface se dá através do acesso remoto a Raspberry Pi 3B+. No intuito de atender à necessidades do usuário, este terá a possibilidade de acessar o microprocessador do seu computador. A interface se dá por meio de aplicativos do Windows (*Windows PowerShell* e *Conexão de trabalho remota*). É possível pelo SSH (*Secure Socket Shell*), uma função do microprocessador em questão, se trata de um protocolo de rede . Permitindo assim a adição de novos usuários e suas respectivas placas.

C. Repositório

O repositório para o projeto se encontra em https://github.com/marcelostosjr/SOE_OCR$_Placa_de_Veiculos$.

D. Link da Apresentação

https://web.microsoftstream.com/video/3bb56892-fcdc-4a01-9bbc-b7bf61aa2fc7

References

SILVA DE SOUZA, Guilherme Stefano; PASSELLA, Paulo Henrique. Reconhecimento Automático De Placas De Veículos Utilizando Processamento Digital De Imagens E Inteligência Artificial. 2011. 173 p. Bacharel (Bacharelado em Ciência da Computação) – Centro Universitário UNISEB, Ribeirão Preto, 2011.

PEREIRA, C. A. DE OLIVEIRA, S. M.M. Detecção De Pessoas Em Imagens, Implementando Técnicas De Visão Computacional Em Um Raspberry Pi. 2016.

LISBOA, Maurício O. P. Raspberry Pi, Linux (Raspbian). USP. 2019.

Muhammad Tahir Qadri, Muhammad Asif, "Automatic Number Plate Recognition System for Vehicle Identification Using Optical Character Recognition" IEEE 2009.

Optasia System Pte Ltd, "The World Leader in License Plate Recognition Technology" Sourced. Disponível em : ¡www.singaporegateway.com/optasia¿ .Acesso em 28 de Setembro de 2021.

S.H. Park, K.I. Kim, K. Jung and H.J. Kim, "Locating car license plate using Neural Network," Electronics Letters, Vol. 35, No.17 ,pp. 1474-1477,1999.

KHANDELWAL, Renu. A Basic Introduction to TensorFlow Lite. 2020. Disponivel em : ¡https://towardsdatascience.com/a-basic-introduction-to-tensorflow-lite-59e480c57292¿ Acesso em : 15 de Outubro de 2021.

KLETTE, Reinhard, "Concise Computer Vision," Springer, ISBN : 978-1-4471-6319-0, 2014

HUANG, T.; VANDONI, Carlo. Computer Vision : Evolution And Promise. pp. 21–25. 19th CERN School of Computing. Geneva: CERN. 1996