

# Controle Eletrônico de Acesso de Veículos

Victhor Alexandre Vilarins Cardoso da Silva

Faculdade de Ciências e Tecnologia em Engenharia(FCTE)

Universidade de Brasília, Campus UnB Gama

Brasília - DF, Brasil

180037978@aluno.unb.br

Yuri Virginio da Silva Santos

Faculdade de Ciências e Tecnologia em Engenharia(FCTE)

Universidade de Brasília, Campus UnB Gama

Brasília - DF, Brasil

190129913@aluno.unb.br

**Resumo**—Este trabalho apresenta o desenvolvimento de um sistema automatizado de controle de acesso veicular baseado no reconhecimento de placas, utilizando técnicas de visão computacional e processamento de imagem. O sistema foi projetado para identificar placas de veículos, permitindo a entrada automática de veículos autorizados e restringindo o acesso de veículos não cadastrados.

**Palavras-chave**—OpenCV, Reconhecimento de placas, Reconhecimento óptico de caracteres, OCR, Processamento de imagens.

## I. INTRODUÇÃO

Apesar do desenvolvimento tecnológico observado nas últimas décadas, há sempre espaço para integrar soluções de automatização e digitalização em processos. Quando se fala em controles de acesso de veículos, ainda há lugares que realizam autorização de acesso manualmente, e, em alguns casos, o controle desses acessos é registrado, quando registrado, em cadernetas. Esse tipo de prática dificulta auditorias, e representa um risco à segurança. Atualmente, no campus Gama da UnB, o controle de acesso de veículos é realizado por um vigia que realiza a identificação e liberação dos veículos credenciados por um selo. Esse sistema, apesar de ser mais moderno, ainda convida melhorias.

Uma forma de automatizar esse processo é através do reconhecimento de caracteres da placa dos veículos. O avanço das tecnologias de visão computacional e inteligência artificial possibilitou o desenvolvimento de sistemas de controle de acesso baseados no reconhecimento automático dessas placas. Esses sistemas têm sido amplamente aplicados em áreas como estacionamentos, portarias de condomínios e zonas de acesso restrito, visando aprimorar a segurança e a eficiência no gerenciamento de entradas e saídas. Ainda em 2004, já haviam modelos de reconhecimento de placa com taxa de sucesso de 93,7%. [2]

Neste trabalho, propõe-se um sistema de controle de acesso de veículos que utiliza a captura e reconhecimento automático de placas para identificar veículos autorizados, e registrar essa entrada em um banco de dados, eliminando a necessidade de validação de acesso por um humano e aumentando a segurança do ambiente monitorado. Não obstante, a liberação manual por um vigia permanece possível, que também será registrada.

Para o controle de acesso, propõe-se utilizar a Raspberry PI 3 para controlar o sinal enviado a um portão eletrônico, simulando o sistema de entrada de veículos da universidade,

para autorizar a entrada de veículos que estão cadastrados automaticamente. Além disso, esse projeto visa armazenar informações da data e hora de entrada de cada veículo, possibilitando a consulta de informações em uma auditoria.

Foram encontrados outros trabalhos semelhantes de uso de visão computacional e OCR para reconhecimento de placa de veículos e controle da cancela. No repositório [3], encontra-se uma solução utilizando as bibliotecas em python do OpenCV e Tesseract, para fazer esse reconhecimento de placas. Em [1], utilizando um computador pessoal, observou-se a aplicação dessa solução na cancela da universidade, alcançando uma taxa de 93,54% de reconhecimento das placas. Contudo, o sistema enfrentou dificuldades para identificar as placas de veículos em movimento, devido ao efeito de desfoque nas imagens capturadas.

Este projeto se concentra no desenvolvimento e teste de um protótipo funcional que utiliza uma câmera para capturar imagens, processá-las para identificar caracteres da placa e tomar decisões de autorização ou negação de acesso do veículo.

## II. DESENVOLVIMENTO

O projeto foi desenvolvido em etapas, visando estabelecer uma progressão que permitisse verificar a compatibilidade de cada parte e assegurar a viabilidade de implementação do projeto.

A figura 2 mostra o fluxograma do projeto.

### A. Descrição do Hardware

Para este projeto é utilizado a Raspberry Pi 3 e uma câmera. A placa possui um processador Broadcom BCM2837 Quad Core 1.2GHz 64 bits, memória RAM com 1 GB e conta também com outras especificações listadas abaixo

- BCM43438 LAN sem fio e Bluetooth Low Energy (BLE) a bordo 100 Base Ethernet
- GPIO estendido de 40 pinos
- 4 portas USB 2
- Saída estéreo de 4 polos e porta de vídeo composto
- HDMI® de tamanho completo
- Porta de câmera CSI para conectar uma câmera Raspberry Pi
- Porta de exibição DSI para conectar uma tela sensível ao toque Raspberry Pi

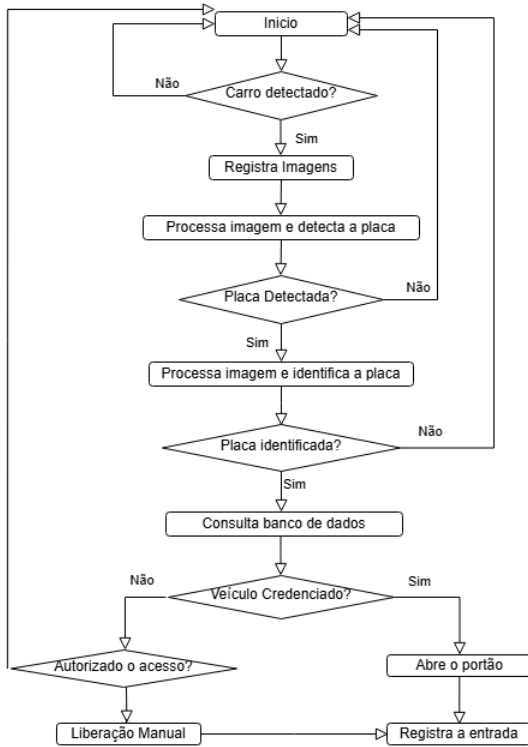


Figura 1. Fluxo de processo

- Porta Micro SD para carregar seu sistema operacional e armazenar dados
- Fonte de alimentação Micro USB comutada atualizada até 2,5 A

Além da placa, é utilizada no projeto uma câmera com entrada USB, compatível com USB 1.1 e 2.0, resolução máxima de 1080p e taxa máxima de transmissão de 30 quadros por segundo. Para detecção do veículo na área destinada à identificação, é empregado o sensor de distância ultrassônico HC-SR04, capaz de medir com precisão, distâncias entre 2cm e 4m, com ângulo de abertura de 15° e frequência de operação de 40kHz. Por fim, o optoacoplador 4N25 atua como uma chave eletrônica em um controle remoto de um portão, já configurado para emitir o sinal RF de abertura e fechamento do portão.

Tabela I  
MATERIAIS UTILIZADOS

Quantidade	Componente	Preço Unit.	Preço Total
1	Kit Raspberry basic	R\$ 446,60	R\$ 446,60
1	Câmera Webcam	R\$ 40,00	R\$ 40,00
1	Resistor 10kΩ	R\$ 0,13	R\$ 0,13
1	Resistor 300Ω	R\$ 0,04	R\$ 0,04
1	Resistor 220Ω	R\$ 0,04	R\$ 0,04
1	Optoacoplador 4N25	R\$ 2,75	R\$ 2,75
1	Sensor HC-SR04	R\$ 10,40	R\$ 10,40
	<b>TOTAL</b>		<b>R\$499,96</b>

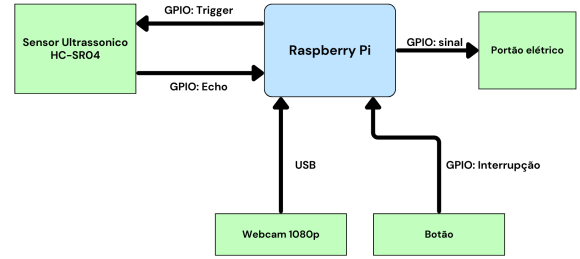


Figura 2. Diagrama de Blocos

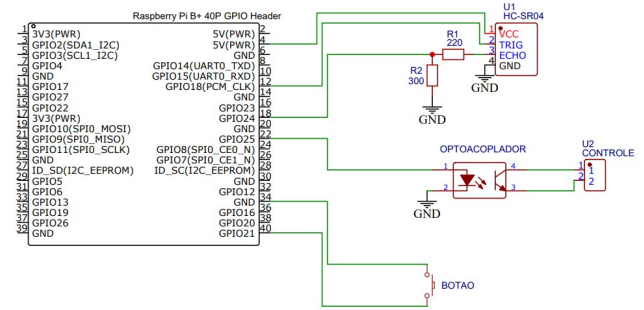


Figura 3. Esquemático do circuito

## B. Descrição do Software

Para este projeto, foi utilizado o Sistema Operacional: Raspberry Pi OS (64-bit) com Desktop.

A versão do Tesseract OCR utilizada foi a versão 5 para Raspberian (disponível em: <https://notesalexp.org/tesseract-ocr/packages5/en/raspbian/buster/armhf/>). A versão do OpenCV utilizada foi a 4.8.0

Para tirar a foto coma webcam conectada ao USB do Raspberry Pi, utiliza-se o código:

```
fswebcam -r 1920x1080 -v -S 10 --set brightness
=50% test_image.jpg
```

Onde o argumento -r define a resolução da imagem, o argumento -S 10 pula 10 frames antes de capturar a imagem (isso ajuda um pouco a evitar glitches nas imagens). O brilho aplicado pode ser incorporado a um switch-case, onde dependendo do horário do dia, o brilho pode ser maior ou menor.

O loop principal do programa consiste em seis etapas:

- 1) Verificar se há detecção de veículos com o sensor de ultrassom
- 2) Tirar fotos com a câmera
- 3) Processar as fotos e detectar a placa do veículo
- 4) Armazenar a placa detectada no banco de dados
- 5) Liberar o acesso do veículo abrindo o portão através de envio de sinal
- 6) Aguardar o veículo se afastar do sensor de ultrassom, enviar sinal para fechar o portão, e reiniciar o processo

O programa usado foi desenvolvido em C++, e segue o fluxograma a seguir:

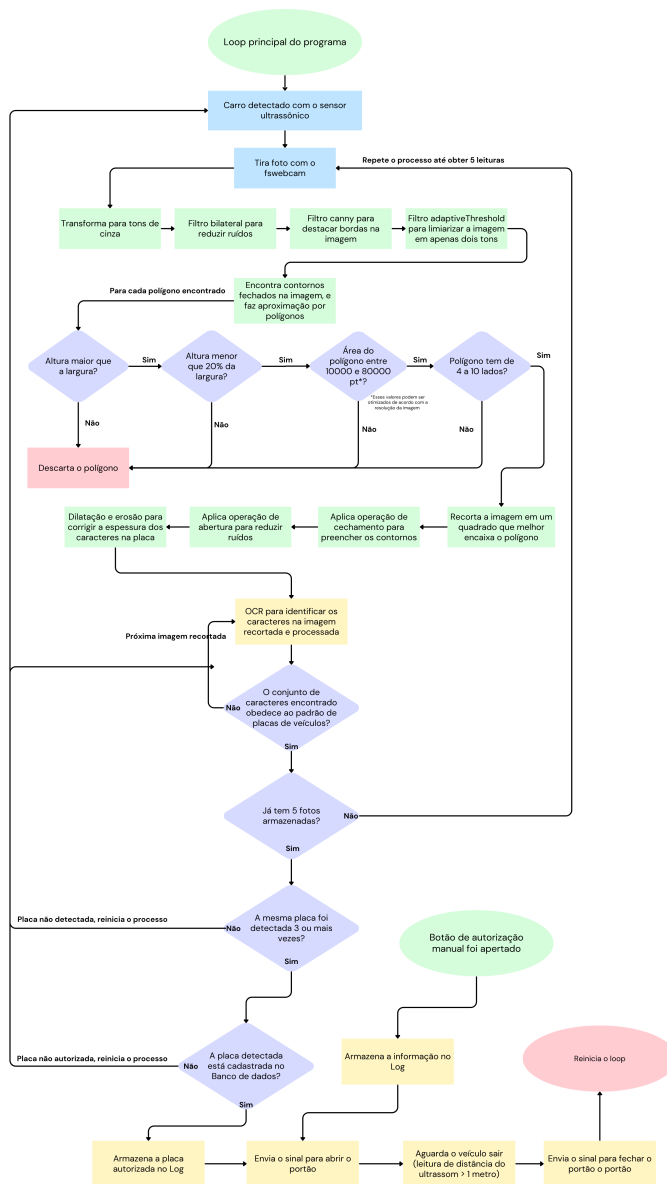


Figura 4. Fluxo de software

Após detectar o veículo (sensor de ultrassom acusar 100cm ou menos), o programa inicia a rotina de verificação. Uma foto é tirada, e armazenada temporariamente, para ser processada.

Primeiro, há um tratamento inicial da imagem, com aplicação de filtros. Depois é feito um processamento para encontrar contornos na imagem, e identificar possíveis retângulos na imagem processada. Nesta etapa, para cada polígono encontrado, cria-se uma thread para validar o polígono, recortar a imagem, e reprocessar a imagem para remover ruídos, e destacar os caracteres na imagem. Após esse processamento, todas as imagens processadas são enviadas para detecção de caracteres do Tesseract OCR. Os conjuntos de caracteres encontrados pelo OCR são verificados, caso obedeçam ao

padrão de placa do Brasil, ou Mercosul. Se o padrão for encontrado, a placa é registrada em um buffer.

Esse processo é repetido para outras quatro fotos, sendo que uma nova foto só é tirada depois que a última foi processada. Caso uma foto não retorne nenhuma leitura válida, o buffer é preenchido com valor nulo para essa tentativa. Quando o buffer estiver completo, verifica-se se uma mesma placa válida (não nula) se repete três ou mais vezes dentro do buffer. Se não, todo o processo é repetido, tirando e processando cinco novas fotos.

Quando a placa for detectada, verifica-se se o veículo está credenciado na base de dados. Se sim, armazena-se a informação no banco de dados, com o dia e hora da identificação. Então o sinal é enviado para abrir o portão. Para um veículo não credenciado, aguarda-se a liberação manual, e armazena-se a informação no banco de dados.

Por fim, aguarda-se a saída do veículo (sensor ultrassônico acusar distância maior que 100cm). Após um atraso (para garantir que o veículo passou pelo portão), é enviado o sinal para fechar o portão. Então, aguarda-se mais um atraso (de acordo com o tempo necessário para o portão fechar) e reinicia o loop principal, onde será verificado se o sensor ultrassônico identifica a presença de um veículo novamente.

A qualquer momento do programa, um botão pode ser pressionado para ativar o portão manualmente. Nesse caso, uma interrupção é ativada, e o portão é aberto. Nesse momento, é adicionado ao banco de dados a data e hora dessa liberação. Essa interrupção também faz o programa pular para o ponto em que espera-se que o veículo se afaste do sensor ultrassom para fechar o portão.

### III. RESULTADOS

O sistema desenvolvido foi capaz de realizar o reconhecimento e validação das placas dos veículos testados. Foram feitos testes isolados com fotos baixadas da internet, fotos tiradas com a câmera do projeto, e com o sistema totalmente integrado, e o reconhecimento funcionou em todos os testes, ainda que não 100% das tentativas.

Apesar disso, este projeto demonstrou que o sistema proposto é possível, e que com as devidas otimizações pode se tornar uma implementação viável.

O vídeo do funcionamento do projeto pode se visto no link abaixo:

[https://youtu.be/tSyP\\_tU2IjM](https://youtu.be/tSyP_tU2IjM)

#### A. Fraquezas

Durante a fase de implementação do projeto, observou-se que há espaço para melhorias. Alguns fatores como a qualidade da câmera usada, e simplicidade do algoritmo de identificação de placas, e reconhecimento de caracteres pode ter reduzido a eficiência do processo. Em alguns casos, não era possível identificar a placa pelo programa. Idealmente, espera-se que o sistema seja capaz de identificar a placa tão bem quanto o olho humano.

Há outras otimizações possíveis. Apesar de ser uma biblioteca feita para reconhecimento de caracteres, o Tesseract não

é a ferramenta mais potente hoje em dia. Certamente uma das mais leves em custo computacional, mas há outras bibliotecas disponíveis que podem entregar resultados melhores. Na maioria dos testes realizados, observou-se que o algoritmo foi bem sucedido em identificar os contornos das placas, mas falhou em identificar os caracteres.

Em alguns casos, também observou-se que o processamento realizado na foto tornava os caracteres irreconhecíveis. Isso acontecia especialmente em fotos ruidosas, pouco iluminadas ou quando a placa estava muito pequena na imagem. Esse problema foi parcialmente remediado no escopo deste projeto, uma vez que as fotos são tiradas a uma distância fixa do veículo.

Por fim, ao contrário do que se esperava, o que mais afetou o tempo de execução do programa foi o comando para tirar a foto e salvar a imagem. Todas as etapas de processamento da imagem, que ocorreram em threads, foram executadas em menor tempo do que o ato de tirar a foto. O mesmo ocorreu para as etapas de reconhecimento de caracteres, que executou mais rapidamente no Raspberry Pi do que o comando de tirar foto. Essas observações foram feitas de forma empírica, e não foi utilizada alguma estratégia de medida de tempo de execução para validar esses resultados.

#### IV. CONCLUSÃO

Conclui-se com esse projeto que uma solução de baixo custo para implementação de um sistema de reconhecimento de placas de veículos e autorização de entrada é possível. Mesmo usando a raspberry pi, que tem poder computacional menor que um computador pessoal, foram obtidos resultados interessantes. Ainda assim, este projeto tem potencial para melhorias.

Os arquivos e códigos do projeto podem ser acessados em: [https://github.com/victhor-alexandre/Projeto\\_SOE\\_placa\\_ocr](https://github.com/victhor-alexandre/Projeto_SOE_placa_ocr)

#### REFERÊNCIAS

- [1] L. Leite, R. Antonello, IDENTIFICAÇÃO AUTOMÁTICA DE PLACA DE VEÍCULOS ATRAVÉS DE PROCESSAMENTO DE IMAGEM E VISÃO COMPUTACIONAL, Instituto Federal Catarinense, 2017. Disponível em: <https://secitec.luzerna.ifc.edu.br/wp-content/uploads/sites/30/2018/04/Versao-Final-Artigo-para-SECITEC-2017-10-paginas.pdf>
- [2] Chang, Shyang-Lih & Chen, Li-Shien & Chung, Yun-Chung & Chen, Sei-Wang. (2004). Automatic License Plate Recognition. Intelligent Transportation Systems, IEEE Transactions on. 5. 42 - 53. 10.1109/TITS.2004.825086.
- [3] G. Lima, C. A. A. Junior, Identificação de placas de veículos, UENP, Salto Grande, Brasil, 2023. Disponível em: <https://github.com/Gabriellimmaa/reconhecimento-e-leitura-placa-carro-ptBR/tree/main>
- [4] Maker Hero, "Sensor de distância ultrassônico HC-SR04." [Online]. Disponível em: <https://www.makerhero.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>. [Acessado: 13-dez-2024].
- [5] Documentação do Tesseract-OCR disponível em: <https://tesseract-ocr.github.io/tessapi/5.x/> [Acessado: 07-dez-2024].
- [6] Raspberry Pi, "Raspberry Pi 3 Model B+." [Online]. Disponível em: <https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/>. [Acessado: 09-dez-2024].
- [7] IBM (não assinado), What is optical character recognition (OCR)? Disponível em: <https://www.ibm.com/think/topics/optical-character-recognition> [Acessado: 09-dez-2024].
- [8] Documentação do Debian - fswebcam, disponível em: <https://manpages.debian.org/jessie/fswebcam/fswebcam.1.en.html> [Acessado: 12-dez-2024].

- [9] Tutorial Sensor ultrassônico com raspberry pi. Disponível em: [https://tutorials--raspberrypi-com.translate.google.com/raspberry-pi-ultrasonic-sensor-hc-sr04/?\\_x\\_tr\\_sl=en&\\_x\\_tr\\_tl=pt&\\_x\\_tr\\_hl=pt&\\_x\\_tr\\_pto=tc](https://tutorials--raspberrypi-com.translate.google.com/raspberry-pi-ultrasonic-sensor-hc-sr04/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt&_x_tr_pto=tc)