

Controle de acesso por reconhecimento de número de placa

Reconhecimento de número de placa para liberação de acesso de veículos a estacionamento

Aubani Júnio Teixeira Cândido
Universidade de Brasília - UnB
Brasília-DF, Brasil
junio@aluno.unb.br

Matheus Pereira Santana
Universidade de Brasília - UnB
Brasília-DF, Brasil
matheuspereiraunb@gmail.com

Resumo— Aplicação do uso de técnicas de processamento digital de imagens sob sistemas embarcados para o reconhecimento da placa e características de veículos automotivos para permitir acesso a locais, tais como estacionamentos ou portarias de pedágio.

Keywords—processamento de imagens, reconhecimento de placa, Raspberry Pi, sistemas embarcados.

I. JUSTIFICATIVA

Atualmente há uma alta demanda de sistemas que sejam capazes de identificar veículos por meio de imagem para se fazer monitoramento e controle de acesso a determinados locais. Muitos desses locais contam com pessoas para verificar se os veículos têm ou não permissão de acesso. Portanto, um sistema autônomo que realize essa identificação, com uma dada premissa de segurança, pode ser utilizado para realizar esse controle e efetuar o registro dos veículos que tiveram e os que não tiveram acesso ao local.

II. OBJETIVO

A. Garantir Segurança no Acesso

Somente veículos que foram cadastrados no sistema teriam acesso. Para isso seriam gravados os dígitos da placa e imagens do veículo para que seja feita a verificação. Por meio desses dados, previamente cadastrados, cada veículo que se aproximar da cancela irá ser analisado por uma câmera que irá captar imagens do veículo, para confrontar as informações de placa e características do mesmo com as informações do banco de dados.

B. Monitoramento de Acesso

O sistema irá contar com uma página web onde serão exibidos logs dos veículos que tiveram acesso, horário e foto do veículo no momento que obtiveram autorização ou tiveram acesso proibido.

III. TABELA DE MATERIAIS UTILIZADOS

UND	MATERIAIS	FABRICANTE
01	RASPBERRY PI 3	RASPBERRY PI
01	WEBCAM 2MP 3808	HARDLINE
01	Motor de passo 17HS8401	NEMA
01	Trafo Abaixador (15 V + 15 V / 2 A)	INDELETRA
01	Ponte retificadora	-
01	Módulo sensor ultrassônico HC-SR04	ELEC FREAKS
01	Driver A4988	POLULU
01	Regulador de tensão 12 V	KIA
01	Capacitor 4700 μ F	-
01	Capacitor 100 μ F	-
01	Resistor 470 Ω	-
01	Resistor 330 Ω	-
01	Jumpers e Conectores	-

IV. REQUISITOS.

O sistema irá contar com uma Raspberry Pi 3 Model B, uma câmera (que tenha, preferencialmente, um controlador

automático de ganho para compensar o efeito da luz), um motor de passo NEMA 17-23, dois sensores de ultrassom (ou outro sensor de presença), uma página web, um banco de dados para as informações de acesso, bibliotecas para reconhecimento de placas de veículos, bibliotecas para processamento digital de imagens com funções de espectro e identificação de contornos.

V. BENEFÍCIOS

O sistema concerne ao usuário a praticidade de não precisar se identificar manualmente a cada acesso ao local.

O sistema irá usar dos sensores de presença para controlar a captação de imagem e o processamento da mesma, para otimizar as operações realizadas com as mesmas na Raspberry Pi ao essencial.

Quando um carro cadastrado tem acesso pela primeira vez no sistema este irá guardar uma quantidade de imagens, as mais descorrelacionadas entre si, e a cada novo acesso irá confrontar essas imagens com o carro presente, para verificar, por meio de técnicas de auto correlação, filtragem, espectro se mesmo o está de acordo com o banco de dados, dada uma margem de erro. Isso concerne ao sistema uma pequena robustez contra plágios de placas.

A página web permite que o usuário insira um novo carro, ou proíba sua entrada, a qualquer momento. Esta também provê acesso ao banco de dados, onde o usuário ou a empresa/firma pode conferir seu histórico de entrada/saída do veículo, ou ver se há recorrência de um carro sem permissão querendo ter acesso.

VI. REVISÃO BIBLIOGRÁFICA

Reconhecimento de placas de automóveis através do uso de processamento digital de imagens é uma técnica amplamente usada em todo mundo por departamentos de trânsito para fiscalização de descumprimento de regras de tráfego. Devido a essa demanda, muito algoritmos, bibliotecas, filtros e classificadores foram desenvolvidos para serem aplicadas nessa técnica, como pode ser visto em [1] e [2].

Trabalhos de reconhecimento de placa, em vários lugares no mundo, têm usado a biblioteca openALPR (Automatic License Plate Recognition), que é utilizada especificamente para identificar placas de veículos em frames de imagem e vídeo. A openALPR é estruturada em cima de outras bibliotecas, sendo estas, openCV (visão computacional), Tesseract (reconhecimento de caractere) e Leptonica (processamento e análise de imagens) [3], [4] e [5].

De acordo com [6] e [7] o procedimento usado para se fazer o reconhecimento de placa é, em geral:

- Aquisição da imagem com a placa;
- Processamento da imagem para melhoria da qualidade;
- Rastreamento da placa na imagem;
- Identificação dos segmentos de caractere na placa rastreada;
- Reconhecimento de caractere;
- Análise de padrão do caractere encontrado.

A Figura 1 apresenta um exemplo da rotina de processamentos descrita acima.

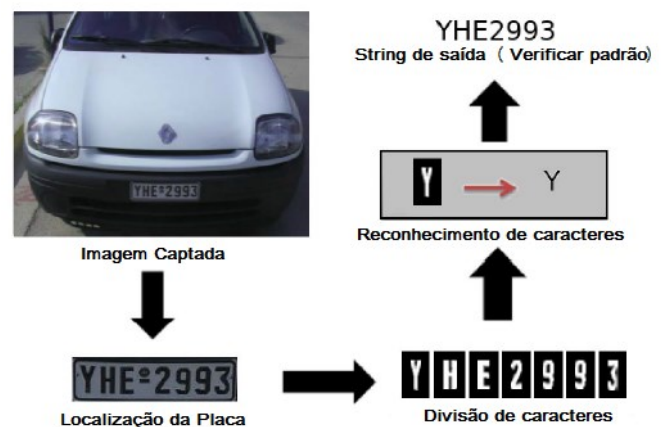


Figura 1. Etapas do processamento para reconhecimento de placa (Figura adaptada de [8]).

Algumas das dificuldades encontradas nessa tarefa são a aquisição de imagens desfocadas, má iluminação, baixo contraste, presença de ruído e uso de diferentes fontes em uma mesma placa [8]. O posicionamento da câmera que capta as imagens das placas também tem grande influência nos resultados e qualidade das imagens obtidas, e por consequência, na velocidade e dificuldade de convergência de um resultado [9].

VII. DESENVOLVIMENTO

O sistema foi pensado de forma sempre esperar um veículo se aproximar da cancela, sua placa é checada no banco de dados e então, caso esteja cadastrado, sua entrada é autorizada.

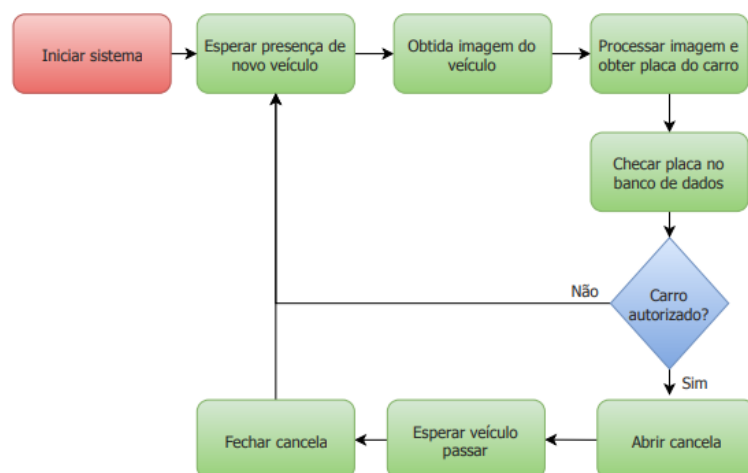


Figura 2. Fluxograma do Sistema de controle

A biblioteca openALPR foi baixada e compilada na placa Raspberry Pi 3, junto com suas dependências para o funcionamento, sendo o openCV, Tesseract e Leptonica.

O código “em_alpr.cpp” é um pequeno trecho do uso a openALPR em C++, onde pelo argumento na chamada da função *main* recebe o arquivo imagem para obter a placa do veículo.

Quanto à parte mecânica foi utilizado um motor de passo, controlado por um pequeno script em *Python*, que apenas gira 90° no sentido anti-horário para abrir a cancela e 90° no sentido horário para fechar a cancela.

Por fim a parte do sensor de presença, foi utilizado um sensor ultrassônico, também controlado em *Python*, que ao detectar um objeto de proximidade menor ou igual a 20cm, seria tirado uma fotografia com a webcam para ser realizado o processamento pelo openALPR. A Figura 3 apresenta uma visão geral do hardware do sistema.

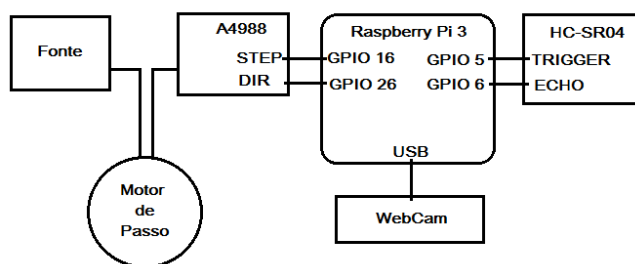


Figura 3. Hardware do sistema.

VIII. RESULTADOS

A eletrônica do sistema foi confeccionada em uma placa de circuito impresso. O esquemático e layout estão presentes na pasta do projeto. O circuito conta com uma fonte 12 V e o driver para o motor de passo.

A implementação do sistema é apresentada nas Figs. 4 e 5. O sistema foi embarcado em uma caixa patola, dentro da qual foram colocados o transformador, fonte DC, driver e motor de passo. A cancela para testes foi feita com papel panamá, acoplada com no eixo do motor com um suporte feito com impressão 3D. Um contrapeso foi colocado na cancela para contra balancear o momento feito pela sua extensão. Na cancela foram colocados a câmera e o ultrassom.

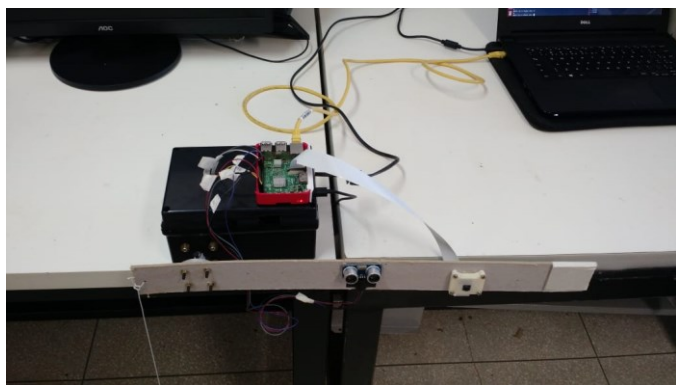


Figura 4. Vista superior do Sistema construído.

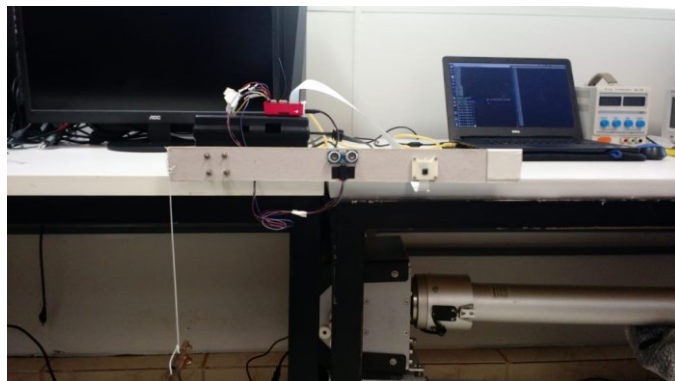


Figura 5. Vista frontal do Sistema construído.

O processamento da placa do veículo através do openALPR se mostrou bastante eficaz, conseguindo uma boa precisão para a placa do carro, contudo é necessário considerar um bom posicionamento da câmera para obter resultados autênticos à placa.

```

pi@raspberrypi:~$ ./em_alpr bmw-328i.jpg
Placa: FMW3777
Placa: FMH3777
Placa: FMW777
Placa: FMH777
Placa: FMWB777
Placa: FMN3777
  
```

Figura 6. Processamento da placa do carro e resultados obtidos.



Figura 7. Imagem do veículo utilizado na Figura 6.

Tempo de processamento da imagem, foi aceitável, considerando o tamanho da imagem, também é um tempo dentro dos padrões de espera para a autorização de acesso ao local desejado.

```

real 0m1.193s
user 0m1.122s
sys 0m0.071s
  
```

Entretanto, alguns casos haviam troca de caracteres da placa do veículo como mostram as figuras 8 e 9.


```
pi@raspberrypi:~$ ./em_alpr Adulterar_a_placa_de_veiculo.jpg
Placa: MHE58B5
Placa: MHE58B5
Placa: SMHE58B5
Placa: ME58B5
Placa: FMHE58B5
```

Figura 8. Placa de carro com troca de alguns caracteres



Figura 9. Placa de carro utilizado na figura 6

Apesar de alguns caracteres terem sido trocados por outros, podem ser corrigidos com posicionamento da câmera e algoritmo para substituição dos caracteres. Como trocar “O” por “0” ou “I” por 1.



Figura 10. Placa de carro trocado “0” por “O”.

```
pi@raspberrypi:~$ alpr -c br Chevrolet-ClassicAV1555_960_640.jpg
plate0: 10 results
- ENW9720 confidence: 92.7351
- ENW9720 confidence: 90.1073
- ENH9720 confidence: 89.7887
```

Figura 11. Leitura da placa do carro da figura 8.

Na figura 11 o processamento da placa do carro da figura 10 resultou em um erro no último caractere, trocando “0” por “O”. Entretanto esse problema foi resolvido com o algoritmo em C++ para realizar a substituição de letras nos lugares de números e vice-versa.

```
Tirando foto do veiculo
Placa: ENW9720
Placa: ENW9720
Placa: ENH9720
```

-

Find Symb... VisualGDB... SSH Cons...

Figura 12. Leitura da placa do carro da figura 8 corrigindo caractere

A figura 10 mostra então o algoritmo solucionando o problema do último caractere. Fazendo assim a leitura correta da placa do veículo.

Outra parte desenvolvida foi o banco de dados, nesse caso, considerando a simplicidade e processamento do Raspberry Pi 3, foi utilizado um sistema de arquivos.

Para armazenar as placas dos carros cadastrados, é criado uma pasta onde irá conter todos os veículos. Logo, para verificar se o veículo está cadastrado é apenas verificado se existe um arquivo cujo o nome é a própria placa do veículo.

```
pi@raspberrypi:~$ cd alpr/
pi@raspberrypi:~/alpr$ ls
AAA1234 FMW3777 JGM2249 JHT1346 PUM4594
```

Figura 13. Banco de dados do sistema

Como mostra a figura 13, o banco de dados conta com duas placas cadastradas, “AAA1234” e “FMW3777”. Qualquer outro veículo com outra placa, o sistema irá recusar e não irá abrir a cancela para a entrada do carro.

A verificação é feita a partir da função “ExisteCarro”, no arquivo “DbManager.h” e implementado no “DbManager.cpp”.

O sistema mecânico utilizou-se do sensor ultrassom, onde caso a distância medida fosse menor que 10 cm, significava que um carro havia chegado ao local, então era capturada uma foto da placa do carro para então ser feito a análise.

```
Tirando foto do veiculo
Placa: FMW3777
Placa encontrada, abrindo cancela
Esperando carro passar
```

-

Find Symb... VisualGDB... SSH Cons...

Figura 14. Reconhecendo uma placa cadastrada

Se o veículo estivesse cadastrado no banco de dados, o motor de passo abria a cancela, esperava até que o sensor

ultrassom detectasse que o carro havia passado e então fechava a cancela, como mostra a figura 14.

Caso o veículo não estivesse cadastrado, era feito mostrado uma mensagem no terminal que o veículo não estava cadastrado.

O sistema conta com uma aplicação web rodando no próprio Raspberry, onde o usuário pode interagir com o sistema a partir de opções disponíveis na página.

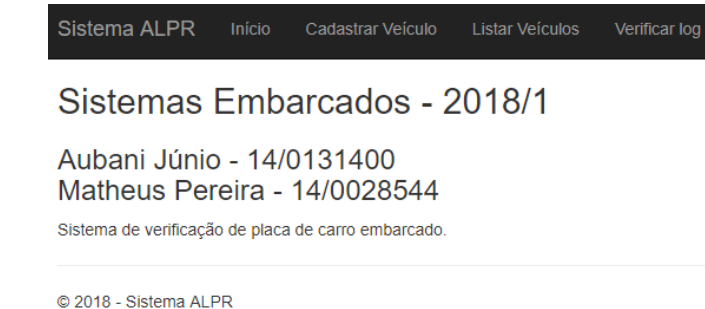


Figura 15. Aplicação web – Página inicial

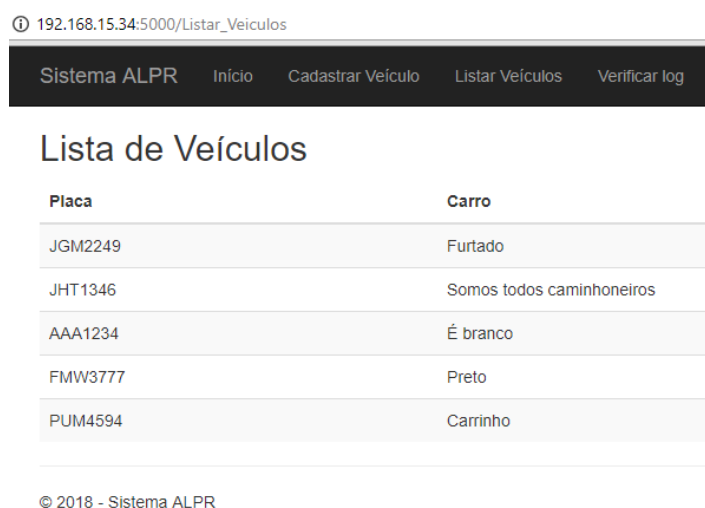


Figura 16. Aplicação web - Veículos



Figura 17. Aplicação web - Cadastrar

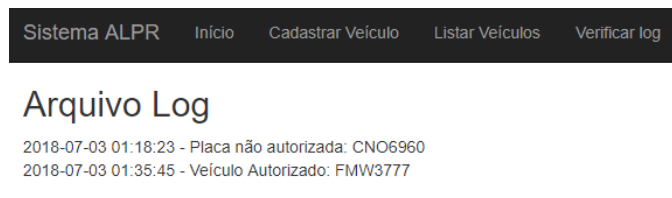


Figura 18. Aplicação web - Log

A página disponibiliza opções para cadastrar e listar veículos, além de mostrar os logs dos veículos que tiveram acesso ao ambiente usando o sistema de reconhecimento de placa.

O sistema de Log, funciona em arquivo, onde é criado um .txt contendo todo o log do sistema que a página carrega e exibe.

Para a aplicação web foi utilizado o .NET Core 2.0 com linguagem ASP.NET, rodando na porta 5000, podendo ser acessado a partir de qualquer dispositivo conectado na mesma rede do Raspberry Pi.

REFERÊNCIAS

- [1] O. Martinsky, "Algorithmic and mathematical principles of automatic number plate recognition systems," B. Sc. Thesis, Department of Intelligent Systems, Brno University of Technology, 2007.
- [2] S. Draghici, "A neural network based artificial vision system for licence plate recognition," Department of Computer Science, Wayne State University, 1997.
- [3] S. Prabhu, S. Kalambur and D. Sitaram, "Recognition of Indian License Plate Number from Live Stream Videos," International Conference in Computing, Communications and Informatics (ICACCI), 2017.
- [4] L.O. Tjandra, S. Nugroho and D. Utomo, "Eletronic Road Pricing System Prototype," International Seminar on Application for Technology of Information and Communication, 2016.
- [5] J. Mistry, A.K. Misraa, M. Agarwal, A. Vyas, V.M. Chudasama and K.P. Upla "Na Automatic Detection of Helmeted and Non-helmeted Motorcyclist with License Plate Extraction using Convolutional Neural Network," Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), 2017.
- [6] H. Kwasnicka and B. Wawrzyniak, "License plate localization and recognition in camera pictures," Faculty Division of Computer Science, Wroclaw University of Technology, 2002.
- [7] S. Du, M. Ibrahim, M. Shehata and W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-Art Review," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 23, NO. 2, FEBRUARY, 2013.
- [8] A. PP and P.P. Thulasidharan, "Automatic License Plate Recognition (ALPR) using HTM Cortical Learning Algorithm," International Conference on Intelligent Computing and Control, 2017.
- [9] A. Makarov, V. Lukic and M. Spanovic, "Deriving Camera and Lens Settings for Fixed Traffic Enforcement and ALPR Cameras," 20th Telecommunications forum TELFLOR, 2012.