

Sistema de Gerenciamento de Garagens ANPR

Charles Santos, Felipe Mello, Paulo Dimas, Rafael Barbosa, Talita Justo

December 16, 2024

1 Introdução

O reconhecimento automático de placas veiculares (ANPR - Automatic Number Plate Recognition) tem revolucionado o controle de acesso e segurança em ambientes urbanos. Com o avanço da visão computacional e Internet das Coisas (IoT), a automatização destes sistemas tornou-se mais acessível e eficiente. Neste contexto, este trabalho propõe um sistema integrado de gerenciamento de garagens que combina reconhecimento de placas veiculares com automação IoT.

O sistema utiliza técnicas de reconhecimento de imagens para identificar as placas dos veículos e integra dispositivos de automação para gerenciar o controle de acesso e a ocupação das vagas. O projeto enfrenta desafios como a precisão de modelos de detecção "on edge" (processamento local no dispositivo, como o Raspberry Pi), e a comunicação entre dispositivos IoT com o protocolo MQTT.

2 Objetivos

2.1 Objetivo Geral

Desenvolver um Sistema de Gerenciamento de Garagem automatizado, implementando tecnologias de ANPR utilizando Raspberry Pi e modelos de computação visual para extrair os caracteres da placa. Integrando também dispositivos físicos (sensores e atuadores) em conjunto com softwares, para melhorar a eficiência e a segurança no controle de estacionamentos.

2.2 Objetivos Específicos

- Implementar o reconhecimento de placas:

- Aplicar modelos de detecção de objetos para identificar placas em imagens capturadas.
- Integrar tecnologia de OCR (Reconhecimento Ótico de Caracteres) para a extração eficiente do texto das placas.
- Aplicar conceitos de IoT:
 - Utilizar sensores de proximidade (infravermelho) e LEDs para controlar a ocupação da garagem.
 - Receber as informações dos sensores através do protocolo MQTT.
- Utilizar uma API de placas de trânsito:
 - Enviar os dados das placas, capturados via MQTT, para uma API que retorne informações detalhadas do veículo, como cor, modelo, ano, etc.
 - Integrar as informações retornadas pela API ao banco de dados.
- Analisar a eficiência do ANPR no gerenciamento de estacionamentos:
 - Avaliar a relevância do reconhecimento automático de placas no controle de fluxo.
 - Identificar problemas ou melhorias necessárias em processos de automação de estacionamentos.

3 Motivação

A proposta de desenvolver um sistema que integra visão computacional com IoT oferece uma oportunidade única de trabalhar com diversas tecnologias modernas simultaneamente, desde processamento de imagens até comunicação entre dispositivos. O interesse particular em visão computacional surge da crescente aplicação desta tecnologia em soluções do dia a dia, e este projeto permite explorar essa área de forma prática, especialmente no contexto de reconhecimento de placas veiculares.

A experiência hands-on com diferentes componentes de hardware (como Raspberry Pi, ESP32 e câmeras) e software (APIs de visão computacional, protocolos de comunicação e bancos de dados) proporciona um aprendizado multidisciplinar valioso, abrangendo tanto o desenvolvimento de software quanto a configuração e integração de sistemas embarcados.

4 Projeto de Hardware

Desenvolvemos uma maquete para simular o funcionamento de uma garagem automatizada.

4.1 Arquitetura do sistema

O sistema foi projetado utilizando o microcontrolador ESP32 como unidade de controle central, conectado com três sensores infravermelhos, três LEDs RGB e um micro servo motor. A montagem foi realizada em duas protoboards utilizando jumpers para as conexões para que não fosse necessária uma placa de circuito impressa dedicada. O Raspberry Pi utilizado estava rodando headless, ou seja, operando sem monitor, teclado ou mouse conectados via SSH.

Construímos a maquete utilizando papel paraná, palito de picolé, cola quente e tinta amarela. Posicionamos os sensores infravermelhos em cada vaga para que fossem detectados os carros que estacionassem e os LEDs foram colocados em cima das vagas para identificar se aquela vaga estava disponível ou não. O micro servo foi fixado na entrada da garagem para simular uma catraca, abrindo e fechando conforme um carro fosse detectado.

Rodar dois modelos de processamento de imagens localmente exigiria um SBC mais completo. Por isso, componente fundamental no projeto foi o Raspberry Pi 5, com um processador Broadcom BCM2712 2.4GHz quad-core 64-bit Arm Cortex-A76. O Raspberry Pi oferece maior capacidade computacional e versatilidade por ser um computador single-board(SBC) completo, com sistema operacional Linux embarcado, enquanto microcontroladores como Arduino e ESP32 são limitados a execução direta de firmware com recursos computacionais mais restritos.

4.2 Projeto do Circuito

O circuito foi projetado para integrar os componentes necessários para a maquete funcionar. O ESP32 foi conectado às protoboards para controlar os periféricos conectados. Os sensores infravermelhos e os LEDs foram conectados aos pinos do ESP para o monitoramento das vagas. O servo motor foi conectado a um pino que realizava a abertura ou fechamento da catraca baseado em mensagens MQTT.

4.3 Prototipagem e Testes

O circuito foi montado em protoboards, utilizando jumpers para a ligação dos componentes e resistores para que os componentes não queimassem. Foram realizados diferentes testes:

1. Sensores infravermelhos: Detecção de carros (miniaturas) nas vagas da maquete
2. LEDs RGB: Teste de controle para representar vagas ocupadas, vazias ou reservadas
3. Micro Servo Motor: Simulação do movimento de abertura e fechamento da catraca

Após os testes individuais, foi realizado um teste com todo o sistema funcionando, demonstrando a interação coordenada entre os sensores (infravermelho) e os atuadores (LEDs e Servo Motor), simulando assim uma garagem automatizada.

5 Projeto de Software

5.1 Geral

O sistema de gerenciamento de garagens ANPR foi desenvolvido utilizando uma arquitetura modular e distribuída, permitindo a integração eficiente entre os componentes de hardware e software. A solução implementada combina tecnologias de visão computacional para reconhecimento de placas veiculares com um sistema de controle de acesso automatizado. O sistema foi desenvolvido em Python devido à sua extensa biblioteca de visão computacional (OpenCV), integração nativa com frameworks de aprendizado de máquina e amplo suporte para desenvolvimento IoT em dispositivos como Raspberry Pi.

5.2 Modelo de Detecção de Objetos

O dataset utilizado para treinar o modelo YOLOv10 foi construído a partir de aproximadamente 300 imagens coletadas de repositórios públicos como Roboflow e Open Images Dataset, focando especificamente em placas veiculares nos padrões brasileiro antigo e Mercosul.

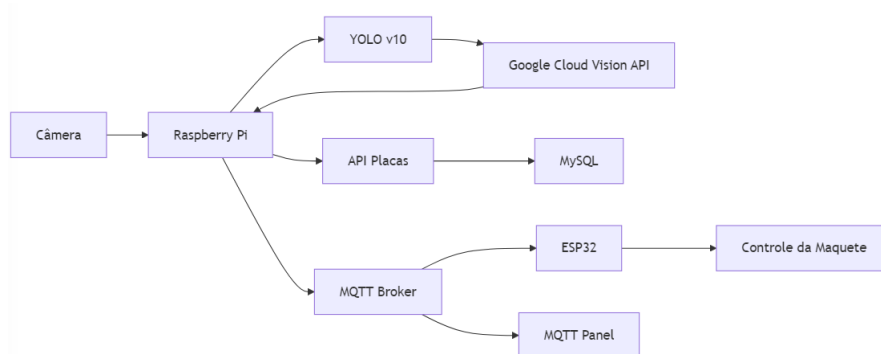


Figure 1: Fluxograma representando o processo completo.

O processo de preparação do dataset incluiu a remoção de imagens duplicadas e renomeação sistemática dos arquivos. A anotação das imagens foi realizada na plataforma CVAT utilizando Bounding Boxes para delimitar as placas, com posterior exportação no formato YOLO compatível com a biblioteca Ultralytics, sendo o conjunto de dados dividido em 80% para treinamento e 20% para validação.

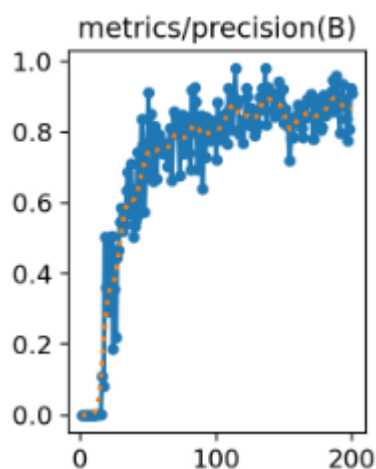


Figure 2: Gráfico de precisão do modelo ao longo do treinamento

5.3 Modelo de Detecção de Texto

Para o reconhecimento óptico de caracteres (OCR) das placas veiculares, foi implementada a Google Cloud Vision API. Antes do processamento pela API, as imagens das placas detectadas passam por um

pipeline de pré-processamento que inclui o recorte da região da placa, conversão para escala de cinza (greyscale), equalização de histograma para aumento de contraste e aplicação de threshold inverso, otimizando assim a precisão do reconhecimento de caracteres.

5.4 Controlador ESP-32

O microcontrolador ESP-32 foi implementado para controlar uma maquete que simula o ambiente real de um estacionamento. O dispositivo gerencia o sistema de catracas e monitora o status das vagas disponíveis, recebendo comandos via protocolo MQTT para controle remoto das funcionalidades e enviando mensagens para atualizar o Painel de Controle.

5.5 Criação e Inserção do Banco de Dados

O sistema utiliza um banco de dados MySQL para armazenamento e gerenciamento de informações relacionadas à gestão de mensalistas, garagens, carros e fluxo de veículos. A estrutura do banco incorpora dados obtidos através da API Placas, incluindo informações detalhadas como marca, modelo, cor e ano dos veículos, além de registrar o histórico de fluxo e associações com garagens específicas.

5.6 Protocolo MQTT

O protocolo MQTT atua como backbone da comunicação entre os componentes do sistema, estabelecendo três principais fluxos de dados: a conexão entre Raspberry Pi e banco de dados para atualização de registros, a comunicação entre Raspberry Pi e ESP32 para controle automatizado da catraca, e a interface com o MQTT Panel para monitoramento remoto do sistema.

6 Custo do Projeto

7 Conclusão

O Sistema de Gerenciamento de Garagens ANPR desenvolvido demonstrou ser uma solução eficaz para a automatização do controle de acesso

Item	Quantidade	Valor
Raspberry Pi e seus componentes (como câmera)	1	R\$800,00
Esp 32 com módulo Wifi	1	R\$80,00
Sensores infravermelho	3	R\$21,00
Leds + jumpers	Conjunto	R\$10,00
Protoboard 400 pontos	1	R\$9,00
Micro servo 9g SG90	1	R\$15,90
Componentes para maquete	Conjunto	R\$10,90
API de placas de trânsito	1	R\$40,00
		Total: 986,80

Figure 3: Estimativa dos custos do projeto, incluindo hardware, software e serviços

veicular, integrando com sucesso tecnologias de visão computacional e Internet das Coisas. A implementação do modelo YOLOv10 para detecção de placas apresentou resultados notáveis, sendo capaz de identificar placas veiculares em tempo real e mantendo alta precisão em diferentes condições de iluminação. O sistema demonstrou versatilidade ao processar tanto placas no padrão Mercosul quanto no padrão antigo brasileiro.

Um dos desafios técnicos mais significativos enfrentados foi durante a migração do sistema para o Raspberry Pi. A transição de um ambiente de desenvolvimento desktop Windows com processador x86 para a arquitetura ARM do Raspberry Pi exigiu uma readaptação importante na abordagem de OCR. A solução inicial que utilizava EasyOCR para processamento local foi substituída pela Google Cloud Vision API, demonstrando a importância da flexibilidade no desenvolvimento e a necessidade de adaptação às limitações de hardware. Esta mudança, embora desafiadora, resultou em um sistema mais robusto para extração de caracteres, ainda que mais sensível às condições de iluminação quando comparado ao módulo de detecção de placas.

Este projeto não apenas atingiu seus objetivos técnicos iniciais, mas também proporcionou importantes aprendizados sobre a implementação prática de sistemas de visão computacional em dispositivos embarcados. A experiência adquirida na superação dos desafios de migração de plataforma e otimização de recursos demonstra a importância de considerar as limitações de hardware desde as fases iniciais do desenvolvimento.

Referências

- CHOUDHAURY, S. R. et al. *Chaurah: A Smart Raspberry Pi based Parking System*. arXiv.org, 2023.
- FAKHAR, S. et al. *Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi*. International Journal of Electrical and Computer Engineering, v. 9, n. 3, p. 1805-1813.
- FERNANDEZ, C. J. et al. *Raspberry Pi based ANPR for Smart Access*. International Conference on Artificial Intelligence in Engineering and Technology, 2021.
- JENITTA, J. et al. *License Plate Recognition Using Raspberry Pi*. World Conference on Communication & Computing (WCONF), 2023.
- *Automatic Number Plate Recognition (ANPR)*. International Journal of Innovations in Engineering and Science, 2023.