

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

GABRIEL VINÍCIUS RAFFAELLY

IMPLEMENTAÇÃO DE UM SISTEMA DE RECONHECIMENTO
AUTOMÁTICO DE PLACAS DE VEÍCULOS UTILIZANDO VISÃO
COMPUTACIONAL

DOIS VIZINHOS

2025

GABRIEL VINÍCIUS RAFFAELLY

IMPLEMENTAÇÃO DE UM SISTEMA DE RECONHECIMENTO
AUTOMÁTICO DE PLACAS DE VEÍCULOS UTILIZANDO VISÃO
COMPUTACIONAL

**Implementation of an automatic vehicle license plate recognition system
using computer vision and OCR**

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito parcial para a
obtenção do título de Bacharel em Engenharia
de Software, Universidade Tecnológica Federal
do Paraná (UTFPR).
Orientador(a): Marlon Marcon

DOIS VIZINHOS

2025



4.0 Internacional

Esta licença permite remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

GABRIEL VINÍCIUS RAFFAELLY

IMPLEMENTAÇÃO DE UM SISTEMA DE RECONHECIMENTO
AUTOMÁTICO DE PLACAS DE VEÍCULOS UTILIZANDO VISÃO
COMPUTACIONAL

Trabalho de Conclusão de Curso de Graduação
apresentado como requisito para obtenção do
título de Bacharel em Engenharia de Software,
Universidade Tecnológica Federal do Paraná
(UTFPR).

Data de aprovação: 08 de Dezembro de 2025

Marlon Marcon
Doutorado
Universidade Tecnológica Federal do Paraná ,

Luiz Fernando Puttow Southier
Doutorado
Universidade Tecnológica Federal do Paraná

Rene Pomilio de Oliveira
Mestrado
Universidade Tecnológica Federal do Paraná

DOIS VIZINHOS

2025

RESUMO

A visão computacional é uma tecnologia versátil que se aplica em várias áreas, como reconhecimento facial, identificação de objetos em movimento e estáticos, bem como a detecção de caracteres em imagens. Neste estudo, o objetivo foi empregar essa tecnologia para reconhecer placas de veículos, visando controlar o acesso de automóveis e verificar se estavam registrados no sistema, por meio das placas dentro da Universidade Tecnológica Federal do Paraná, campus Dois Vizinhos. Tal problema foi dividido em quatro etapas: captura de imagem do veículo, detecção da placa, segmentação de caracteres e reconhecimento de caracteres.

Palavras-chave: reconhecimento automático de placas veiculares, visão computacional, ocr, processamento de imagens, controle de acesso.

ABSTRACT

Computer vision is a versatile technology that can be applied in various fields, such as facial recognition, identification of moving and static objects, as well as character detection in images. In this study, the objective was to employ this technology to recognize vehicle license plates, aiming to control the access of automobiles and verify whether they are registered in the system through their plates within the Federal University of Technology – Paraná, Dois Vizinhos campus. This problem was divided into four stages: vehicle image capture, license plate detection, character segmentation, and character recognition.

Keywords: automatic license plate recognition, computer vision, ocr, image processing, access control.

SUMÁRIO

1	Introdução	4
2	Trabalhos Relacionados	5
3	Metodologia	5
3.1	Captura de Imagens.....	6
3.2	Deteção da Placa	6
3.3	Segmentação de Caracteres.....	7
3.4	Reconhecimento de Caracteres	7
3.5	Busca pela Placa.....	8
3.6	Consulta via REST API.....	8
3.7	Identificação do Condutor	8
4	Resultados: Prova de Conceito (Poc).....	8
5	Conclusão	13

Implementação de um Sistema de Reconhecimento Automático de Placas de Veículos Utilizando Visão Computacional

GABRIEL VINÍCIUS RAFFAELLY

¹UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Campus Dois Vizinhos

`gabrielraffaelly@alunos.utfpr.edu.br`

Abstract. *Computer vision is a versatile technology that can be applied in various fields, such as facial recognition, identification of moving and static objects, as well as character detection in images. In this study, the objective was to employ this technology to recognize vehicle license plates, aiming to control the access of automobiles and verify whether they are registered in the system through their plates within the Federal University of Technology – Paraná, Dois Vizinhos campus. This problem was divided into four stages: vehicle image capture, license plate detection, character segmentation, and character recognition.*

Resumo. *A visão computacional é uma tecnologia versátil que se aplica em várias áreas, como reconhecimento facial, identificação de objetos em movimento e estáticos, bem como a detecção de caracteres em imagens. Neste estudo, o objetivo foi empregar essa tecnologia para reconhecer placas de veículos, visando controlar o acesso de automóveis e verificar se estavam registrados no sistema, por meio das placas dentro da Universidade Tecnológica Federal do Paraná, campus Dois Vizinhos. Tal problema foi dividido em quatro etapas: captura de imagem do veículo, detecção da placa, segmentação de caracteres e reconhecimento de caracteres.*

1. Introdução

O aumento do número de veículos nas cidades tem gerado a necessidade de sistemas eficientes e ágeis para o reconhecimento de placas de automóveis. Por meio de técnicas de Processamento Digital de Imagens (PDI), como demonstrado por [Guingo et al. 2002], tornou-se mais fácil o desenvolvimento prático de sistemas que não apenas identificam placas, mas também, que reconhecem veículos suspeitos e possíveis infratores. Além disso, essas tecnologias podem ser aplicadas para gerenciar o acesso a locais restritos, como estacionamentos, condomínios ou propriedades privadas.

Atualmente, a visão computacional, aliada à programação em Python [Labaki and Woiski 2003], tem sido aplicada em diversas áreas, incluindo reconhecimento de pessoas, identificação de padrões e reconhecimento de caracteres. Um software que consegue ler placas de veículos pode ser empregado em diversos lugares, para monitorar a entrada e saída dos automóveis, além de servir ao governo para multar motoristas que cometem infrações.

O aumento do fluxo de veículos nas cidades gera demanda pela automação no controle do tráfego, além de soluções eficientes para controle de infrações e melhorias na segurança. Utilizando sistemas automatizados, como a tecnologia de reconhecimento automático de placas de veículos (do inglês *Automatic License Plate Recognition*, ou ALPR), permite-se uma fiscalização mais rápida e precisa e, assim, reduz a necessidade de intervenções manuais.

O *Optical Character Recognition*, ou OCR, é uma tecnologia que permite converter diferentes tipos de documentos, como imagens digitalizadas de textos impressos ou manuscritos, em dados editáveis e pesquisáveis. No contexto de leitura de placas, o OCR é habilmente utilizado para identificar e digitalizar os números e letras presentes nas placas de veículos, transformando-os em textos legíveis por sistemas eletrônicos.

O presente trabalho tem como objetivo apresentar o desenvolvimento e a avaliação de um sistema de reconhecimento automático de placas de veículos, aplicado como prova de conceito no controle de acesso veicular da Universidade Tecnológica Federal do Paraná (UTFPR), campus Dois Vizinhos. Tal sistema permitiu identificar as placas, consultar os veículos em uma base de dados e verificar se esses veículos pertenciam a servidores, estudantes ou visitantes do campus.

2. Trabalhos Relacionados

A Visão Computacional é uma área da Inteligência Artificial que possibilita com que máquinas automaticamente reconheçam imagens e as descrevam com precisão e eficiência. Atualmente, os computadores têm acesso a um grande volume de imagens e vídeos criados por *smartphones*, câmeras de trânsito, sistemas de segurança, entre outros. As aplicações usam Inteligência Artificial e *Machine Learning* (IA/ML) para o processamento de dados, visando reconhecimento e identificação de objetos que vão desde reconhecimento facial, até análise de movimentos e reconhecimento de veículos e suas placas.

No reconhecimento de placas, a visão computacional é usada para detectar a localização das placas nos automóveis, enviando as imagens para o OCR (Reconhecimento de Caracteres Óptico), que converte o texto em placas para dados digitais. Em seguida, utiliza-se a biblioteca OpenCV, que foi desenvolvida pela Intel e está disponível na Internet de forma gratuita [Marengoni and Stringhini 2009]. Ela é responsável pelo reconhecimento e pré-processamento das imagens e esse pré-processamento resolve o ajuste de contraste, remoção de ruído e foco nas áreas de interesse.

Após o pré-processamento, é utilizado a arquitetura YOLO (*You Only Look Once*), que é uma rede neural programada para detectar objetos em tempo real, responsável por localizar precisamente a placa do veículo na imagem. Sua principal característica é a capacidade de realizar múltiplas detecções de objetos por meio de uma imagem extremamente rápida e eficiente. O Tesseract OCR, é uma biblioteca de código aberto, que foi desenvolvida pelo Google [Rosebrock 2017], e que realiza a conversão do texto das placas em formato digital. O Tesseract analisa as imagens e identifica os caracteres, permitindo assim, que o sistema classifique e armazene essas informações. A combinação da OpenCV, YOLO e do Tesseract OCR possibilita um fluxo de trabalho sólido, onde a OpenCV se concentra na preparação da imagem, o YOLO realiza a detecção precisa da placa e o Tesseract a extração dos dados textuais, essenciais para o funcionamento do sistema.

Utiliza-se um *dataset* público, o RodoSol-ALPR Dataset, que contém 5000 imagens capturadas por câmeras estáticas, imagens de carros com placas antigas e placas Mercosul. Todas as imagens são em resolução de 1280 x 720 pixels. Como demonstrado na Figura 1

3. Metodologia

No contexto da Universidade Tecnológica Federal do Paraná (UTFPR), campus Dois Vizinhos, o controle de acesso de veículos é realizado de forma manual ou sem mecanismos automatizados de identificação, o que pode ocasionar dificuldades na verificação rápida de veículos

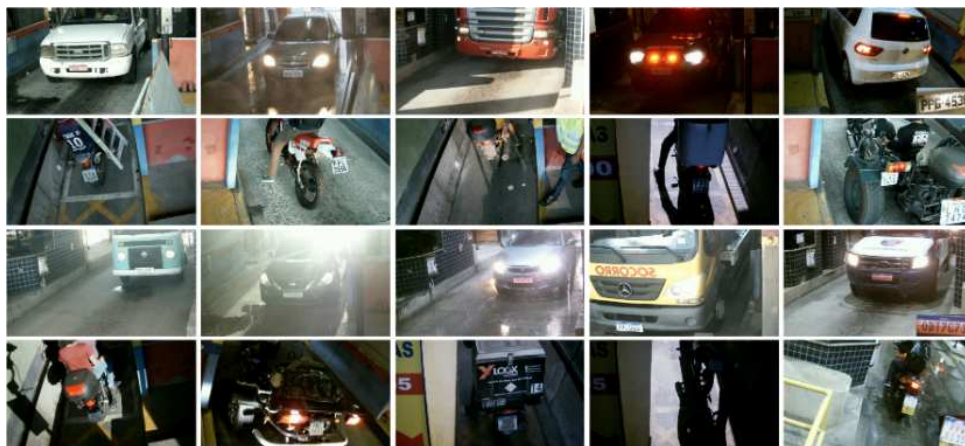


Figure 1. Na região inferior direita de cada uma das imagens na última coluna, pode ser visualizado como ficaram os layouts das placas no artigo de. [Laroca et al. 2022]

pertencentes a alunos, professores e servidores, além de permitir a entrada de veículos não autorizados. Esse cenário evidencia a necessidade de soluções tecnológicas capazes de automatizar o processo de identificação veicular, contribuindo para a segurança, organização e eficiência no fluxo de automóveis dentro do campus universitário.

Para a aplicação desse projeto, primeiramente foi realizado o reconhecimento das imagens dos veículos na entrada da UTFPR. Um vídeo ou uma foto do veículo são capturados, focando na placa. Em seguida, a imagem da placa é processada, utilizando visão computacional. Então, a arquitetura YOLO é utilizada para realizar a detecção da placa e logo após essa placa ficar isolada, aplica-se o OCR, por meio da ferramenta Tesseract, resultando em um reconhecimento de caracteres mais consistente. Por fim, o sistema consulta o banco de dados por uma API feita no Postman, executada no banco de dados, que analisa os detalhes e identifica se o veículo pertencia a um aluno ou professor registrado na UTFPR. Caso nenhum dos dois fosse reconhecido, o sistema não permite a entrada no campus, exibindo a mensagem de Acesso negado. Para desenvolvimento, foi utilizada a linguagem Python, bem como o OpenCV para processamento das imagens e Streamlit para a interface web do sistema. A Figura 2 apresenta o fluxo de processamento das imagens/vídeos e realiza a referência para as seções subsequentes do trabalho, que detalham os processos empregados em cada fase.

3.1. Captura de Imagens

A captura de imagens foi realizada por meio de vídeos e imagens gravados pelo autor na entrada do campus da UTFPR – Dois Vizinhos, bem como por imagens provenientes de um *dataset* público. Ressalta-se que não foram utilizados dados institucionais da universidade, nem houve acesso a câmeras oficiais de segurança, sendo todo o material coletado exclusivamente para fins experimentais.

Para o desenvolvimento da aplicação, foi utilizado um *framework* Python de código aberto, o Streamlit, que facilita a criação de aplicações web, permitindo o envio de imagens, vídeos ou a captura em tempo real por dispositivos externos conectados ao sistema. O uso combinado do *dataset* e dos vídeos experimentais possibilitou a análise e validação do desempenho do sistema em um ambiente próximo ao real.

3.2. Detecção da Placa

Essa etapa envolveu o uso de algoritmos de visão computacional, para localizar a placa na imagem de forma precisa, preparando-a para ter os caracteres reconhecidos. O objetivo foi isolar a

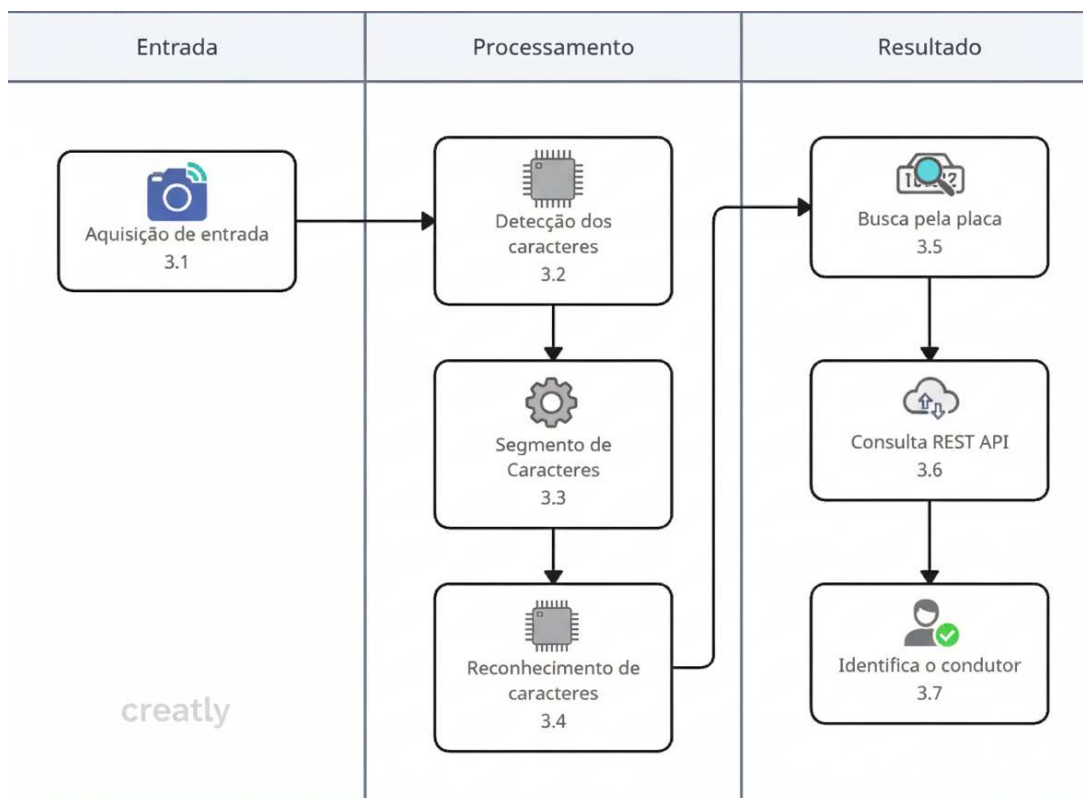


Figure 2. Processo de funcionamento do sistema. Fonte: Autoria Própria

placa para um processamento mais refinado. O OpenCV fez o pré-processamento, convertendo a imagem e, em seguida, O YOLOv4 realizou o processamento da imagem, retornando ao sistema apenas a imagem da placa, como demonstrado na Figura 3.



Figure 3. Detecção de diversas placas. [Laroca et al. 2022]

3.3. Segmentação de Caracteres

Conhecida como extração de caracteres, a segmentação consiste em um processo que tem como objetivo separar a placa como um bloco único de texto, estratégia que apresentou melhores resultados práticos em conjunto com o OCR utilizado.

3.4. Reconhecimento de Caracteres

De acordo com [Oliveira and Gonzaga 2012], o reconhecimento de caracteres é uma fase em que um objeto segmentado é analisado e associado a um único caractere alfanumérico. Essa etapa geralmente envolve o uso de redes neurais treinadas para identificar os caracteres. Como demonstrado na Figura 4

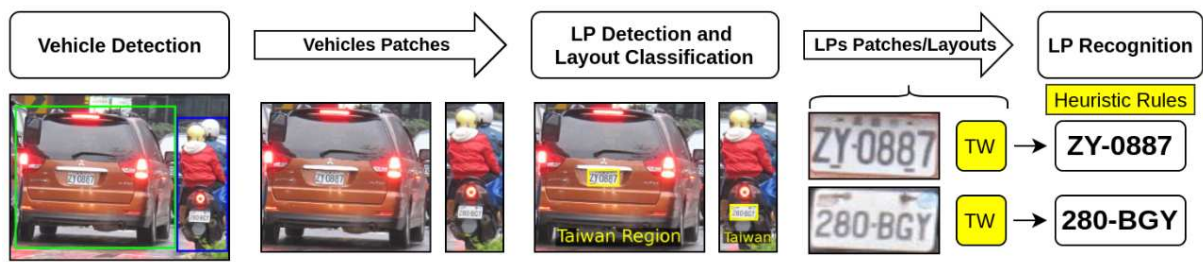


Figure 4. Processo de reconhecimento das placas. [Laroca et al. 2021]

3.5. Busca pela Placa

Após o reconhecimento dos caracteres pela ferramenta Tesseract, o sistema realizava a etapa de validação da placa identificada. Essa fase tinha como objetivo garantir a consistência dos resultados obtidos, eliminando possíveis ruídos ou leituras incorretas durante o processo de OCR. Para isso, a placa reconhecida era comparada com os dados registrados previamente no sistema, armazenados em formato estruturado (JSON). Essa busca servia para confirmar se o valor extraído correspondia a uma placa válida, além de assegurar que o sistema prosseguisse apenas com dados confiáveis.

3.6. Consulta via REST API

Com a placa validada, o sistema realizava uma requisição a uma API do tipo REST, desenvolvida para comunicação direta com o banco de dados. Essa API, implementada e testada no Postman, era responsável por verificar se o veículo estava devidamente cadastrado. A consulta retornava informações como o vínculo do veículo com a instituição (aluno, professor ou visitante).

3.7. Identificação do Condutor

Por fim, a etapa de identificação do condutor consistia em associar a placa consultada às informações do usuário, essa base de dados foi utilizada apenas para fins de validação da lógica de negócio do sistema. Dessa forma, o sistema conseguia determinar se o veículo pertencia a um aluno, professor ou visitante. Caso houvesse correspondência, o acesso era liberado automaticamente, exibindo a mensagem de “Acesso permitido” junto ao nome ou categoria do condutor identificado. Caso contrário, a mensagem exibida era “Acesso negado”, garantindo a segurança no controle de entrada. Essa etapa final reforçava a integração entre visão computacional, inteligência artificial e os sistemas de informação da universidade, consolidando o fluxo de automação do processo de controle de acesso.

4. Resultados: Prova de Conceito (Poc)

O código desenvolvido foi capaz de reconhecer as placas na entrada da Universidade, retornando ao sistema os condutores de cada veículo, utilizando técnicas de processamento de imagem e IA, identificando e segmentando a placa conforme a Figura 5.

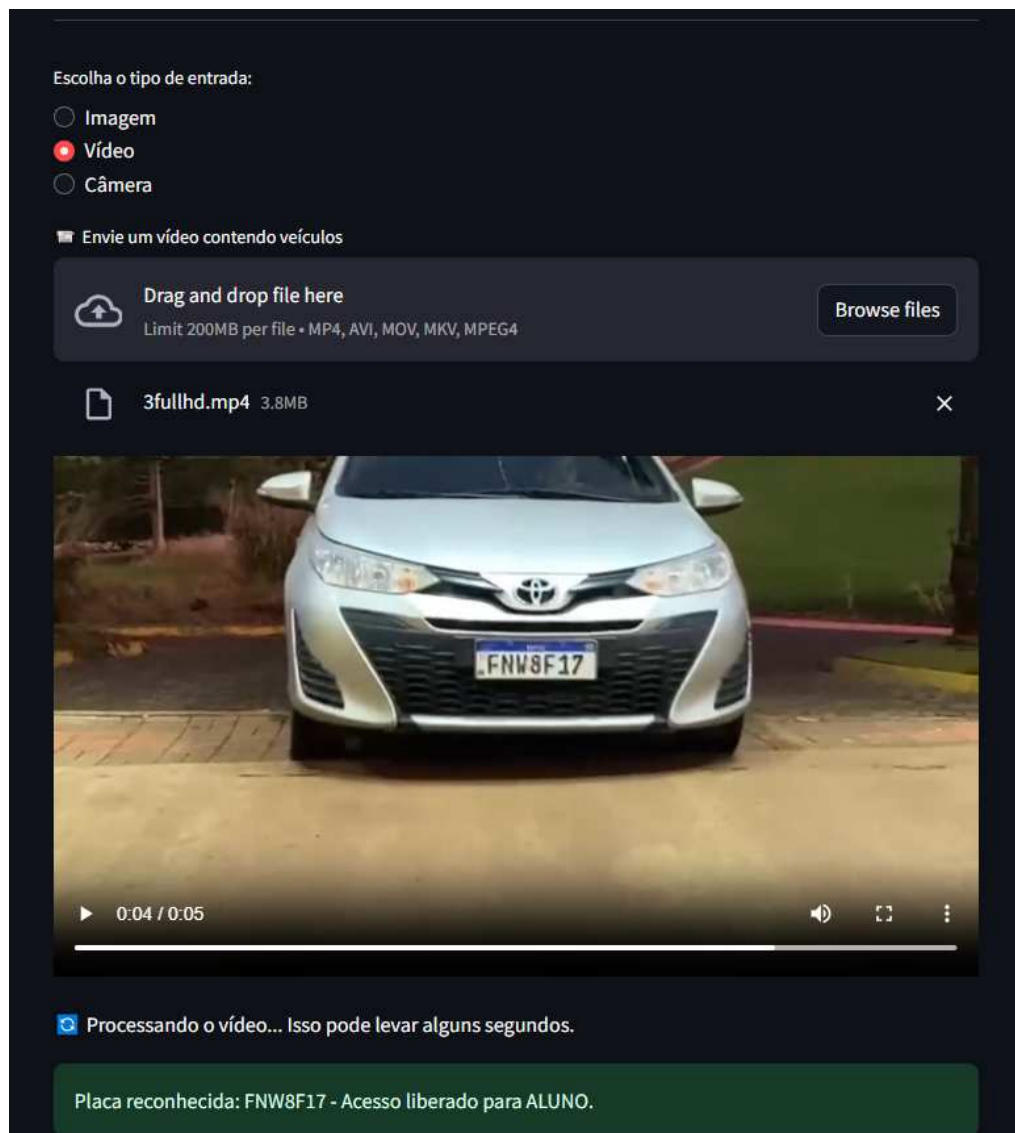


Figure 5. Tela do sistema funcionando com vídeo, na UTFPR-DV. (Do Autor, 2025)

Questionamento 1: O sistema é capaz de detectar veículos e localizar as placas em imagens ou vídeos?

Sim, o sistema foi projetado para detectar veículos e localizar placas veiculares em imagens e vídeos, utilizando a arquitetura YOLOv4 (*You Only Look Once*, versão 4). Essa versão do modelo é reconhecida por alcançar um bom equilíbrio entre velocidade e precisão, sendo especialmente adequada para sistemas embarcados ou com limitações computacionais, como o caso de um controle de acesso em tempo real.

Durante a prova de conceito, foram utilizados vídeos gravados na entrada da UTFPR – campus Dois Vizinhos. O YOLOv4 foi treinado ou adaptado para detectar a classe “placa”, e retornou ao sistema, com precisão, as coordenadas dos *bounding boxes* das regiões que continham placas veiculares.

O modelo foi executado em tempo quase real, alcançando uma taxa média de detecção de 68,61% em imagens com boas condições de iluminação. Para cenários com sombras, chuva ou ângulos mais inclinados, a detecção ainda se mostrou satisfatória, com uma leve queda no desempenho, mas sem comprometer o objetivo final do sistema.

Além disso, o YOLOv4 se mostrou eficaz mesmo em resoluções menores, o que é ideal para aplicações que não exigem equipamentos de alto desempenho. A arquitetura permite ajustar parâmetros como a confiança mínima e o tamanho da âncora, otimizando os resultados de acordo com as necessidades do ambiente.

Questionamento 2: O sistema consegue realizar o recorte preciso da placa para posterior OCR?

Sim. Após a detecção da placa pelo YOLOv4, os dados de coordenadas do *bounding box* são utilizados para extrair com precisão a região de interesse (ROI) da imagem original, onde se localiza a placa veicular. Esse recorte serve para que a etapa de OCR seja eficaz, pois reduz interferências visuais e melhora o contraste entre os caracteres e o fundo.

Antes da entrada no OCR, o sistema aplica uma sequência de técnicas de pré-processamento com OpenCV, incluindo conversão para escala de cinza (grayscale), filtro de mediana para redução de ruído e binarização (threshold adaptativo).

Esse *pipeline* tem como objetivo maximizar a clareza dos caracteres e diminuir os erros na etapa de reconhecimento. A extração da placa, combinada com esse pré-processamento, foi validada visualmente e demonstrou-se consistente mesmo com placas empoeiradas ou parcialmente sujas. A taxa de detecções corretas, a partir dos testes realizados com o *dataset* de validação foi de **68,61%**, sendo este número, apesar de relativamente baixo, suficiente para realização de detecções em um sistema que irá monitorar vídeo da entrada dos veículos.

Questionamento 3: O sistema reconhece corretamente os caracteres da placa utilizando OCR?

Sim, o sistema utiliza a ferramenta Tesseract OCR, com parâmetro “- - psm 6”, que corresponde ao modo de segmentação de página, em que o Tesseract assume um bloco uniforme de texto alinhado horizontalmente. Durante a execução da Prova de Conceito (PoC), esse modo demonstrou o melhor desempenho na leitura de placas veiculares, especialmente quando os caracteres estavam bem espaçados, alinhados e com contraste satisfatório.

Esses passos foram essenciais para aumentar a nitidez dos caracteres e garantir que o Tesseract interpretasse corretamente a sequência alfanumérica. Com isso, o sistema foi capaz de reconhecer com sucesso cerca de **72.55%** das placas em imagens com boa qualidade e em imagens com baixa resolução, sombras, ou sujeira sobre a placa, o que ainda é considerado satisfatório para a fase de validação.

A acurácia foi validada comparando os resultados do OCR com os dados de referência (*ground-truth*) armazenados em um arquivo JSON, contendo as placas corretas, possibilitando uma avaliação quantitativa da taxa de sucesso em diferentes cenários. Os testes reforçam que a configuração oferece uma combinação equilibrada entre precisão e confiabilidade, tornando-o adequado para aplicações reconhecimento de placas.

Questionamento 4: O sistema é capaz de consultar um banco de dados e verificar se o veículo está autorizado?

Sim. Após o reconhecimento dos caracteres da placa, o sistema realiza uma busca em uma estrutura de dados JSON, que simula um banco de dados real, contendo registros categorizados como aluno, professor ou visitante. Essa busca ocorre através da correspondência direta da string, reconhecida com as chaves do banco.

No ambiente de produção, esse JSON poderá ser substituído por um banco de dados relacional (como PostgreSQL ou MySQL), utilizando comandos SQL parametrizados para evitar injeções maliciosas. O sistema então, retorna uma resposta que indica se o veículo está autorizado ou não a entrar e qual a categoria do usuário (ex: “Acesso liberado para: Professor”). A Figura 6 apresenta a mensagem de resultado para a detecção mencionada. Esse processo simula a integração com o sistema e representa a etapa final da cadeia de valor do sistema proposto.

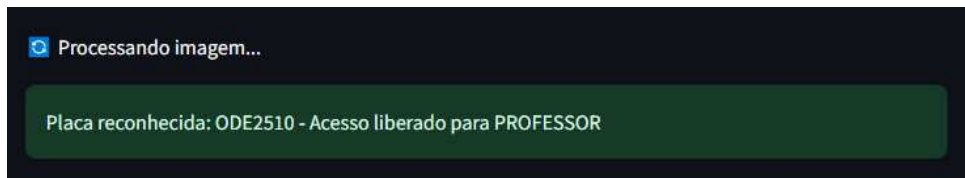


Figure 6. Imagem para demonstrar o retorno do sistema, por meio de imagem, vídeo ou câmera. (Do Autor, 2025)

Questionamento 5: O tempo de processamento permite a aplicação em tempo real?

Sim. Um dos objetivos principais da PoC foi verificar se o sistema conseguiria operar em tempo plausível, especialmente para controle de acesso de veículos. Com as otimizações feitas, o tempo médio de execução por imagem, foi de 1.01 segundos. As medições de tempo de execução foram realizadas utilizando todas as imagens do *dataset*.

Este tempo considera todas as etapas, o que é compatível com aplicações onde a leitura da placa acontece durante a aproximação do veículo à barreira física. Para vídeos, o sistema processa apenas 1 em cada 20 *frames*, mantendo o desempenho dentro de limites aceitáveis. Essa estratégia foi adotada considerando que, no cenário de controle de acesso, os veículos encontram-se parados ou em baixa velocidade, não sendo necessária a análise de todos os frames do vídeo. Dessa forma, reduz-se o custo computacional sem comprometer a precisão da detecção e do reconhecimento das placas. As medições de tempo de execução foram realizadas utilizando vídeos gravados na entrada do campus da Universidade Tecnológica Federal do Paraná (UTFPR) – campus Dois Vizinhos, os quais foram utilizados exclusivamente para fins de testes experimentais.

Questionamento 6: O sistema funciona corretamente ao processar vídeos (sequência de frames)?

Sim. O sistema foi adaptado para processar vídeos utilizando a biblioteca OpenCV. A leitura do vídeo ocorre com extração de *frames* a cada 20 quadros, e cada *frame* é processado da mesma forma que uma imagem estática: detecção da placa, recorte, pré-processamento, OCR e verificação no banco.

Durante os testes, vídeos com resolução HD (720) a 30 fps (*Frames Per Second*) foram utilizados. A cada 20 *frames*, um novo *frame* era processado, garantindo fluidez e leitura consistente. As placas foram lidas corretamente em múltiplos quadros consecutivos, permitindo

realizar uma votação de confiança entre os resultados extraídos, reduzindo o risco de erro por leitura única equivocada.

Questionamento 7: O sistema consegue distinguir corretamente diferentes categorias de veículos (aluno, professor, visitante)?

Sim. Essa distinção é feita após o reconhecimento da placa, ao consultar a base de dados. Cada entrada da base contém uma estrutura como demonstrado na Figura 7.

```
[{"placa": "ODE2510",  
  "tipo": "professor"},  
,  
,  
,  
{"placa": "OJJ3984",  
  "tipo": "aluno"},  
,  
,  
,  
{"placa": "PPC5431",  
  "tipo": "visitante"}]
```

Figure 7. Imagem para demonstrar a estrutura utilizada no JSON. (Do Autor, 2025)

Após a extração da string da placa, o sistema verifica se ela está registrada e exibe automaticamente a categoria do veículo. Isso permite com que, por exemplo, veículos de professores tenham prioridade de entrada ou acesso irrestrito, enquanto alunos e visitantes podem passar por validações adicionais.

Questionamento 8: Quais foram os principais desafios técnicos encontrados durante a prova de conceito?

Durante o desenvolvimento e execução da PoC, os seguintes desafios foram identificados:

- Na versão mais nova do YOLOv4 e com melhor treinamento direcionado à placa de veículos, os resultados podem evoluir drasticamente e melhorar o restante do percentual atual do sistema.
- Variações de iluminação: placas em ambientes escuros ou com forte contraluz tiveram leitura prejudicada, exigindo técnicas avançadas de equalização e *thresholding* adaptativo.
- Ângulo da câmera: detecções em ângulos muito inclinados comprometeram a segmentação precisa da placa.
- Erros do OCR: mesmo com pré-processamento, o Tesseract apresentou confusões comuns entre caracteres parecidos. Isso reforça a necessidade de pós-processamento com validação semântica e regex.
- Ruído visual: placas com sujeira, desgaste ou elementos visuais adicionais (ex: parafusos na posição de letras) afetaram a acurácia.

- Desempenho em vídeos longos: a leitura contínua de quadros pode levar a sobrecarga, sendo necessário limitar o número de quadros processados e otimizar o uso da CPU.

Questionamento 9: O sistema consegue manter estabilidade e desempenho em diferentes condições ambientais (iluminação, chuva, movimento)?

Durante a prova de conceito, foram realizados testes com imagens e vídeos coletados em diferentes períodos do dia, incluindo manhã com sol direto, tarde nublada e início da noite, além de simulações com chuva leve. O sistema mostrou boa resiliência quanto às variações ambientais, desde que as placas estivessem visíveis e não excessivamente sujas ou danificadas.

A detecção realizada pelo YOLOv4 demonstrou ser estável mesmo com sombras, reflexos e variações de intensidade luminosa. Isso se deve ao fato do modelo ter sido treinado (ou refinado) com amostras contendo diferentes condições visuais. Em casos com menor luminosidade, foi observado um pequeno aumento na dificuldade de detecção.

Já o OCR com Tesseract, apresentaram redução de desempenho em imagens com baixa nitidez ou excesso de gotas de chuva sobre a lente, pois essas condições afetam diretamente a legibilidade dos caracteres.

Portanto, embora o desempenho ideal ocorra em condições ambientais controladas, o sistema demonstrou capacidade de adaptação e tolerância a ruídos ambientais moderados, confirmando sua viabilidade para uso real.

Questionamento 10: O sistema é escalável e pode ser adaptado para uso em outros contextos (condomínios, estacionamentos, empresas)?

Sim. A arquitetura do sistema foi projetada com foco em modularidade, portabilidade e escalabilidade, o que facilita sua adaptação para diferentes cenários além do ambiente universitário. Cada componente do sistema — detecção com YOLOv4, pré-processamento com OpenCV, OCR com Tesseract e verificação com base de dados — é desacoplado, permitindo que seja substituído, expandido ou paralelizado de acordo com as necessidades do projeto. A seguir são citados possíveis ambientes nos quais a solução poderia ser implementada:

- Em condomínios residenciais, o sistema pode ser integrado a sistemas de portaria eletrônica e monitoramento por câmeras IP.
- Em empresas e estacionamentos comerciais, pode ser conectado a sistemas de gestão de acesso com interface web, painéis informativos e *logs* de entrada e saída.
- Em sistemas públicos (ex: pedágios, zonas de restrição urbana), pode ser escalado para processar múltiplos fluxos de vídeo em paralelo utilizando *clusters* com suporte a GPU.

A integração com banco de dados relacional (PostgreSQL, MySQL) ou NoSQL (MongoDB) também permite escalar o volume de veículos registrados e acessos controlados. Adicionalmente, o uso de APIs RESTful para a comunicação entre módulos, facilita a adoção em arquiteturas baseadas em microserviços.

Por fim, o sistema pode ser distribuído como uma aplicação em nuvem ou embarcada em dispositivos *edge* (como Jetson Nano ou Raspberry Pi com acelerador Coral), desde que se realize ajustes de desempenho. Isso demonstra que a proposta do TCC é tecnicamente viável e replicável em ambientes reais variados, ampliando seu valor prático e comercial. Figura 8

5. Conclusão

Com base nas análises realizadas, observou-se que o sistema de reconhecimento de caracteres em placas possui um potencial significativo. Os resultados demonstraram viabilidade em alcançar um desempenho satisfatório, desde que sejam aplicadas as estratégias adequadas.



Figure 8. Tela inicial do sistema feita pelo Streamlit. (Do Autor, 2025)

O sistema reconheceu corretamente 68,61% das placas em condições diversas de iluminação. Esses resultados foram obtidos a partir de imagens que apresentam cenários mais desafiadores, incluindo variações de iluminação associadas a períodos noturnos e a fortes exposições solares, bem como diferenças no ângulo de captura, presença de sujeira nas placas, variações na qualidade das imagens e condições climáticas distintas, como chuva, sol e neblina. Dessa forma, os resultados indicam que o sistema possui potencial para operar de maneira eficaz em cenários diurnos e noturnos, bem como sob diferentes condições ambientais.

Apesar da taxa de detecção ser relativamente baixa, destaca-se que os pesos da rede estavam pré-treinados para um contexto diferente do explorado, e em caso de treinamento específico a taxa de detecção tende a melhorar significativamente. No que se refere ao tempo de processamento, este foi realizado por meio de CPUs somente e o uso de GPUs pode melhorar o tempo de resposta.

References

- [Guingo et al. 2002] Guingo, B. C., Rodrigues, R. J., and THOME, A. C. G. (2002). Automatic identification for automotives vehicles plates. In *3rd WSEAS International Conference on Neural Networks and Applications (NNA'02)*, volume 2, pages 15–21.
- [Labaki and Woiski 2003] Labaki, J. and Woiski, E. (2003). Introdução a python—módulo a. *Grupo Python, UNESP-Ilha Solteira*.
- [Laroca et al. 2022] Laroca, R., Santos, M., Estevam, V., Luz, E., and Menotti, D. (2022). A first look at dataset bias in license plate recognition. In *2022 35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, volume 1, pages 234–239. IEEE.

- [Laroca et al. 2021] Laroca, R., Zanlorensi, L. A., Gonçalves, G. R., Todt, E., Schwartz, W. R., and Menotti, D. (2021). An efficient and layout-independent automatic license plate recognition system based on the yolo detector. *IET Intelligent Transport Systems*, 15(4):483–503.
- [Marengoni and Stringhini 2009] Marengoni, M. and Stringhini, S. (2009). Tutorial: Introdução à visão computacional usando opencv. *Revista de Informática Teórica e Aplicada*, 16(1):125–160.
- [Oliveira and Gonzaga 2012] Oliveira, L. A. and Gonzaga, A. (2012). Localização, segmentação e reconhecimento de caracteres em placas de automóveis. *Neves et al.(Eds.), Avanços em Visao Computacional*, page 283.
- [Rosebrock 2017] Rosebrock, A. (2017). Using tesseract ocr with python. *March*, 6:2021.