

# Identificação de Placas de Veículos com Processamento de Imagem e OCR

Guilherme Zanini<sup>a</sup>, Gabriel Strada<sup>a</sup>

<sup>a</sup>Escola Politécnica, Universidade do Vale do Rio dos Sinos - UNISINOS, São Leopoldo, Brasil

## ARTICLE INFO

### Keywords:

Visão Computacional  
Reconhecimento de Placas  
OCR  
Processamento de Imagens  
Tesseract  
OpenCV

## ABSTRACT

Este artigo apresenta o desenvolvimento de um sistema para a identificação de placas de veículos utilizando técnicas de processamento de imagem e OCR. Com a aplicação de filtros e algoritmos de reconhecimento óptico de caracteres, a ferramenta foi capaz de detectar e interpretar as informações contidas nas placas de maneira eficiente. Os resultados indicam que o sistema é eficaz em diferentes condições de iluminação e formatos de placas, demonstrando a viabilidade da aplicação para cenários reais.

## 1. Introdução

O reconhecimento automático de placas de veículos é um recurso essencial para segurança pública e controle de tráfego, permitindo a identificação rápida e precisa em cenários urbanos e rodoviários. Com o aumento do número de veículos, surgem desafios como a variação de iluminação e diferentes estilos de placas, exigindo técnicas robustas de processamento de imagem e OCR (Optical Character Recognition). Este trabalho apresenta um sistema que utiliza a biblioteca OpenCV e o Tesseract OCR para detectar e interpretar caracteres de placas, aplicando filtros e segmentação de contornos para assegurar a precisão mesmo em condições adversas. A solução desenvolvida busca facilitar o monitoramento automatizado de veículos, com aplicações em pedágios, segurança e controle de tráfego.

## 2. Materiais e métodos

Para construir um sistema de reconhecimento de placas, utilizamos uma abordagem que combina processamento de imagem e OCR, dividida em etapas específicas. Esse processo prepara a imagem para identificar regiões de interesse e aplica algoritmos de reconhecimento óptico para interpretar os caracteres das placas com precisão, considerando as diferentes condições visuais encontradas em cenários reais.

### 2.1. Processamento de Imagem

Para iniciar o processo de identificação, as imagens capturadas passam por uma etapa de pré-processamento utilizando a biblioteca OpenCV [1]. Primeiramente, a imagem original é convertida para tons de cinza, facilitando a detecção de bordas e contornos.

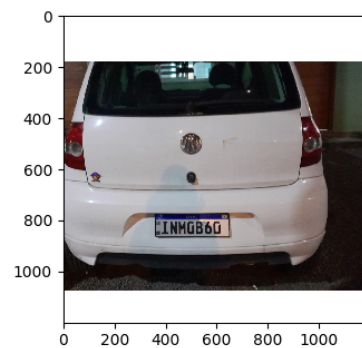


Figure 1: Imagem Original

Em seguida, aplicamos um filtro bilateral para suavizar a imagem, removendo ruídos sem comprometer as bordas dos caracteres das placas.

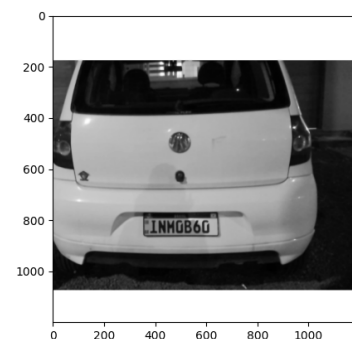



Figure 2: Imagem em tons de cinza e suavizada

Por fim, é aplicada uma limiarização adaptativa gaussiana, que converte a imagem em um formato binário, destacando as regiões de interesse e minimizando os efeitos de variações de luminosidade.

 gzaninis@edu.unisinos.br (G. Zanini); gstrada@edu.unisinos.br (G. Strada)  
ORCID(s):

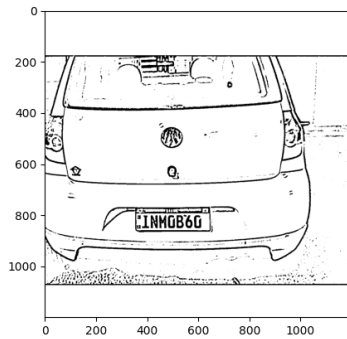


Figure 3: Imagem Binarizada

## 2.2. Segmentação de Contornos

A partir da imagem processada, identificamos os contornos que possam representar placas de veículos, aplicando filtros específicos [3].

Os contornos identificados passam pelas seguintes regras de filtragem:

- **Proporção entre altura e largura:** contornos cuja altura seja significativamente maior que a largura, ou vice-versa, são descartados.
- **Área mínima e máxima:** são eliminados os contornos com área muito pequena, que indicam ruído, e os muito grandes, que provavelmente não representam uma placa.

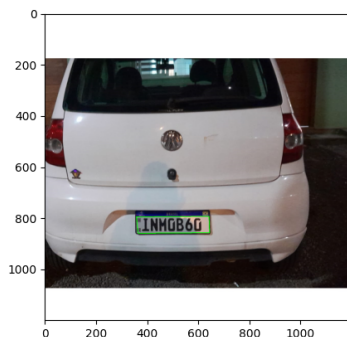


Figure 4: Imagem processada mostrando o contorno detectado

Após essa filtragem, os contornos que atendem aos critérios são considerados como possíveis placas.

A imagem recortada da possível placa passa por uma sequência de operações para otimizar a detecção dos caracteres:

- **Conversão para tons de cinza:** facilita o processamento dos caracteres.



Figure 5: Possível placa detectada

- **Limiarização adaptativa:** destaca caracteres grandes e reduz o ruído de caracteres menores.
- **Operação de fechamento:** preenche regiões internas dos contornos para tornar os caracteres mais sólidos.
- **Operação de abertura:** remove ruídos indesejados nos contornos.
- **Dilatação:** aumenta a espessura dos caracteres, facilitando a identificação pelo OCR.
- **Erosão:** reduz a espessura dos caracteres para aprimorar a definição dos contornos.



Figure 6: Placa após transformações

## 2.3. Reconhecimento de Caracteres (OCR)

Com os contornos das placas identificados, a próxima etapa é o reconhecimento dos caracteres. Utilizamos o Tesseract OCR para interpretar o texto contido nas placas [2], configurando o idioma como português (lang=por). Caso o reconhecimento inicial não seja satisfatório, aplicamos uma nova tentativa em inglês (lang=eng), ampliando as possibilidades de sucesso na identificação dos caracteres. O OCR retorna as informações detectadas, que são posteriormente validadas para assegurar que atendem ao formato esperado de uma placa. A integração com Python foi realizada por meio da biblioteca Pytesseract [4].

Placa Detectada  
INH0B60

Figure 7: Placa identificada

Se mesmo assim nenhuma placa for identificada, o resultado do OCR em português é manipulado para identificar possíveis letras que sejam números no modelo Mercosul e antigo. Dessa forma, todas as possibilidades de placas são retornadas, considerando uma margem de erro do OCR. [5]

Placa Detectada  
LLW5116  
LLW9116  
  
Mercosul:  
LLWS116  
LLW5116  
LLW9116

Figure 8: Lista de possibilidades

## 2.4. Exibição do Resultado Final

Após o processamento completo, o sistema exibe o resultado final, mostrando a imagem original com as regiões de possíveis placas destacadas e uma lista das placas detectadas, juntamente com os caracteres reconhecidos. Esse passo permite validar visualmente a precisão do sistema, comparando a entrada com a saída processada.

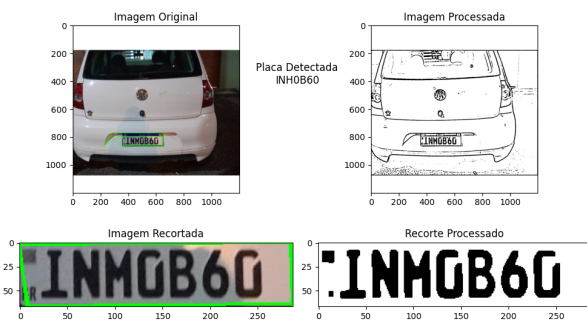


Figure 9: Exibição do Resultado Final

## 3. Resultados e Discussão

O sistema desenvolvido foi testado em um conjunto de imagens de placas de veículos, capturadas em condições

variadas de iluminação e ângulos. Observou-se que o processo de pré-processamento e segmentação de contornos permitiu uma detecção consistente das placas, com uma taxa de sucesso elevada para imagens com boa visibilidade dos caracteres.

Nos testes iniciais, os filtros de contornos e as operações de dilatação e erosão demonstraram eficiência na preparação dos caracteres para o OCR, resultando em uma leitura precisa das placas. No entanto, algumas limitações foram identificadas em cenários de baixa iluminação ou em imagens com reflexos excessivos, onde o OCR apresentou dificuldades para interpretar os caracteres com exatidão. Ainda assim, a aplicação de limiarização adaptativa mostrou-se eficaz para reduzir esses impactos, melhorando a legibilidade dos caracteres em muitos casos.

Outro ponto de discussão envolve a taxa de falsos positivos e negativos nas etapas de segmentação de contornos. Contornos não pertencentes a placas foram em alguns casos identificados erroneamente, mas as regras de área e proporção ajudaram a minimizar esse problema.

## 4. Conclusão

Este trabalho apresentou o desenvolvimento de um sistema de reconhecimento de placas de veículos utilizando técnicas de processamento de imagem e OCR. Combinando o uso das bibliotecas OpenCV e Tesseract OCR, foi possível construir um sistema que identifica e interpreta caracteres de placas com alta precisão, especialmente em condições controladas de iluminação e ângulo.

A metodologia aplicada mostrou-se eficaz, com cada etapa contribuindo para a robustez e precisão do sistema. Apesar de limitações observadas em situações de baixa iluminação ou presença de reflexos, o método de limiarização adaptativa e as operações de morfologia foram capazes de mitigar parte desses efeitos adversos, destacando-se como componentes essenciais do pipeline de processamento.

Para trabalhos futuros, recomenda-se o aprimoramento do sistema com a adição de técnicas de aprendizado de máquina para uma segmentação de contornos mais precisa e adaptável. Essas melhorias podem aumentar ainda mais a aplicabilidade do sistema em cenários variados, contribuindo para áreas como segurança, fiscalização de tráfego e monitoramento automatizado de veículos.

## References

- [1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- [2] Google Developers. *Tesseract OCR: An Open-Source Optical Character Recognition Engine*, 2024. Disponível em: <https://github.com/tesseract-ocr/tesseract>.
- [3] OpenCV Developers. *OpenCV Documentation*, 2024. Disponível em: <https://docs.opencv.org/>.
- [4] Pytesseract Community. *Pytesseract Documentation*, 2024. Disponível em: <https://pypi.org/project/pytesseract/>.
- [5] Tesseract OCR Community. *Tesseract OCR Language Data Files*, 2024. Disponível em: <https://github.com/tesseract-ocr/tessdata>.