

Avaliação de Desempenho entre Bibliotecas Populares no Reconhecimento Digital de Placas

Rafael Garay Trindade,* Claudio Schepke

¹Laboratório de Estudos Avançados em Computação (LEA)
Universidade Federal do Pampa (UNIPAMPA) – Alegrete – RS – Brazil

{rafaeltrindade.aluno, claudioschepke}@unipampa.edu.br

Resumo. *Este trabalho avalia diversas abordagens para detecção e reconhecimento de placas de veículos utilizando bibliotecas populares. Ferramentas como Google Cloud Vision, OpenCV com Pytesseract e YOLOv8 são comparadas em relação à eficiência e precisão. Os resultados demonstram a influência significativa da qualidade das imagens e das condições de iluminação no desempenho dos métodos avaliados.*

1. Introdução

A visão computacional tem se tornado cada vez mais presente em diversas áreas do conhecimento e aplicações práticas, possibilitando desde reconhecimento facial e de objetos até a leitura automatizada de caracteres. A maioria de suas aplicações está centrada no reconhecimento de padrões complexos para resolver problemas que apresentariam dificuldade considerável para solução humana, transformando processos que demandariam dias de trabalho em tarefas realizadas em milésimos de segundo.

O reconhecimento digital de placas de veículos constitui um exemplo emblemático da aplicação dessa tecnologia, com utilidade comprovada em sistemas de fiscalização eletrônica, monitoramento de trânsito, segurança patrimonial e controle de acesso veicular. No entanto, as soluções automatizadas enfrentam desafios significativos, sendo sensíveis a fatores como condições de iluminação, qualidade de imagem, obstruções parciais e variações angulares.

Neste contexto, o presente trabalho investiga métodos distintos para detectar e reconhecer placas de veículos utilizando três abordagens principais: *Google Cloud Vision*, *OpenCV com Pytesseract* e *YOLOv8*. A análise comparativa busca identificar pontos fortes e limitações de cada método sob diferentes condições operacionais.

As contribuições deste trabalho incluem: Avaliação comparativa de diferentes técnicas de detecção de placas veiculares; Análise das bibliotecas e *frameworks* mais utilizados neste domínio; Discussão sobre o desempenho relativo dos métodos analisados, considerando variáveis ambientais; Diretrizes para a seleção da abordagem mais adequada, conforme o cenário de aplicação.

2. Trabalhos Relacionados

A literatura especializada reporta diversas abordagens para o problema de reconhecimento automático de placas veiculares. Entre as bibliotecas mais utilizadas para detecção de

*Bolsista PRO-IC Unipampa 2024.

caracteres em imagens destaca-se o *OpenCV*, frequentemente combinado com ferramentas de OCR como o *Pytesseract* [Figuroa 2025]. Estudos anteriores demonstraram a aplicação dessas ferramentas para reconhecimento de placas, obtendo resultados variáveis que correlacionam-se diretamente com a qualidade das imagens analisadas.

Abordagens tradicionais geralmente envolvem um pipeline de processamento que inclui etapas de: (1) detecção de regiões de interesse; (2) pré-processamento da imagem; (3) segmentação de caracteres; e (4) reconhecimento óptico dos caracteres identificados. Cada uma dessas etapas apresenta desafios específicos e pontos de falha em potencial.

Mais recentemente, técnicas baseadas em aprendizado profundo vêm ganhando destaque na literatura. Redes neurais convolucionais (CNNs) e arquiteturas do tipo *You Only Look Once* (YOLO) mostraram-se particularmente eficazes para a tarefa de detecção de placas em condições variadas, superando em muitos casos as abordagens tradicionais.

Métodos baseados em serviços hospedados na nuvem, como *Google Cloud Vision*, beneficiam-se de modelos treinados com grandes conjuntos de dados diversificados, apresentando geralmente melhor performance quando confrontados com formatos não vistos durante o treinamento específico do sistema. Contudo, essa vantagem implica dependência da conectividade e possíveis preocupações com a privacidade dos dados.

O presente trabalho diferencia-se pela análise comparativa abrangente que inclui tanto abordagens baseadas em serviços em nuvem (*Google Cloud Vision*) quanto soluções locais baseadas em frameworks de visão computacional (*OpenCV com Pytesseract*) e abordagens de estado da arte baseadas em aprendizado profundo (*YOLOv8*).

3. Metodologia e Tecnologias

Para este estudo comparativo, selecionamos três abordagens representativas de diferentes paradigmas de reconhecimento digital de placas veiculares:

- **Google Cloud Vision:** API comercial da Google para reconhecimento de texto e objetos em imagens, representando soluções baseadas em serviços na nuvem com modelos pré-treinados em grandes conjuntos de dados [Google, Inc. 2025];
- **OpenCV & Pytesseract:** Combinação de um framework de visão computacional com uma *engine* de OCR, representando abordagens tradicionais de processamento de imagem [OpenCV team 2025, Python Software Foundation 2025];
- **YOLOv8:** Implementação recente da arquitetura YOLO para detecção de objetos em imagens e vídeos, representando abordagens modernas baseadas em aprendizado profundo [Roboflow, Inc. 2025].

O conjunto de dados utilizado incluiu imagens capturadas em diferentes condições de iluminação e qualidade, cuidadosamente selecionadas para testar a robustez dos métodos frente a variações ambientais. Para a realização dos testes, foram selecionadas 3000 imagens com diferentes níveis de qualidade, organizadas em três categorias: **Imagens de teste:** Conjunto balanceado com qualidade razoável e proximidade adequada da placa; **Imagens com qualidade superior:** Conjunto caracterizado por alta qualidade gráfica, com variação controlada na distância em relação à placa; **Imagens com qualidade inferior:** Conjunto contendo imagens com diversos problemas de iluminação, saturação, nitidez e obstruções parciais.

Tabela 1. Resultados comparativos dos métodos de detecção de placas

Método	OpenCV & Pytesseract	Google Cloud Vision	YOLOv8
Imagens de Teste	759/1000	930/1000	789/1000
Imagens de Qualidade	879/1000	912/1000	789/1000
Imagens sem Qualidade	687/1000	832/1000	720/1000
Tempo Médio (ms)	320	980	150

Todas as imagens foram obtidas a partir de *datasets* públicos e confiáveis. A seleção buscou garantir uma ampla variedade de cenários e boa representatividade das situações comuns de leitura de texto e detecção visual, incluindo imagens com diferentes resoluções, contrastes, níveis de ruído, tipos de fonte e condições de iluminação.

Para cada abordagem, implementamos um pipeline completo de processamento que incluiu: aquisição da imagem, pré-processamento (quando aplicável), detecção da placa, segmentação de caracteres e reconhecimento final do texto. Os experimentos foram conduzidos em um ambiente controlado, garantindo condições equitativas de comparação. A avaliação de performance considerou tanto métricas quantitativas (acurácia, precisão, *recall*, *F1-score*) quanto observações qualitativas sobre o comportamento dos algoritmos frente aos desafios descritos na Seção ??.

4. Resultados e Análise

Os resultados obtidos na avaliação experimental são sumarizados na Tabela 1, que demonstra a eficiência comparativa de cada método nas diferentes categorias de imagens.

A análise dos resultados revela padrões interessantes sobre o desempenho relativo das abordagens avaliadas. O *Google Cloud Vision* demonstrou a maior acurácia geral, particularmente no conjunto de imagens com qualidade inferior, beneficiando-se provavelmente dos extensivos conjuntos de dados utilizados no treinamento de seus modelos e dos avançados algoritmos de pré-processamento implementados na plataforma cloud.

A combinação *OpenCV com Pytesseract* apresentou performance satisfatória em condições ideais (imagens de qualidade superior), mas mostrou-se significativamente sensível à degradação da qualidade da imagem, com decréscimo pronunciado de acurácia no conjunto de imagens com problemas. Esta observação corrobora a premissa de que abordagens tradicionais de processamento de imagem dependem criticamente da qualidade do material de entrada.

O *YOLOv8* destacou-se pelo equilíbrio entre acurácia e eficiência computacional, apresentando o menor tempo médio de processamento entre as abordagens avaliadas. Este resultado era esperado, dada a arquitetura eficiente da família YOLO, projetada especificamente para aplicações que demandam processamento em tempo real.

Além das métricas quantitativas, realizamos uma análise qualitativa aprofundada do comportamento dos algoritmos diante dos desafios específicos:

- **Iluminação inadequada:** O *Google Cloud Vision* mostrou-se mais resiliente a variações de iluminação, enquanto o *OpenCV com Pytesseract* exigiu ajustes manuais nos parâmetros de pré-processamento para diferentes condições;

- **Variações angulares:** O YOLOv8 demonstrou melhor desempenho em imagens com ângulos não ortogonais, graças à capacidade de aprendizado de características invariantes a transformações geométricas;
- **Oclusões parciais:** Abordagens baseadas em *deep learning* (YOLOv8 e Google Cloud Vision) mostraram-se superiores na manipulação de oclusões, conseguindo, em muitos casos, inferir caracteres parcialmente obstruídos;
- **Variações de fontes:** O Google Cloud Vision apresentou a melhor performance com fontes não convencionais, devido ao treinamento com dados diversificados.

5. Considerações Finais

Este trabalho apresentou uma análise comparativa abrangente de três abordagens para reconhecimento automático de placas veiculares: Google Cloud Vision, OpenCV com Pytesseract e YOLOv8. Os resultados demonstram que cada método possui vantagens e limitações específicas, tornando necessária uma análise criteriosa do contexto de aplicação para a seleção da solução mais adequada.

O *Google Cloud Vision* destacou-se pela acurácia superior, particularmente em condições desfavoráveis, mas à custa de maior latência (devido à comunicação com servidores remotos) e dependência de conectividade de rede. Esta abordagem mostra-se mais adequada para aplicações que não demandam processamento em tempo real e onde a máxima acurácia é prioritária.

A combinação *OpenCV com Pytesseract* mostrou-se uma solução viável para aplicações com restrições orçamentárias e em que as condições de captura de imagem podem ser controladas. No entanto, sua sensibilidade a variações na qualidade da imagem limita sua aplicabilidade em cenários não controlados.

O *YOLOv8* apresentou o melhor equilíbrio geral entre acurácia e eficiência, com tempos de processamento adequados para aplicações em tempo real. Esta abordagem mostrou-se particularmente promissora para implementações embarcadas ou onde a independência de conectividade é um requisito.

Trabalhos futuros podem explorar a combinação híbrida dessas abordagens, aproveitando as vantagens de cada paradigma. Adicionalmente, a expansão do conjunto de dados para incluir mais exemplos de condições extremas e placas de diferentes regiões geográficas pode proporcionar percepções adicionais sobre a robustez dos algoritmos.

Referências

- Figuerola, F. T. (2025). Automatic number plate recognition with Python, YOLOv8 and EasyOCR — Computer vision tutorial. https://youtu.be/fyJB1t0o0ms?si=Qj7_Ch3EsSSmSY1j.
- Google, Inc. (2025). Documentação da API Cloud Vision. <https://cloud.google.com/vision/docs?hl=pt-br>.
- OpenCV team (2025). OpenCV is the world's biggest computer vision library. <https://opencv.org>.
- Python Software Foundation (2025). pytesseract 0.3.13. <https://pypi.org/project/pytesseract/>.
- Roboflow, Inc. (2025). Explore Ultralytics YOLOv8. <https://yolov8.com>.