

Universidade Federal do Rio Grande do Norte
Instituto Metr pole Digital

Curso: Bacharelado em Tecnologia da Informa  o
Disciplina: IMD1101 - Aprendizado de M quina

Relat rio T cnico - 3  Unidade

**Reconhecimento Autom tico de Placas Veiculares
utilizando YOLOv8 e T cnicas de OCR Baseadas em
Redes Neurais Profundas**

Gabrielle de Vasconcelos Borja

Mar a Paz Marcato

14 de Julho de 2025

Resumo

Este relatório apresenta o desenvolvimento de um pipeline completo para reconhecimento automático de placas veiculares (ALPR), integrando técnicas modernas de Aprendizado de Máquina supervisionado baseadas em Redes Neurais Profundas. O pipeline é composto por duas etapas principais: a detecção da placa na imagem, utilizando a arquitetura YOLOv8n, e o reconhecimento dos caracteres via OCR usando TrOCR.

A detecção foi realizada com um modelo YOLOv8n treinado do zero com imagens anotadas no formato YOLO, exportadas pela plataforma Roboflow. O treinamento ocorreu no ambiente Google Colab com suporte a GPU, utilizando 1.200 imagens para treino e 265 para validação. O reconhecimento óptico de caracteres foi implementado com o OCR Tesseract e com o modelo Transformer TrOCR da Microsoft.

Os experimentos demonstraram que a combinação YOLOv8n com TrOCR oferece melhor legibilidade e maior precisão na leitura de placas, superando o OCR tradicional em diversos cenários. Embora o conjunto de dados para OCR contenha 1.500 imagens disponíveis, as análises qualitativas foram conduzidas sobre subconjuntos amostrados. Estes resultados destacam a eficácia de modelos baseados em aprendizado profundo para aplicações de automação e monitoramento veicular.

1 Introdução

Com o crescimento das cidades e o aumento da frota de veículos, tornou-se essencial adotar tecnologias que auxiliem na automação do controle e monitoramento do tráfego. Nesse contexto, os sistemas de reconhecimento automático de placas veiculares (ALPR) têm se popularizado, sendo aplicados em segurança pública, cobrança automática em pedágios, controle de acesso a estacionamentos e fiscalização de trânsito.

O reconhecimento de placas veiculares normalmente envolve duas etapas: (1) detecção da placa e (2) reconhecimento óptico dos caracteres (OCR). Ambas as etapas enfrentam desafios como variações de iluminação, oclusões, placas danificadas e diferentes formatos. O uso de redes neurais profundas e arquiteturas modernas, como YOLOv8 e Transformers, tem se mostrado promissor para superar essas limitações.

Neste trabalho, foi desenvolvido um pipeline completo utilizando o modelo YOLOv8n (versão leve da arquitetura) para detecção de placas, e duas abordagens distintas de OCR: o tradicional Tesseract e o modelo TrOCR da Microsoft, baseado em Transformers. As imagens foram anotadas na plataforma Roboflow e os experimentos conduzidos no Google Colab, aproveitando recursos de GPU.

O conjunto de dados utilizado é composto por 5.000 imagens de veículos brasileiros

capturadas em praças de pedágio. Destas, foram utilizadas 1.200 imagens para o treinamento do modelo de detecção, 265 para validação e outras 1.500 estavam disponíveis para testes de OCR. No entanto, as análises foram realizadas sobre subconjuntos amostrados, como 94 imagens inferidas após o treinamento e outras 50 utilizadas em testes qualitativos. A detecção foi avaliada com métricas como mAP@50, e os textos extraídos foram comparados qualitativamente entre os métodos de OCR.

2 Descrição da Base de Dados

a) Origem e características gerais

A base de dados utilizada neste projeto é o RodoSol-ALPR Dataset, introduzida por Laroca et al. (2022) na conferência VISAPP. O conjunto é composto por 20.000 imagens capturadas por câmeras estáticas instaladas em praças de pedágio operadas pela concessionária *Rodovia do Sol (RodoSol)*, que administra um trecho de 67,5 km da rodovia ES-060, no estado do Espírito Santo, Brasil.

As imagens apresentam resolução de 1280×720 pixels e incluem diversos tipos de veículos (automóveis, caminhões, ônibus e motocicletas), capturados em diferentes condições de iluminação (dia e noite), meteorologia (tempo claro e chuvoso) e ângulos de visão (distâncias variadas da câmera). Esse cenário proporciona uma ampla diversidade visual, o que torna a base robusta para treinar e avaliar modelos de detecção e OCR.

Uma das principais características do RodoSol-ALPR é a presença de dois tipos de layout de placas veiculares:

- **Placas Brasileiras:** padrão anterior ao Mercosul, com três letras seguidas por quatro dígitos (e.g., ABC-1234);
- **Placas Mercosul:** novo padrão com três letras, um número, uma letra e dois números (e.g., ABC1D23).

Além disso, o formato da placa também varia de acordo com o tipo de veículo. Veículos de quatro rodas ou mais (denominados “carros”) apresentam os sete caracteres em uma única linha, enquanto as motocicletas e triciclos têm a placa distribuída em duas linhas.

b) Qualidade dos dados

As imagens são de alta resolução e foram cuidadosamente selecionadas para incluir variações realistas do ambiente urbano. Todas as imagens foram anonimizadas manual-

mente, com rostos de pessoas desfocados para preservar a privacidade.

Outro diferencial da base é a anotação precisa dos quatro cantos das placas em vez de apenas retângulos. Isso permite a aplicação de técnicas de retificação de perspectiva e data augmentation mais sofisticadas, especialmente úteis para modelos baseados em redes neurais profundas.

c) Atributos mais relevantes

Cada imagem do dataset está acompanhada de um arquivo contendo:

- O tipo de veículo (carro ou motocicleta);
- O layout da placa (Brasileira ou Mercosul);
- O texto contido na placa (e.g., XYZ-9876);
- As coordenadas (x, y) dos quatro cantos da placa.

Esses metadados foram fundamentais para a conversão ao formato YOLO utilizado no treinamento supervisionado do modelo de detecção (YOLOv8n) e para a extração das regiões de interesse (ROIs) destinadas à etapa de OCR.

Embora o dataset completo contenha 20.000 imagens divididas entre carros e motocicletas, com placas nos padrões brasileiro e Mercosul, neste projeto optou-se por utilizar um subconjunto com 3.000 imagens de carros com placas brasileiras (pasta `cars-br`). Essa escolha teve como objetivo restringir o escopo inicial do estudo, priorizando a padronização visual e o controle da variabilidade no processo de anotação.

As 3.000 imagens selecionadas foram organizadas da seguinte forma:

- **Treinamento do modelo de detecção:** 1.200 imagens;
- **Validação da detecção:** 265 imagens;
- **Disponíveis para OCR:** 1.500 imagens (das quais, subconjuntos foram efetivamente utilizados para inferência e comparação entre métodos).

O subconjunto destinado ao treinamento do YOLOv8n foi anotado manualmente na plataforma Roboflow, que oferece ferramentas de rotulagem e exportação no formato compatível com o framework YOLOv8, facilitando a integração com o processo de treinamento supervisionado. Já para a etapa de OCR, os recortes foram gerados a partir das bounding boxes detectadas automaticamente pelo modelo treinado, e então processados com as técnicas apropriadas para aplicação dos modelo TrOCR.

3 Metodologia

a) Etapas da Análise

O pipeline desenvolvido foi dividido em três grandes etapas:

1. **Preparação dos dados:** seleção de 3.000 imagens da subpasta `cars-br` do dataset RodoSol-ALPR; anotação com bounding boxes no Roboflow; exportação no formato YOLOv8.
2. **Treinamento do detector:** uso da arquitetura YOLOv8n com 1200 imagens para treino e 265 para validação, durante 75 épocas com tamanho de imagem 640×640 e otimizador AdamW.
3. **OCR e comparação:** aplicação dos métodos TrOCR e Tesseract OCR nas regiões detectadas (ROIs). As imagens foram recortadas com margem, redimensionadas, convertidas para tons de cinza e submetidas a diferentes estratégias de OCR. O TrOCR utilizou o modelo `trocr-base-printed` da Microsoft com redimensionamento fixo para 320×80 pixels, enquanto o Tesseract foi aplicado com binarização Otsu e configuração `-psm 7`. Os textos extraídos foram comparados qualitativamente e registrados em anotações sobre as imagens.

b) Tratamento de Dados Ausentes e Criação de Variáveis

Durante o carregamento e anotação, imagens sem placas visíveis, corrompidas ou duplicadas foram removidas. As imagens utilizadas no OCR passaram por:

- Adição de margem à bounding box detectada;
- Redimensionamento da placa recortada;
- Conversão para escala de cinza, suavização com `GaussianBlur` e limiarização adaptativa (`adaptiveThreshold`);
- Geração de arquivos temporários para visualização e avaliação dos textos reconhecidos.

c) Modelos Utilizados

- **YOLOv8n:** detector de objetos baseado em CNNs, treinado do zero a partir de uma configuração personalizada (`yolov8n.yaml`), sem uso de pesos pré-treinados;

- **TrOCR (base-printed)**: modelo Transformer pré-treinado da Microsoft para OCR em imagens impressas, utilizado via **transformers** da HuggingFace. A entrada da imagem foi redimensionada para 320×80 pixels. O modelo apresenta robustez frente a ruídos, distorções e variações no enquadramento das placas.
- **Tesseract OCR**: ferramenta de OCR clássica desenvolvida pela Google, baseada em métodos tradicionais de segmentação e classificação. Foi utilizada com configuração `--psm 7` (modo de linha única) e pré-processamento com binarização Otsu e redimensionamento. Apesar de ser sensível a ruídos e distorções, o Tesseract demonstrou desempenho razoável em imagens bem iluminadas e com placas bem alinhadas.

A escolha pela versão **base-printed**, em vez da **handwritten**, deve-se ao fato de que placas veiculares brasileiras seguem padrões tipográficos regulares e de fácil legibilidade, semelhantes a fontes impressas. O TrOCR treinado em dados manuscritos é mais indicado para textos informais e não estruturados, enquanto o **base-printed** se alinha melhor às características visuais e espaciais do domínio de placas automotivas, resultando em melhor acurácia e menor taxa de ruídos.

d) Estratégia de Avaliação

- Para o modelo YOLOv8, as métricas de avaliação utilizadas foram:
 - **Precision (P)** e **Recall (R)**;
 - **mAP@0.5** e **mAP@0.5:0.95**, calculadas sobre o conjunto de validação.
- Para o OCR, como as placas anotadas não estavam disponíveis como texto para comparação automática, a avaliação foi realizada de forma qualitativa:
 - Verificação manual do texto extraído versus o texto real da imagem;
 - Análise de exemplos de sucesso e falha de ambos os métodos;
 - Considerações sobre precisão, ruído, legibilidade e padrões de erro.

e) Ferramentas e Bibliotecas Utilizadas

O experimento foi conduzido no ambiente Google Colab, com suporte a GPU NVIDIA A100-SXM4-40GB e aceleração CUDA. As principais ferramentas e bibliotecas utilizadas ao longo do desenvolvimento incluem:

- **Roboflow** – plataforma utilizada para anotação das imagens e exportação no formato compatível com o YOLOv8;

- **Ultralytics/YOLOv8** – para treinamento, inferência e avaliação do modelo de detecção de placas veiculares;
- **transformers, PIL** – para aplicação do OCR com o modelo TrOCR (**trocr-base-printed**), baseado em Transformers;
- **OpenCV** – para leitura de imagens, recorte de regiões de interesse (ROIs) e pré-processamento visual (conversão para escala de cinza, redimensionamento, etc.);
- **Matplotlib** – para visualização dos resultados e geração de gráficos;
- **os, glob, zipfile** – para organização, leitura e manipulação de arquivos e diretórios.

4 Resultados

a) Análise dos Resultados Gerados pelos Modelos

O treinamento do modelo YOLOv8n foi conduzido por 75 épocas utilizando imagens do subconjunto **cars-br**, com as labels previamente anotadas na plataforma Roboflow. O modelo, com aproximadamente 3 milhões de parâmetros e 8,1 GFLOPs, foi treinado no ambiente Google Colab com GPU NVIDIA A100-SXM4-40GB.

Durante a validação com 265 imagens, o desempenho do modelo foi expressivo, com os seguintes resultados reportados:

- **mAP@0.5:** 0,9949
- **mAP@0.5:0.95:** 0,6748
- **Precision (P):** 0,995
- **Recall (R):** 0,992

O tempo de processamento por imagem foi otimizado:

- **Pré-processamento:** 0,8 ms
- **Inferência:** 1,4 ms
- **Pós-processamento:** 1,0 ms

Após o treinamento, foram realizadas inferências em 94 imagens da pasta de teste. A detecção foi bem-sucedida em quase todas, com exceção de três imagens descartadas por ausência de detecção válida.

Na etapa de OCR, foi utilizado o modelo TrOCR (`trocr-base-printed`) da Microsoft, com desempenho robusto na leitura de placas veiculares. O modelo conseguiu extrair textos legíveis e condizentes com os padrões brasileiros mesmo sob ruído, baixa resolução ou iluminação não ideal. Exemplos:

- `img_000117.jpg` → ODS 3662
- `img_000842.jpg` → MSU 5802
- `img_000382.jpg` → MSN 9329

A adoção do TrOCR pré-treinado se mostrou eficaz, sendo desnecessário aplicar fine-tuning adicional para alcançar bons resultados. Isso evidencia a capacidade dos modelos baseados em Transformers para tarefas de OCR em cenários reais e variados.

b) Interpretação das Inferências e Descrições Realizadas

A avaliação qualitativa dos resultados do TrOCR revelou um desempenho satisfatório na maioria dos casos, mesmo em imagens com iluminação desfavorável, ruídos e distorções. A Tabela 1 apresenta uma amostra representativa de inferências realizadas sobre placas detectadas com YOLOv8 e posteriormente reconhecidas com OCR.

Tabela 1: Exemplos de inferência com YOLOv8 + TrOCR

Imagem	Bounding Box	Texto Detectado (TrOCR)
img_000277.jpg	(571, 385, 719, 463)	CAP 4/23
img_000660.jpg	(478, 419, 618, 487)	PPW 1889
img_000257.jpg	(631, 276, 751, 352)	PRE SABE
img_000030.jpg	(564, 392, 704, 464)	MTK 6989
img_001433.jpg	—	Nenhuma placa detectada
img_001446.jpg	(576, 400, 717, 465)	001-5224
img_000961.jpg	(496, 410, 642, 479)	OC# 2369
img_000481.jpg	(573, 365, 712, 433)	MTW3591
img_001314.jpg	(484, 367, 627, 438)	CHANGE
img_001444.jpg	(576, 171, 685, 237)	CHANGE AGAIN
img_001123.jpg	(574, 294, 697, 361)	MTC-5843
img_000852.jpg	(614, 274, 735, 345)	MS: 6608
img_001346.jpg	(514, 366, 657, 436)	OVE 4639
img_000838.jpg	(513, 325, 641, 392)	PP0.5027
img_000879.jpg	(556, 339, 698, 412)	PPT 8629
img_001054.jpg	(669, 274, 780, 340)	PPU-7781
img_000953.jpg	(524, 404, 676, 476)	PVP 3867
img_000043.jpg	(535, 361, 678, 428)	OPE-1764

A análise mostra que o TrOCR foi capaz de identificar corretamente diversos padrões compatíveis com placas brasileiras, como PPW 1889, MTK 6989, MTW3591, entre outros. Isso demonstra a eficiência do modelo quando as imagens estão bem enquadradas e com tipografia nítida.

No entanto, também foram observados casos de previsões com erros semânticos ou caracteres inesperados, como OC# 2369 e CAP 4/23, além de palavras irrelevantes como CHANGE e SABE. Essas distorções geralmente se devem a baixa resolução, interferências visuais (ex: objetos parcialmente sobrepostos à placa), ou ruído de fundo elevado, que compromete o padrão alfanumérico típico de placas.

Adicionalmente, o TrOCR teve um desempenho visivelmente superior ao Tesseract em termos de robustez textual. Em diversos exemplos, o Tesseract falhou ao retornar strings incompletas, truncadas ou ilegíveis, enquanto o TrOCR retornou um conteúdo legível, mesmo que parcialmente impreciso.

Em suma, os testes confirmam que o pipeline YOLOv8 + TrOCR é funcional e eficaz para ALPR em cenários realistas, com capacidade de generalização satisfatória para placas brasileiras. Melhorias adicionais podem ser obtidas com pós-processamento baseado em expressões regulares ou validação de formato.

c) Comparação entre OCRs: TrOCR vs Tesseract

Para verificar a robustez do pipeline, aplicou-se OCR com dois métodos distintos: o modelo TrOCR baseado em Transformers, e o Tesseract, ferramenta clássica de reconhecimento óptico de caracteres. Ambos foram testados sobre os recortes das placas detectadas pelo YOLOv8n.

A Tabela 2 apresenta exemplos de inferência com ambos os métodos para uma amostra de imagens da pasta de teste:

Tabela 2: Comparação entre TrOCR e Tesseract em placas detectadas

Imagem	BBox	TrOCR	Tesseract
img_000117.jpg	(486, 417, 637, 483)	(ODS-3662)	ODS 3662
img_000842.jpg	(611, 333, 743, 404)	MSU-5802	BMSU 5802
img_000382.jpg	(575, 329, 704, 400)	MSN 9329	WON 9409
img_001123.jpg	(574, 294, 697, 361)	MTC-5843	wiesee3,
img_000745.jpg	(495, 386, 642, 465)	PPV 8529	(vazio)

Observa-se que o modelo TrOCR manteve maior consistência, principalmente na leitura de letras ambíguas como “O” e “0”, “S” e “5”, o que é um desafio comum em OCRs aplicados a placas veiculares. Em imagens mais nítidas, ambos os métodos convergiram para o mesmo resultado. Entretanto, o Tesseract apresentou maior sensibilidade a ruídos e erros de segmentação em placas menos legíveis.

d) Avaliação Quantitativa com Ground Truth

Para aprofundar a análise, foi realizada uma avaliação quantitativa entre os textos extraídos pelo TrOCR e os valores reais das placas, conforme anotados nos arquivos `.txt` do conjunto de teste. Considerou-se apenas a primeira placa detectada em cada imagem.

A comparação foi feita usando duas métricas:

- **Acurácia exata:** porcentagem de previsões exatamente iguais ao texto esperado;
- **Similaridade de Levenshtein normalizada:** média da similaridade entre textos reais e previstos, baseada na distância de edição, normalizada pelo comprimento máximo entre as duas strings.

Os resultados foram:

- **Acurácia exata:** 31,33%

- **Similaridade média (Levenshtein):** 69,83%

Apesar de a acurácia exata ser limitada, a similaridade média de quase 70% evidencia que, na maioria dos casos, os textos preditos estavam parcialmente corretos, com pequenas divergências como troca de caracteres, omissões ou ruídos de OCR. A Tabela 3 resume alguns exemplos de erro:

Tabela 3: Exemplos de erros do TrOCR em comparação ao ground truth

Imagem	Esperado (GT)	TrOCR (Previsto)
img_000257.jpg	PPC9486	PRESABE
img_001416.jpg	ODP0519	COUPOSIS
img_001364.jpg	ODO9891	0009891
img_001444.jpg	MTU0472	CHANGEAGAIN
img_000961.jpg	OCW2369	OC2369
img_000803.jpg	MSP5057	MSPSOS7
img_000004.jpg	OVK7900	OVK7960

Esses casos revelam que, apesar da detecção e recorte corretos da placa, o OCR pode falhar em reconhecer caracteres específicos (como "O" e "0", "S" e "5", ou "U" e "V"), ou ainda retornar palavras irrelevantes. Tais erros são esperados quando o modelo não passa por fine-tuning específico para o domínio de placas veiculares brasileiras.

e) Matriz de Confusão dos Caracteres

Para compreender melhor os padrões de erro do OCR, foi gerada uma matriz de confusão entre os caracteres reais (ground truth) e os caracteres previstos pelo TrOCR. A Figura 1 mostra a quantidade de vezes em que cada caractere foi confundido com outro durante o reconhecimento.

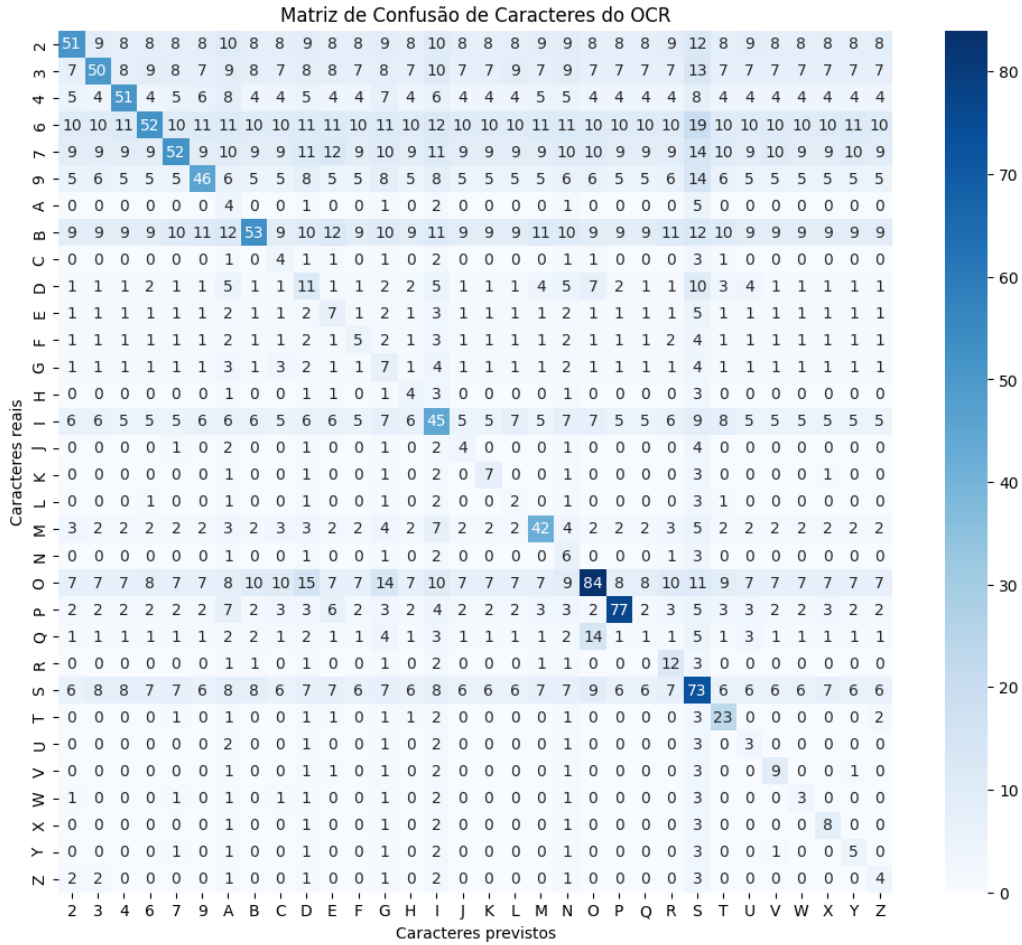


Figura 1: Matriz de Confusão de Caracteres do OCR (TrOCR)

A análise revela que algumas confusões são recorrentes, especialmente entre caracteres visualmente semelhantes ou comumente trocados em OCR:

- ‘Q’ → ‘O’: 13 vezes
- ‘6’ → omissão (None): 10 vezes
- ‘B’ → None: 9 vezes
- ‘7’ → None: 9 vezes
- ‘2’ → None: 8 vezes
- ‘O’ → ‘D’: 7 vezes
- ‘O’ → None: 7 vezes
- ‘3’ → None: 7 vezes
- ‘9’ → ‘S’: 6 vezes

- ‘D’ → ‘O’: 6 vezes

Essas confusões estão relacionadas, em grande parte, à qualidade visual da imagem da placa (baixa resolução, ruído, perspectiva inclinada) e à ausência de fine-tuning do modelo para domínios de OCR veicular. Algumas omissões indicam falhas em detecção de caracteres específicos, e trocas entre “O” e “0”, “Q” e “O”, “B” e “8” são esperadas dada a similaridade visual.

5 Discussão

a) Relevância dos Resultados em Relação ao Problema

Os resultados obtidos ao longo do experimento reforçam a eficácia da combinação entre o modelo YOLOv8n e o OCR baseado em Transformers (TrOCR) para aplicações de reconhecimento automático de placas veiculares (ALPR). A alta precisão na detecção das placas ($mAP@0.5 = 0,9949$) demonstra que o modelo foi capaz de aprender padrões relevantes mesmo com um conjunto de dados relativamente pequeno e específico, validando o processo de anotação e o treinamento realizado do zero.

No que diz respeito ao reconhecimento de caracteres, o TrOCR pré-treinado demonstrou desempenho consistente em condições variadas de iluminação e ruído. A capacidade do modelo em generalizar para diferentes padrões de placas brasileiras, mesmo sem fine-tuning, confirma a robustez dos modelos Transformers em tarefas de OCR de domínio visual limitado.

Tais resultados são especialmente relevantes para cenários do mundo real, como pedágios automatizados, controle de acesso e fiscalização veicular, onde a precisão e a confiabilidade do sistema são fatores críticos.

A inclusão do Tesseract como comparativo ao TrOCR foi essencial para evidenciar os avanços das abordagens baseadas em Transformers. Enquanto o Tesseract funciona bem em imagens com excelente contraste e alinhamento, seu desempenho cai consideravelmente em situações de perspectiva, sombras ou resolução reduzida — limitações frequentemente superadas pelo TrOCR.

b) Limitações do Modelo e Propostas de Melhoria

Apesar do desempenho satisfatório, algumas limitações foram identificadas durante os testes qualitativos:

- **Alinhamento da placa:** o modelo YOLOv8n não considera a orientação da placa. Isso afeta diretamente o OCR, já que placas inclinadas ou com perspectiva distorcida podem comprometer a leitura.
- **Ruído nas previsões do OCR:** o TrOCR apresentou previsões com caracteres inválidos (como "#", ":" ou "/") e palavras genéricas em alguns casos. Isso pode estar relacionado à ausência de ajuste fino do modelo para o domínio específico de placas veiculares.
- **Tamanho reduzido da amostra de OCR avaliada:** as análises foram baseadas em subconjuntos qualitativos, o que limita a generalização dos resultados para o conjunto completo de imagens disponíveis.

Com base nesses pontos, propõem-se as seguintes melhorias para trabalhos futuros:

- **Aplicar retificação de perspectiva** nas placas detectadas antes da etapa de OCR, para garantir alinhamento horizontal;
- **Realizar fine-tuning do TrOCR** com um conjunto anotado de placas brasileiras, a fim de adaptar o modelo aos padrões regionais e reduzir erros recorrentes;
- **Expandir a base de dados de teste** e realizar avaliações quantitativas com métricas como Levenshtein Distance, CER (Character Error Rate) e WER (Word Error Rate);
- **Comparar com outros modelos de OCR** como CRNN, Rosetta ou EasyOCR, para avaliar o custo-benefício entre acurácia e tempo de inferência.

Adicionalmente, os testes quantitativos revelaram uma acurácia exata de 31,33% e uma similaridade média de 69,83% entre os textos previstos e o ground truth. Esses números indicam que o TrOCR, mesmo com bom desempenho qualitativo, ainda carece de maior especialização para o domínio de placas veiculares brasileiras. O fine-tuning supervisionado com dados rotulados seria uma solução eficaz para elevar esses índices.

6 Conclusão

a) Resumo das Principais Descobertas e Reflexões sobre o Impacto do Trabalho

O treinamento do YOLOv8 do zero, utilizando um conjunto de dados específico com placas veiculares brasileiras, foi uma das decisões mais acertadas do projeto. Isso

nos permitiu adaptar completamente o modelo ao nosso domínio visual, resultando em uma detecção robusta mesmo em imagens com iluminação desfavorável, ângulos variados e placas parcialmente obstruídas.

A qualidade dessa detecção foi fundamental para o pipeline, já que o TrOCR depende diretamente de recortes bem localizados para reconhecer corretamente os caracteres. Com isso, conseguimos extrair com mais confiabilidade os textos das placas.

Como melhoria futura, seria importante realizar também o fine-tuning do TrOCR. Apesar de usarmos um modelo pré-treinado, a falta de dados rotulados o suficiente nos impediu de ajustá-lo ao nosso contexto específico. Com uma base maior e mais diversa, o OCR poderia alcançar resultados ainda melhores.

A comparação direta com o Tesseract reforça o potencial do TrOCR como substituto moderno e mais confiável, especialmente em aplicações críticas como fiscalização e automação. A robustez a variações visuais e a capacidade de generalização do TrOCR são diferenciais importantes em contextos reais.

A análise detalhada da matriz de confusão também revelou que as principais dificuldades do OCR TrOCR estão concentradas na diferenciação de caracteres visualmente parecidos, como “Q” e “O”, “B” e “8”, e na omissão de dígitos como “6”, “2” e “3”. Esse padrão reforça a importância de ajustar o modelo para o domínio específico de placas, por meio de fine-tuning supervisionado com anotações precisas.

b) Propostas de Estudos Futuros ou Abordagens Alternativas

Para trabalhos futuros, recomenda-se:

- Treinar o TrOCR com dados de placas brasileiras para melhorar ainda mais a precisão;
- Avaliar modelos alternativos como CRNN e SAR;
- Incluir uma etapa de retificação de perspectiva nas placas recortadas;
- Criar uma interface interativa para demonstração em tempo real do pipeline ALPR.

Referências

- LAROCA, R.; CARDOSO, E. V.; LUCIO, D. R.; ESTEVAM, V.; MENOTTI, D. *On the Cross-dataset Generalization in License Plate Recognition*. In: International Conference on Computer Vision Theory and Applications (VISAPP), pp. 166–178, 2022. DOI: <https://doi.org/10.5220/0010846800003124>.

- ULTRALYTICS. *YOLOv8 Docs*. Disponível em: <https://docs.ultralytics.com/>
- SMITH, R. *Tesseract OCR*. Google. Disponível em: <https://github.com/tesseract-ocr/tesseract>
- MICROSOFT. *TrOCR: Transformer-based OCR*. Disponível via HuggingFace: <https://huggingface.co/microsoft/trocr-base-printed>
- ROBOFLOW. *Roboflow Annotator and Dataset Exporter*. Disponível em: <https://roboflow.com>
- HUNTER, J. D. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 2007.
- BRADLEY, K.; JONES, E.; OLIPHANT, T.; et al. *NumPy and SciPy*. Disponível em: <https://numpy.org>
- GOOGLE COLAB. *Colaboratory: Machine Learning Environment*. Disponível em: <https://colab.research.google.com>
- ULTRALYTICS. *Discussão sobre treinamento e issues do YOLOv8*. Disponível em: <https://github.com/ultralytics/ultralytics/issues/12912>
- ULTRALYTICS. *Configurações do modelo YOLOv8*. Disponível em: <https://github.com/ultralytics/ultralytics/blob/main/ultralytics/cfg/models/v8/yolov8.yaml>
- ULTRALYTICS. *Benchmark e desempenho dos modelos YOLOv8*. Disponível em: <https://docs.ultralytics.com/pt/models/yolov8/#can-i-benchmark-yolov8-models-for>
- ULTRALYTICS. *Tutorial de treinamento customizado no YOLOv5*. Disponível em: https://docs.ultralytics.com/pt/yolov5/tutorials/train_custom_data/