



UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE TUCURUÍ  
FACULDADE DE ENGENHARIA DE COMPUTAÇÃO

BRENO AIRES DA HORA

**SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES  
UTILIZANDO VISÃO COMPUTACIONAL**

TUCURUÍ  
2024

BRENO AIRES DA HORA

**SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES  
UTILIZANDO VISÃO COMPUTACIONAL**

Trabalho de Conclusão de Curso apresentado à Faculdade de Engenharia de Computação, do Campus Universitário de Tucuruí, da Universidade Federal do Pará, como requisito parcial para obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a): Dr. Daniel da Conceição Pinheiro

TUCURUÍ  
2024

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD  
Sistema de Bibliotecas da Universidade Federal do Pará  
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

---

H811s Hora, Breno Aires da.  
Sistema de reconhecimento automático de placas veiculares  
utilizando visão computacional / Breno Aires da Hora. — 2024.  
61 f. : il. color.

Orientador(a): Prof. Dr. Daniel da Conceição Pinheiro  
Trabalho de Conclusão (Graduação) - Universidade Federal do  
Pará, Campus Universitário de Tucuruí, Faculdade de Engenharia  
da Computação, Tucuruí, 2024.

1. visão computacional. 2. conjunto de dados. 3. placas de  
licenciamento veicular. 4. YOLO. 5. OCR. I. Título.

CDD 006.37

---



**UNIVERSIDADE FEDERAL DO PARÁ**  
**CAMPUS UNIVERSITÁRIO DE TUCURUÍ**  
**FACULDADE DE ENGENHARIA DE COMPUTAÇÃO**

**TÍTULO:** SISTEMA DE RECONHECIMENTO AUTOMÁTICO DE PLACAS VEICULARES UTILIZANDO VISÃO COMPUTACIONAL.

**DISCENTE:** Breno Aires da Hora

**MATRÍCULA:** 201833840048

#	BANCA EXAMINADORA	CONDIÇÃO
1	<i>Prof. Dr. Daniel da Conceição Pinheiro</i>	<i>Orientador</i>
2	<i>Prof. Eng. Vigner Vieira dos Santos</i>	<i>Membro interno</i>
3	<i>Eng. Erick Vinícius Damaceno da Silva</i>	<i>Membro externo</i>
4	<i>Eng. Fellipe Augusto Santana Queiroz</i>	<i>Membro externo</i>

**Data da Defesa:** 05/12/2024 | **Hora Início:** 16:04 | **Hora Término:** 16:28

Trabalho Escrito (0 a 10 pontos por critério)	Examinador 1	Examinador 2	Examinador 3	Examinador 4
Formatação	9,00	9,00	9	9
Linguagem (gramática e semântica)	9,00	8,00	9,00	9,00
Conteúdo técnico	8,00	8,00	10	9
Defesa Oral (0 a 10 pontos por critério)	Examinador 1	Examinador 2	Examinador 3	Examinador 4
Sequência lógica de apresentação	10,00	10,00	9,5	9
Administração do tempo	10,00	9,00	10	9
Expressão oral	10,00	10,00	10	9
Domínio do tema	10,00	9,00	9,50	9,00
<b>Média por examinador</b>	9,43	9,00	9,57	9,00
<b>Média Final</b>	9,25			
<b>Conceito Final</b>	EXC			

Tucuruí-Pa, 05 de dezembro de 2024.

Orientador

Membro externo

Membro interno

Membro externo

## AGRADECIMENTOS

Agradeço primeiramente a Deus, pela força, saúde, sabedoria e perseverança concedidas durante esta jornada. Sem Sua presença em minha vida, seria impossível superar os desafios e alcançar esta importante conquista.

Ao meu pai Adilson e minha mãe Rosário, por todo o amor, apoio incondicional e pelos valores que me ensinaram ao longo da vida. Vocês sempre acreditaram no meu potencial e foram minha inspiração para seguir em frente, mesmo nos momentos mais difíceis.

À minha namorada Milca, pelo amor, carinho, paciência e incentivo em cada etapa deste percurso. Sua presença foi um alicerce fundamental durante os momentos de dificuldade e de alegria.

À Universidade Federal do Pará, que proporcionou a oportunidade única de cursar Engenharia de Computação em Tucuruí. Essa oportunidade foi essencial para a construção de conhecimentos e experiências que levarei para toda a vida.

Ao meu professor orientador Dr. Daniel, cuja orientação e dedicação foram imprescindíveis para a realização deste trabalho. Sua expertise e suas palavras de encorajamento foram determinantes para que eu mantivesse o foco e a qualidade do projeto.

À Chyanne, que teve um papel decisivo no início da minha jornada em 2018, ao me avisar que meu nome estava no listão de convocação da universidade. Talvez, sem sua ajuda, eu não teria realizado minha matrícula e, conseqüentemente, não estaria vivenciando este momento de conquista devido ao contexto em que eu estava inserido.

À minha equipe de trabalho, que compartilhou conhecimentos e desafios comigo, contribuindo diretamente para o meu crescimento pessoal e profissional.

Aos meus colegas de faculdade e demais professores, pela amizade, pelas parcerias em projetos e pelas discussões construtivas ao longo do curso. Vocês foram uma parte importante da minha formação e me ajudaram a tornar este percurso mais leve e significativo.

Por fim, agradeço a cada pessoa que, de alguma forma, contribuiu para a minha evolução pessoal, acadêmica ou profissional. Seja por meio de ensinamentos, palavras de apoio ou exemplos de dedicação, todos vocês deixaram marcas importantes nesta trajetória que culmina em mais esta conquista.

## RESUMO

Este trabalho apresenta a detecção e o reconhecimento de placas de identificação veicular com uso de técnicas de visão computacional aplicadas à fiscalização de trânsito. Foi criado um conjunto de dados próprio com placas brasileiras, isso inclui as etapas de gravação, seleção e anotação de imagens, combinado a um conjunto de dados internacional para o treinamento de variantes do modelo YOLO, seguido de uma análise do desempenho geral desses modelos. Além da detecção de placas, o reconhecimento óptico de caracteres (OCR) foi realizado com os modelos EasyOCR e PaddleOCR, enquanto esse último foi o mais eficiente. Os experimentos mostraram que o modelo YOLOv8s-gb superou o YOLOv5su-g em confiança média, sensibilidade média e tempo de processamento. A combinação do PaddleOCR com o YOLOv8s-gb e YOLOv5su-g aplicados a um total de 460 placas, resultou no reconhecimento de 244 e 208, respectivamente, enquanto o EasyOCR reconheceu 118 e 89 placas nos mesmos cenários. O estudo destaca a importância de conjuntos de dados específicos para aprimorar modelos de visão computacional em contextos locais, isso contribui para o avanço do reconhecimento automático de placas veiculares no Brasil.

**Palavras-chave:** visão computacional; conjunto de dados; placas de licenciamento veicular; YOLO; OCR.

## ABSTRACT

This work presents the detection and recognition of vehicle identification plates using computer vision techniques applied to traffic enforcement. A proprietary dataset with Brazilian license plates was created, including the steps of recording, selection and annotation of images, combined with an international dataset for training variants of the YOLO model, followed by an analysis of the overall performance of these models. In addition to license plate detection, optical character recognition (OCR) was performed with the EasyOCR and PaddleOCR models, the latter being the most efficient. The experiments showed that the YOLOv8s-gb model outperformed YOLOv5su-g in average confidence, average sensitivity and processing time. The combination of PaddleOCR with YOLOv8s-gb and YOLOv5su-g applied to a total of 460 license plates resulted in the recognition of 244 and 208, respectively, while EasyOCR recognized 118 and 89 license plates in the same scenarios. The study highlights the importance of specific datasets to improve computer vision models in local contexts, contributing to the advancement of automatic license plate recognition in Brazil.

**Keywords:** computer vision; dataset; vehicle license plates; YOLO; OCR.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Composição de uma imagem RGB .....	15
Figura 2 - Imagem monocromática representada no par de eixos $(x, y)$ .....	16
Figura 3 - Efeito da resolução na qualidade da imagem.....	17
Figura 4 - Identificação de cenário .....	17
Figura 5 – Por dentro de uma rede neural convolucional.....	19
Figura 6 - Detecção de objetos com YOLO .....	20
Figura 7 - Reconhecimento óptico de caracteres com EasyOCR.....	21
Figura 8 – Arquitetura básica de processamento EasyOCR.....	21
Figura 9 - Reconhecimento óptico de caracteres com PaddleOCR.....	22
Figura 10 - Arquitetura básica de processamento PaddleOCR .....	23
Figura 11 - Amostra do conjunto de dados generalista .....	29
Figura 12 – Amostra do conjunto de dados próprio .....	29
Figura 13 - Fluxograma de construção do conjunto de dados próprio .....	30
Figura 14 - Suporte e alerta de gravação .....	32
Figura 15 - Locais de gravação no Google Maps.....	33
Figura 16 - Ferramenta de anotação de objetos.....	34
Figura 17 – Pontuação-F1 em experimentação de hiperparâmetros YOLO .....	38
Figura 18 - Pontuação-F1 de variantes do YOLOv5.....	39
Figura 19 – Velocidade de processamento de lotes de dados.....	40
Figura 20 - Fluxograma de reconhecimento automático de placas de licenciamento .....	41
Figura 21 - Desempenho geral de modelos YOLO .....	43
Figura 22 - Confiança média de detecção de placas de licenciamento .....	44
Figura 23 - Tempo total de predição .....	45
Figura 24 - Exemplo de detecção de placa no local 1 .....	46
Figura 25 – Exemplo de detecção de placas no local 2.....	47
Figura 26 - Total de reconhecimento de placas.....	48
Figura 27 - Reconhecimento de placas de licenciamento: Mercosul e Cinza .....	49
Figura 28 - Reconhecimento de caracteres – YOLOv8s-gb.....	50
Figura 29 - Reconhecimento de caracteres – YOLOv5su-g.....	51
Figura 30 - Matriz de confusão de caracteres EasyOCR.....	52
Figura 31 - Matriz de confusão de caracteres PaddleOCR.....	53

## LISTA DE SIGLAS

AI	<i>Artificial Intelligence</i> (Inteligência Artificial)
ALPR	<i>Automatic License Plate Recognition</i> (Reconhecimento Automático de Placas)
AP	<i>Average Precision</i> (Precisão Média)
AR	<i>Average recall</i> (Sensibilidade Média)
CNN	<i>Convolutional Neural Network</i> (Rede Neural Convolutacional)
COCO	<i>Common Objects in Context</i> (Objetos Comuns em Contexto)
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
CTC	<i>Connectionist Temporal Classification</i> (Classificação Temporal Conexionista)
F1-score	Pontuação-F1
FN	<i>False Negative</i> (Falso Negativo)
FPS	<i>Frames Per Second</i> (Quadros Por Segundo)
FP	<i>False Positive</i> (Falso Positivo)
IoU	<i>Intersection Over Union</i> (Interseção Sobre a União)
JSON	<i>JavaScript Object Notation</i> (Notação de Objetos de JavaScript)
LSTM	<i>Long Short-Term Memory</i> (Memória de Longo e Curto Prazo)
ML	<i>Machine Learning</i> (Aprendizado de Máquina)
OCR	<i>Optical Character Recognition</i> (Reconhecimento Óptico de Caracteres)
P	<i>Precision</i> (Precisão)
R	<i>Recall</i> (Sensibilidade)
RAM	<i>Random Access Memory</i> (Memória de Acesso Aleatório)
ReLU	<i>Rectified Linear Unit</i> (Unidade Linear Retificada)
ResNet	<i>Residual Neural Network</i> (Rede Neural Residual)
TP	<i>True Positive</i> (Verdadeiro Positivo)
TPU	<i>Tensor Processing Unit</i> (Unidade de Processamento de Tensor)
TN	<i>True Negative</i> (Verdadeiro Negativo)
YOLO	<i>You Only Look Once</i> (Você Só Olha Uma Vez)

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>11</b>
<b>1.1</b>	<b>Contextualização do problema.....</b>	<b>11</b>
<b>1.2</b>	<b>Justificativa.....</b>	<b>12</b>
<b>1.3</b>	<b>Objetivos.....</b>	<b>13</b>
1.3.1	Objetivo geral.....	13
1.3.2	Objetivos específicos.....	13
<b>1.4</b>	<b>Estrutura do trabalho.....</b>	<b>14</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS.....</b>	<b>15</b>
<b>2.1</b>	<b>Referencial teórico.....</b>	<b>15</b>
2.1.1	Imagem digital.....	15
2.1.2	Visão Computacional.....	17
2.1.3	Redes Neurais Convolucionais.....	18
2.1.4	Detecção de objetos com YOLO.....	19
2.1.5	Reconhecimento óptico de caracteres.....	20
2.1.6	Métricas de avaliação de desempenho.....	23
2.1.7	Padrão de placas veiculares do Brasil.....	25
<b>2.2</b>	<b>Trabalhos relacionados.....</b>	<b>25</b>
<b>3</b>	<b>METODOLOGIA.....</b>	<b>28</b>
<b>3.1</b>	<b>Conjuntos de dados.....</b>	<b>28</b>
3.1.1	Conjunto de dados generalista.....	28
3.1.2	Conjunto de dados próprio.....	29
3.1.3	Divisão dos conjuntos de dados.....	35
<b>3.2</b>	<b>Ambiente de execução de algoritmos.....</b>	<b>36</b>
3.2.1	Especificações técnicas.....	36
3.2.2	Linguagem e bibliotecas de algoritmos.....	37
<b>3.3</b>	<b>Modelos de visão computacional.....</b>	<b>37</b>
3.3.1	Detecção de placas de licenciamento.....	37
3.3.2	Reconhecimento de caracteres.....	40
<b>3.4</b>	<b>Processo de detecção e reconhecimento de placas.....</b>	<b>41</b>
3.4.1	Mapa de conversão de caracteres.....	41
<b>4</b>	<b>RESULTADOS E DISCUSSÕES.....</b>	<b>43</b>
<b>4.1</b>	<b>Detecção de placas de licenciamento.....</b>	<b>43</b>

4.1.1	Desempenho geral .....	43
4.1.2	Confiança média .....	44
4.1.3	Tempo total de predição .....	45
4.1.4	Exemplos de detecções .....	46
<b>4.2</b>	<b>Reconhecimento de placas de licenciamento.....</b>	<b>47</b>
4.2.1	Desempenho geral .....	48
4.2.2	Desempenho por caractere.....	50
4.2.3	Matriz de confusão de caracteres.....	52
<b>5</b>	<b>CONCLUSÃO E TRABALHOS FUTUROS.....</b>	<b>55</b>
	<b>REFERÊNCIAS.....</b>	<b>57</b>
	<b>APÊNDICE A – REPOSITÓRIOS DE CÓDIGO-FONTE UTILIZADOS.....</b>	<b>60</b>

## 1 INTRODUÇÃO

O reconhecimento automático de placas de licenciamento (*automatic license plate recognition* - ALPR), tem se consolidado como uma ferramenta crucial no suporte ao crescimento de cidades inteligentes, em áreas como segurança pública, controle de tráfego, monitoramento de estacionamentos e automação de pedágios (Bansal *et al.*, 2024). Este trabalho propõe uma análise do uso de variantes do modelo YOLO (*you only look once*, você só olha uma vez), conhecido por sua eficiência na detecção de objetos em tempo real, combinado a ferramentas de reconhecimento óptico de caracteres (*optical character recognition* - OCR), para reconhecer placas veiculares de forma automatizada.

### 1.1 Contextualização do problema

No Brasil, a crescente frota de veículos impõe desafios significativos à gestão do tráfego e segurança pública. De acordo com o relatório da frota de veículos, o país já conta com uma frota de 122.259.745 veículos, que inclui automóveis, caminhões, motocicletas e outros tipos de transporte rodoviário (SENATRAN, 2024). Essa gestão pode ser otimizada com tecnologias acessíveis e aplicações inteligentes que possam executar tarefas de forma automatizada.

Segundo Ismail *et al.* (2021), a maturidade de aplicações de ALPR tende a ser uma questão crítica quando se trata de desenvolver e testar modelos de detecção e reconhecimento de objetos, enquanto não há ampla pesquisa e divulgação de sistemas desse tipo em regiões onde o desenvolvimento nessa área é emergente. Nesse contexto, é possível afirmar que essa lacuna apresenta tanto um desafio quanto uma oportunidade. A falta de maturidade tecnológica exige maiores esforços de pesquisa e desenvolvimento, mas abre espaço para a criação de soluções inovadoras que possam atender às demandas específicas dessas regiões.

O reconhecimento de placas de licenciamento veicular pode apresentar falhas em condições adversas, como baixa qualidade de imagem, ângulo desfavorável do veículo em relação a câmera, iluminação inadequada, interferência de objetos próximos, entre outros fatores (Kesumah *et al.*, 2024). Ademais, a qualidade do conjunto de dados utilizado para o treinamento do modelo de detecção de placas é determinante para reduzir essas falhas, uma vez que modelos treinados com dados mais variados e realistas tendem a lidar melhor com cenários desafiadores.

O treinamento de modelos de visão computacional é desafiador devido à falta de conjuntos de dados com anotações adequadas, o que eleva tanto os custos quanto o tempo de preparação (Dias *et al.*, 2024). A ausência de dados suficientemente representativos pode resultar em um ajuste excessivo do modelo aos dados de treinamento. Esse problema, conhecido como ajuste excessivo (*overfitting*), implica na incapacidade do modelo de generalizar adequadamente para novas situações, isso compromete a sua eficácia em aplicações do mundo real (Aliferis *et al.*, 2024). A criação de conjuntos de dados representativos se torna, portanto, uma etapa fundamental para garantir o sucesso das implementações em visão computacional.

## 1.2 Justificativa

A fiscalização de infrações de trânsito tradicional depende de um processo manual que exige muitas pessoas para realizar tarefas como conferência visual de imagens, seleção e identificação de veículos, além da busca manual de informações e validação por autoridades competentes. Essa abordagem não apenas é repetitiva e custosa, mas lenta e suscetível a erros diversos (SERPRO, 2018). Dessa forma, é evidente que o uso de métodos manuais se torna cada vez mais impraticável, especialmente diante do aumento do volume de veículos e da necessidade de eficiência operacional.

A disponibilidade de poucos conjuntos de dados públicos voltados para placas veiculares brasileiras, atualmente, representa uma lacuna significativa no desenvolvimento de soluções específicas para o contexto nacional. Essa carência dificulta a adaptação de modelos existentes às particularidades das placas utilizadas no Brasil, isso reforça a importância deste projeto.

A automação aplicada a diferentes áreas, como industrial, mineração e rodoviária, tem se mostrado fundamental para a melhoria e a simplificação de processos operacionais. No contexto rodoviário, por exemplo, sistemas automatizados de monitoramento veicular não apenas aumentam a precisão e a agilidade da fiscalização, mas também contribuem para a segurança viária e a redução de custos operacionais. Em setores como a mineração e a indústria, a integração de tecnologias automatizadas para identificação e rastreamento de veículos ou equipamentos tem permitido maior controle logístico e eficiência produtiva. Esses benefícios evidenciam a importância de expandir o uso de automações também na área de fiscalização de trânsito.

O controle de acesso em ambientes restritos, como condomínios residenciais, estacionamentos privados e áreas empresariais, é uma ferramenta indispensável para otimizar processos e aumentar a segurança. Soluções automatizadas agilizam a identificação de veículos autorizados, registram e monitoram entradas e saídas em tempo real, além de reduzir a possibilidade de acesso não autorizado e os custos operacionais. Nesse contexto, o desenvolvimento de um conjunto de dados específico para placas veiculares brasileiras é essencial para assegurar a eficácia de sistemas adaptados às particularidades locais, isso promove maior confiabilidade e precisão em aplicações voltadas à segurança em ambientes controlados.

### **1.3 Objetivos**

#### **1.3.1 Objetivo geral**

O objetivo principal deste trabalho é construir e aplicar um conjunto de dados com placas de licenciamento de veículos brasileiros, para melhorar o desempenho do modelo YOLO, em suas variantes YOLOv5su, YOLOv6s, YOLOv8s, YOLOv9s e YOLOv10s. Além disso, aplicar e comparar o desempenho no reconhecimento óptico de caracteres de placas com EasyOCR e PaddleOCR.

#### **1.3.2 Objetivos específicos**

- Construir um conjunto de dados com placas brasileiras;
- Treinar o YOLOv5su, YOLOv6s, YOLOv8s, YOLOv9s e YOLOv10s com o conjunto de dados de treinamento e conjunto de dados generalista;
- Analisar e comparar o desempenho geral dos modelos de detecção de placas de licenciamento, com às métricas de precisão, sensibilidade e taxa de detecções corretas baseadas em interseção sobre a união, aplicados ao conjunto de dados de testes;
- Analisar o desempenho geral dos modelos EasyOCR e PaddleOCR no reconhecimento de caracteres de placas, com às métricas de acertos totais e por caractere.

## 1.4 Estrutura do trabalho

Este trabalho está estruturado em cinco capítulos, de maneira a abordar sistematicamente o treinamento e a avaliação de modelos de visão computacional no reconhecimento de placas de licenciamento veicular.

O primeiro capítulo apresenta uma introdução ao problema abordado, o contexto, a justificativa para a realização desta pesquisa, o objetivo principal e os específicos.

O segundo capítulo expõe os conceitos teóricos fundamentais relacionados à imagem digital, visão computacional, redes neurais convolucionais, detecção de objetos com YOLO, e modelos de OCR.

O terceiro capítulo detalha a abordagem metodológica adotada no projeto. Inclui a descrição dos conjuntos de dados utilizados, o ambiente de execução de algoritmos, modelos de visão computacional e o processo de detecção e reconhecimento de placas.

O quarto capítulo apresenta os resultados obtidos a partir da aplicação dos modelos de detecção de objetos e OCR, acompanhados de análises e discussões. Esse capítulo inclui gráficos e quadros que ilustram o desempenho entre as variantes do YOLO e os modelos de OCR utilizados.

Por fim, o quinto capítulo apresenta conclusões do estudo e destaca os resultados alcançados em relação aos objetivos propostos. Além disso, são discutidas as limitações do trabalho, sugestões para futuras pesquisas e melhorias em aplicações de detecção automática de placas de licenciamento.

## 2 REFERENCIAL TEÓRICO E TRABALHOS RELACIONADOS

Neste capítulo, serão abordados os conceitos teóricos fundamentais relacionados a este projeto, além da apresentação de trabalhos correlatos que contribuem para a compreensão do tema.

### 2.1 Referencial teórico

O referencial teórico deste trabalho é baseado em dois pilares principais: imagem digital e visão computacional. A seguir, serão apresentados os conceitos básicos sobre esses pilares, bem como uma visão geral sobre redes neurais convolucionais (*convolutional neural network* - CNN), YOLO (*you only look once*, você só olha uma vez) e arquitetura básica dos modelos de reconhecimento óptico de caracteres (*optical character recognition* - OCR).

#### 2.1.1 Imagem digital

As imagens digitais podem ser monocromáticas (em escala de cinza) ou coloridas, as últimas representadas por três canais: vermelho, verde e azul (*red, green and blue* - RGB). Cada canal armazena informações de cor em uma escala que varia de 0 a 255, isso resulta em milhões de combinações possíveis para cores (Filho e Neto, 1999). A Figura 1 ilustra a composição de uma imagem colorida.

Figura 1 – Composição de uma imagem RGB: Imagem colorida (a), componente R (b), componente G (c) e componente B (d)



Fonte: Filho e Neto (1999).

Nessa figura, a imagem colorida apresentada em (a) é decomposta em suas três componentes primárias: R (b), G (c) e B (d), que representam, respectivamente, as intensidades das cores vermelha (*red*), verde (*green*) e azul (*blue*). Essa decomposição é

essencial em diversas aplicações de processamento digital de imagens, pois possibilita o isolamento e o processamento específico de informações presentes em cada canal, isso otimiza algoritmos que dependem de características cromáticas para realizar tarefas como segmentação, reconhecimento de padrões ou realce de detalhes.

Uma imagem digital pode ser descrita matematicamente por uma função  $f(x, y)$ , onde  $x$  e  $y$  representam qualquer ponto de coordenadas espaciais (finitas e discretas) e seu valor é proporcional ao brilho (nível de cinza em imagens monocromáticas) da imagem nesse ponto (Filho e Neto, 1999). A Figura 2 ilustra a representação visual de uma imagem digital monocromática em seus eixos  $(x, y)$ .

Figura 2 - Imagem monocromática representada no par de eixos  $(x, y)$



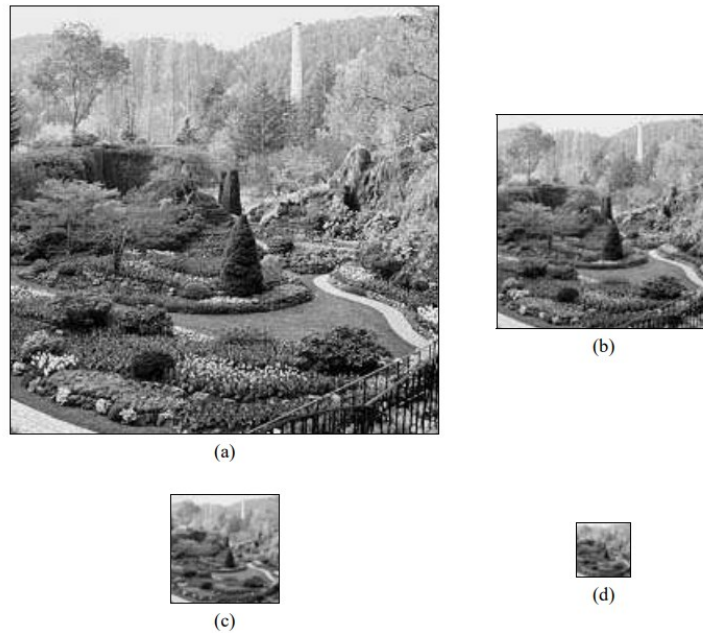
Fonte: Filho e Neto (1999).

Segundo Filho e Neto (1999), o processo de digitalização de uma imagem é caracterizado pela amostragem do sinal analógico em uma matriz de  $M$  por  $N$  pontos, cada qual denominado pixel, como demonstra a equação (1).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix} \quad (1)$$

Gonzalez e Woods (2018), afirmam que quanto maiores forem os valores de  $M$  e  $N$  na matriz da equação (1), maior será a resolução da imagem, conseqüentemente isso elevará a qualidade e o detalhe da imagem digitalizada. A Figura 3 ilustra esse efeito.

Figura 3 - Efeito da resolução na qualidade da imagem: 256x256 (a), 128x128 (b), 64x64 (c) e 32x32 (d)



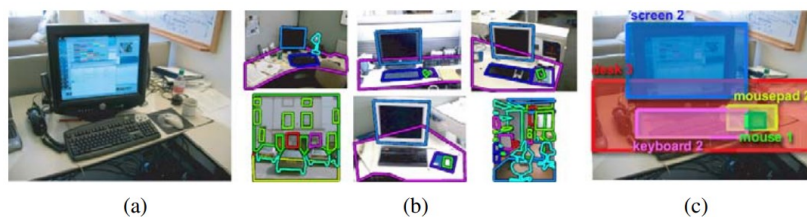
Fonte: Filho e Neto (1999).

A Figura 3 apresenta diferentes resoluções de uma mesma imagem, onde essas resoluções diminuem e resultam em uma perda gradativa de detalhes. Esse efeito demonstra como a redução dos valores de  $M$  e  $N$  na matriz impacta diretamente a nitidez e a capacidade de identificar elementos finos na imagem.

### 2.1.2 Visão Computacional

Como um ramo da inteligência artificial, que possui o objetivo de emular a inteligência humana, a visão computacional propõe o uso de computadores para emular a visão humana, inclusive com a capacidade de aprender e fazer inferências (Gonzalez e Woods, 2018). A Figura 4 ilustra a identificação do contexto em uma cena.

Figura 4 - Identificação de cenário: Imagem de entrada (a), imagens com cenários similares conhecidos (b) e classificação final da imagem de entrada (c)



Fonte: adaptado (Russel *et al.*, 2007).

Conforme ilustrado na Figura 4, o algoritmo de visão computacional processa as informações visuais de forma semelhante à percepção humana, com a segmentação e

classificação automática dos elementos presentes na cena, como o teclado, mouse e monitor. As imagens (b) e (c) exemplificam um método que permite ao sistema definir os limites e categorias dos objetos, através de cenários similares conhecidos, de modo a favorecer a compreensão do contexto. Esse processo é fundamental para aplicações avançadas, como reconhecimento de padrões, veículos autônomos e robótica, onde a precisão na detecção e interpretação do ambiente impacta diretamente na execução de tarefas automatizadas.

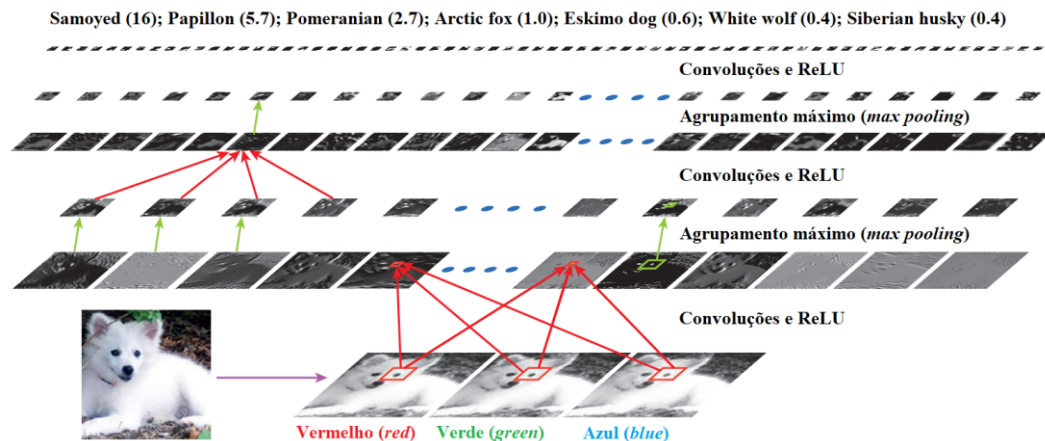
### 2.1.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais são especialmente eficazes em aprender características altamente abstratas de objetos, especialmente dados espaciais, isso os identifica de maneira eficiente. Um modelo profundo de CNN consiste em várias camadas de processamento que aprendem diferentes características dos dados de entrada, como imagens, em diversos níveis de abstração. As primeiras camadas extraem características de alto nível, enquanto as camadas mais profundas aprendem características de baixo nível com maior abstração (Ghosh *et al.*, 2020).

Segundo Lecun (2015), a convolução é uma forma especializada de operação linear, mas com uma diferença importante: em vez de multiplicar todos os pesos por todas as entradas, pequenos filtros percorrem a entrada aos poucos. Essa abordagem permite que a Rede Neural Convolucional capture características locais, como bordas, texturas e outras estruturas importantes dentro da imagem. Dessa forma, a convolução facilita o aprendizado de padrões visuais que são cruciais para tarefas como reconhecimento de imagens e detecção de objetos. Ao aplicar a convolução, a CNN consegue reduzir o número de parâmetros, de modo a tornar o processo mais eficiente ao mesmo tempo em que mantém a capacidade de identificar características em diferentes regiões da imagem.

A Figura 5 ilustra a saída de cada camada de uma arquitetura típica de CNN aplicada à imagem de um cachorro da raça Samoieda (canto inferior esquerdo). O fluxo de informações nessa figura ocorre de baixo para cima, através de várias camadas de convolução e *ReLU* (*rectified linear unit*, unidade linear retificada), onde cada camada detecta diferentes características da imagem.

Figura 5 – Exemplo de processamento em uma rede neural convolucional



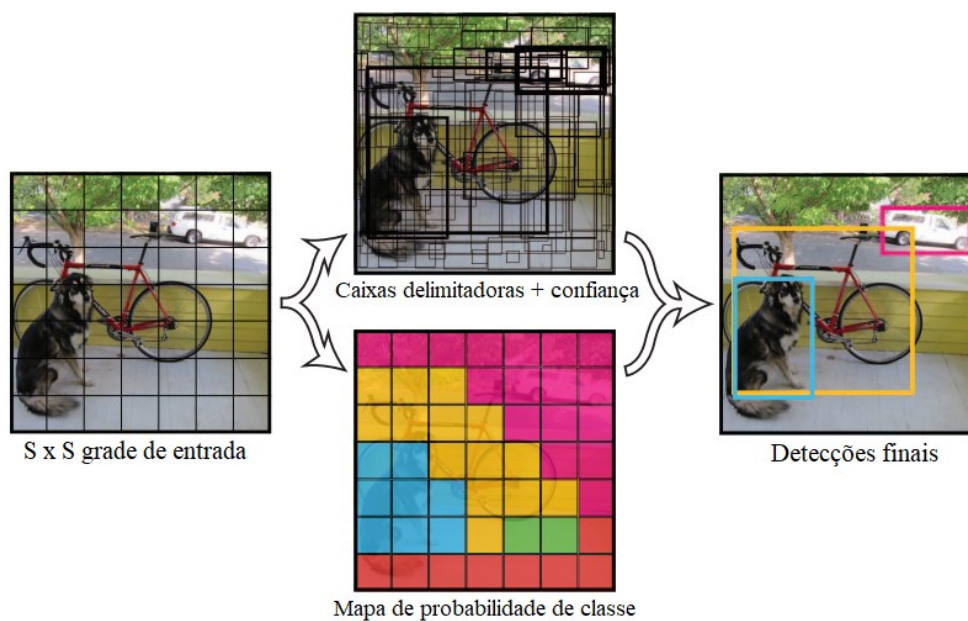
Fonte: adaptado (Lecun *et al.*, 2015).

No início do processamento, as camadas inferiores da rede neural convolucional identificam características mais simples da imagem, como bordas orientadas. À medida que a informação avança para as camadas mais profundas, a rede captura características mais complexas e abstratas, isso permite uma representação mais detalhada do objeto na imagem. Os mapas de características (imagens retangulares) mostram a saída de cada camada e representam as diferentes características aprendidas em várias posições da imagem original. Por fim, a CNN computa um valor de pontuação para classificar a imagem em diferentes categorias, que estão ilustradas na região superior da Figura 5, acompanhadas das respectivas pontuações atribuídas para cada classe. Com isso, a classe que possui a maior pontuação representa a classe predita com maior confiabilidade.

#### 2.1.4 Detecção de objetos com YOLO

A arquitetura YOLO é uma abordagem inovadora para a detecção de objetos que se destaca por sua capacidade de realizar a detecção em tempo real. Ao contrário de métodos anteriores que faziam a detecção em várias etapas, o YOLO realiza a detecção e a classificação simultaneamente. A imagem é dividida em uma grade e previsões são realizadas em cada célula da grade (Redmon *et al.*, 2016). A Figura 6 demonstra essa característica, que permite uma detecção mais rápida e eficiente.

Figura 6 - Detecção de objetos com YOLO



Fonte: adaptado (Redmon *et al.*, 2016).

A imagem de entrada é dividida em uma grade de tamanho  $S \times S$ . Em cada uma das células dessa grade, o modelo gera previsões que incluem caixas delimitadoras e valores de confiança. Além disso, o modelo cria o mapa de probabilidade de classe, onde para cada célula é definido uma classe provável para o objeto. Por fim, nas detecções finais, as caixas delimitadoras são ajustadas para marcar com precisão os objetos identificados, nesse caso, a bicicleta, o cachorro e o carro. Essa abordagem permite ao YOLO não apenas detectar, mas também classificar simultaneamente os objetos, com um processo de detecção mais rápido e eficiente em tempo real.

Variantes de modelos do YOLO, oferecem melhorias contínuas em termos de precisão e velocidade, com ajustes na arquitetura e nos hiperparâmetros do modelo para lidar melhor com diferentes tipos de dados e condições de captura (ULTRALYTICS, 2024). Além disso, novas integrações também são implementadas com sistemas de terceiros, essas integrações facilitam a análise de modelos. Por isso, a escolha da variante apropriada é crucial para garantir um desempenho ideal, especialmente em contextos como o reconhecimento de placas veiculares, onde a precisão e a velocidade são fundamentais.

### 2.1.5 Reconhecimento óptico de caracteres

A Figura 7 demonstra a capacidade do modelo EasyOCR para o reconhecimento óptico de caracteres, através de uma imagem que contém placas de sinalização que indicam

direções para importantes pontos turísticos em Paris, França, como o Museu do Louvre e o Teatro do Palácio Real.

Figura 7 - Reconhecimento óptico de caracteres com EasyOCR

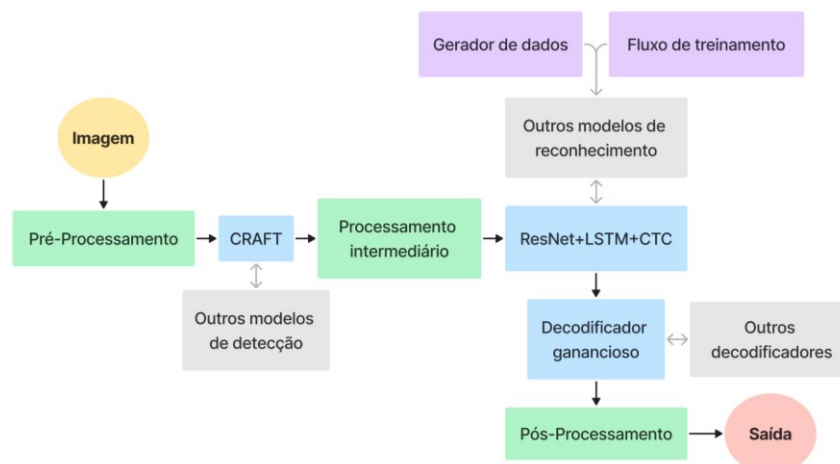


Fonte: adaptado (JAIDEDAI, 2024).

Nessa imagem, o modelo EasyOCR faz o reconhecimento do texto em cada placa, faz a segmentação das palavras da primeira placa superior em duas partes, enquanto na última placa o modelo realiza o reconhecimento da quebra de linha com dois segmentos de texto.

A Figura 8 demonstra o fluxo principal de processamento do modelo EasyOCR.

Figura 8 – Arquitetura básica de processamento EasyOCR



Fonte: adaptado (JAIDEAI, 2024).

O EasyOCR utiliza uma arquitetura que combina diferentes componentes para realizar a detecção e o reconhecimento de caracteres de maneira eficiente. Ele emprega uma ResNet (*Residual Neural Network*, Rede Neural Residual) para extrair características das imagens,

seguida por uma LSTM (*Long Short-Term Memory*, Memória de Longo e Curto Prazo) que modela a sequência dos caracteres detectados, e, por fim, um módulo CTC (*Connectionist Temporal Classification*, Classificação Temporal Conexionista), responsável por alinhar as predições aos caracteres reais sem a necessidade de segmentação prévia. Essa integração simplifica o processo de reconhecimento óptico, torna o EasyOCR uma ferramenta prática e precisa. Além disso, sua facilidade de uso e implementação faz com que seja amplamente utilizado em projetos de ALPR e outras aplicações que exigem OCR.

Segundo Du *et al.* (2020), o PaddleOCR foi otimizado para fornecer suporte a uma ampla gama de idiomas, além de lidar com tarefas de detecção e reconhecimento de texto em tempo real. A Figura 9 demonstra a capacidade do PaddleOCR na tarefa de reconhecimento de texto em imagem, vale destacar que essa imagem é idêntica à da Figura 8.

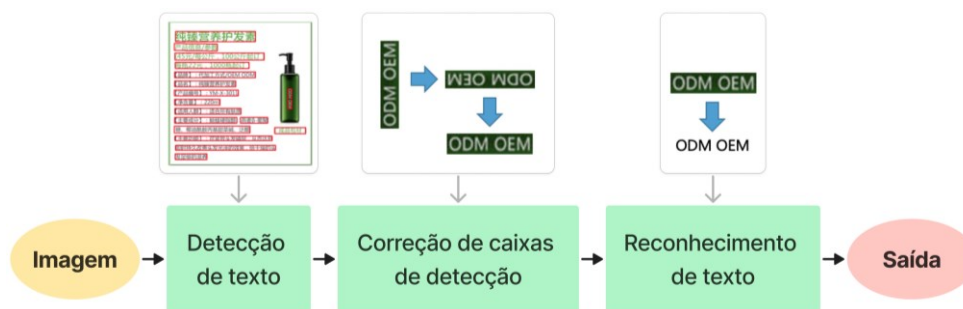
Figura 9 - Reconhecimento óptico de caracteres com PaddleOCR



Fonte: adaptado (Du *et al.*, 2020).

Nessa figura, é possível observar as caixas delimitadoras (*bounding boxes*) de texto, que fazem a segmentação entre linhas identificadas. Desse modo, o reconhecimento é confiável, no sentido de que o resultado é organizado de maneira mais precisa de acordo com a organização do texto original. A Figura 10 ilustra a estrutura principal do PaddleOCR.

Figura 10 - Arquitetura básica de processamento PaddleOCR



Fonte: adaptado (Du *et al.*, 2020).

Essa arquitetura modular permite uma integração com componentes personalizados, por exemplo com o uso de outros algoritmos de detecção de texto, como binarização diferenciável (*differentiable binarization*). Isso possibilita personalizar e melhorar a precisão em cenários desafiadores, como textos em ângulos complexos ou com fontes variadas. Essa flexibilidade e eficiência fazem do PaddleOCR uma solução robusta para aplicações industriais, como sistemas ALPR e outras implementações de OCR em grande escala.

#### 2.1.6 Métricas de avaliação de desempenho

Nesta seção, serão apresentadas as principais métricas ao avaliar o desempenho para modelos de detecção de objetos.

Segundo Gray *et al.* (2011), ao classificar a predição de um modelo de detecção de objetos, é possível aplicar a classificação binária (*binary classification*), que consiste em apenas uma destas quatro classes:

1. Verdadeiro positivo (*true positive* - TP), ocorre quando um dado rotulado como positivo é predito corretamente como positivo;
2. Verdadeiro negativo (*true negative* - TN), ocorre quando um dado rotulado como negativo é predito corretamente como negativo;
3. Falso positivo (*false positive* - FP), ocorre quando um dado rotulado como negativo é predito incorretamente como positivo;
4. Falso negativo (*false negative* - FN), ocorre quando um dado rotulado como positivo é predito incorretamente como negativo.

Esses valores podem ser organizados, por exemplo, em uma matriz de confusão e obter outras métricas que representem o desempenho do modelo.

A equação (2) representa a precisão (*precision* - P), que mede a proporção de verdadeiros positivos (TP) em relação a todas as predições positivas (TP + FP):

$$P = \frac{TP}{TP + FP} \quad (2)$$

A equação (3) representa a sensibilidade (*recall* - R), que mede a proporção de verdadeiros positivos (TP) em relação ao total de verdadeiros (TP + FN):

$$R = \frac{TP}{TP + FN} \quad (3)$$

A equação (4) representa a pontuação-F1 (*F1-score*), que é a média harmônica da precisão (P) e da sensibilidade (R), isso considera tanto a taxa de verdadeiros positivos (TP) quanto a minimização de falsos negativos (FN) e falsos positivos (FP):

$$F1 = \frac{2 \times P \times R}{P + R} \quad (4)$$

A equação (5) representa a precisão média (*average precision* - AP), que mede a proporção da somatória das precisões (P) em relação ao total de predições (N):

$$AP = \frac{1}{N} \sum_{i=1}^N P(i) \quad (5)$$

A equação (6) representa a sensibilidade média (*average recall* - AR), que mede a proporção da somatória das sensibilidades (R) em relação ao total de predições (N):

$$AR = \frac{1}{N} \sum_{i=1}^N R(i) \quad (6)$$

#### 2.1.6.1 Interseção sobre a união

A interseção sobre a união (*intersection over union* - IoU), é uma métrica crucial na avaliação de algoritmos de detecção de objetos. Segundo Everingham *et al.* (2010), o IoU mede a sobreposição entre a caixa delimitadora (*bounding box* - B) predita (*predicted* - p) e a caixa delimitadora verdadeira fundamental (*ground truth* - gt), isso é definido como a razão entre a área da interseção e a área da união das duas caixas. A equação (7) representa o IoU:

$$IoU = \frac{\text{área}(B_p \cap B_{gt})}{\text{área}(B_p \cup B_{gt})} \quad (7)$$

Essa métrica fornece uma maneira eficaz de quantificar a precisão da detecção, com um valor de IoU maior que um limiar pré-estabelecido (geralmente 0,5), isso indica que a

detecção é verdadeiramente positiva. O IoU é especialmente importante em sistemas onde a precisão da localização do objeto deve ser a maior possível, como em reconhecimento automático de placas de licenciamento.

### 2.1.7 Padrão de placas veiculares do Brasil

Segundo a Resolução Nº 729, de 06 de março de 2018, do Conselho Nacional de Trânsito (CONTRAN), o sistema de placas de identificação veicular do Brasil adota o padrão Mercosul para veículos registrados no território nacional. Nesse contexto, determina que a sequência de caracteres nas placas do tipo Mercosul deve seguir o padrão composto por 7 (sete) caracteres alfanuméricos, em alto relevo, na sequência “LLLNLNN”, onde “L” representa uma letra e “N” um número. Além disso, placas do padrão antigo (cinza), seguem a sequência “ABC1234”.

## 2.2 Trabalhos relacionados

Nesta seção serão apresentados e discutidos alguns estudos prévios que contribuíram para o treinamento e aplicações de reconhecimento automático de placas de licenciamento (*automatic license plate recognition* - ALPR). Essa análise é fundamental para situar o presente trabalho no contexto científico e tecnológico. Diversos pesquisadores têm utilizado técnicas de Visão Computacional e Aprendizado Profundo para aprimorar a detecção e o reconhecimento de placas, ao explorar diferentes abordagens. A seguir, essas contribuições, com seus principais resultados obtidos e suas implicações para o aprimoramento dos sistemas ALPR em cenários desafiadores, serão apresentados.

Rahman *et al.* (2021) apresentaram um sistema de detecção e reconhecimento de placas veiculares em tempo real para veículos de Bangladesh. O sistema proposto utilizou o modelo YOLOv4 para a detecção, um algoritmo baseado em grafos para segmentação, e um modelo de rede neural convolucional (CNN) para o reconhecimento dos caracteres. O trabalho destacou a criação de um conjunto de dados robusto com 5.087 imagens, alcança 99,89% de precisão na detecção e 99,33% no reconhecimento. A pesquisa também buscou enfrentar cenários desafiadores, como baixa resolução e iluminação inadequada.

Vedhaviyassh *et al.* (2022) conduziram uma análise comparativa entre o EasyOCR e o TesseractOCR para reconhecimento automático de placas veiculares. Utilizaram o modelo YOLOv5 para detecção de placas e comparam os resultados dos dois sistemas de OCR, a

pesquisa mostra que o EasyOCR apresentou uma precisão superior a 95%, contra 90% do TesseractOCR. O trabalho destacou o uso de Aprendizado Profundo para aumentar a precisão e a eficiência do reconhecimento em tempo real, isso torna o EasyOCR uma escolha preferível para aplicações de reconhecimento de placas.

Mane *et al.* (2024) propuseram uma análise comparativa entre os modelos YOLOv8 e YOLO-NAS para detecção de placas veiculares, com foco em aplicações de tempo real, como prevenção de crimes e vigilância de tráfego. A pesquisa evidenciou que o YOLO-NAS oferece uma arquitetura otimizada que melhora a precisão em cenários específicos, enquanto o YOLOv8 se destaca pela eficiência em termos de velocidade de inferência. O estudo contribui para a escolha de modelos baseados nas necessidades de precisão ou velocidade, a depender das aplicações, e propõe um balanço entre eficiência computacional e acurácia.

Ismail *et al.* (2021) apresentam o PGTLTP, o primeiro conjunto de dados de placas veiculares da Tunísia, composto por 3.000 imagens capturadas por um robô móvel de segurança. Além disso, os autores testaram o desempenho de um detector baseado no modelo YOLOv4-tiny para realizar a detecção em tempo real das placas no conjunto de dados construído, com resultados de referência para a comunidade de visão computacional.

Srinivasan *et al.* (2024) propõem o uso de uma nova rede YOLOv8 com convoluções deformáveis (DEN-YOLOnet) para a classificação e reconhecimento de placas veiculares em tempo real. O modelo utiliza uma série de técnicas de pré-processamento, como realce de imagem com baixa iluminação e remoção de neblina, para melhorar a qualidade das imagens. Além disso, o modelo apresenta um desempenho superior em relação às versões anteriores do YOLO, com uma precisão geral de 98,9%. Essa solução é adequada para ambientes de tráfego real, devido à sua capacidade de lidar com variações de iluminação, condições climáticas adversas e diferentes formatos de placas.

Hu *et al.* (2023) propuseram uma versão otimizada do YOLOv5s, com uso da rede MobileNetV2 como extrator de características e o módulo de atenção CBAM para focar na área da placa veicular. O algoritmo mostrou-se eficaz na detecção de placas em ambientes complexos, como sob baixa iluminação ou em longas distâncias. Com isso, o modelo alcançou uma precisão média de 96,58% no conjunto de dados CCPD e apresenta um bom equilíbrio entre desempenho e leveza.

Entre os trabalhos relacionados, o estudo de Vedhaviyassh *et al.* (2022) apresenta particular relevância para este projeto, ao explorar a integração entre o modelo YOLOv5 para

a detecção de placas e o reconhecimento óptico de caracteres (OCR) com EasyOCR e TesseractOCR. A análise comparativa realizada destacou a superioridade do EasyOCR em termos de precisão, com um desempenho acima de 95%. No entanto, este trabalho vai além ao realizar uma análise comparativa entre diferentes variantes do YOLO e, principalmente, ao confrontar o desempenho do EasyOCR com o PaddleOCR, isso aprofunda a investigação sobre qual ferramenta pode oferecer resultados melhores em cenários variados.

### 3 METODOLOGIA

Este capítulo descreve as etapas metodológicas adotadas para a realização deste trabalho. A pesquisa segue um enfoque aplicado, com base em implementação prática e análise quantitativa de resultados, alinha-se ao que Moresi (2003) afirma sobre a importância de uma metodologia bem definida. O autor ressalta que uma pesquisa estruturada garante a validade e a confiabilidade dos resultados obtidos.

A abordagem quantitativa foi escolhida para este trabalho devido à necessidade de mensuração objetiva do desempenho dos modelos de Visão Computacional. Essa análise possibilita o uso da média de sensibilidade (*average recall* - AR), média de precisão (*average precision* - AP) e outras métricas de desempenho para comparação entre diferentes modelos de detecção de objetos. A coleta de dados para medir o desempenho aplicado dos modelos foi focada na quantidade de acertos e erros, isso permite a aplicação de métodos estatísticos para a análise geral dos resultados. A identificação de padrões de erros e acertos contribui para um diagnóstico claro das áreas onde os modelos podem ser melhorados, isso também fornece informações sobre a eficácia das abordagens utilizadas.

#### 3.1 Conjuntos de dados

Esta seção apresenta os materiais e técnicas empregados na construção ou aquisição do conjunto de dados generalista e do conjunto de dados próprio, utilizados para compor os dados de treinamento e de testes.

##### 3.1.1 Conjunto de dados generalista

O conjunto de dados generalista é composto por imagens de placas de diferentes países, tipos e condições ambientais, isso possibilita um maior potencial de generalização no aprendizado dos modelos de detecção de objetos. A Figura 11 expõe uma amostra de imagens desse conjunto de dados.

Figura 11 - Amostra do conjunto de dados generalista



Fonte: adaptado (ROBOFLOW, 2024).

O conjunto de dados generalista utilizado neste trabalho é derivado do repositório público "License Plate Recognition" (ROBOFLOW, 2024). Esse repositório contém imagens e anotações obtidas a partir da junção de diversos projetos, isso inclui fontes variadas que capturam placas veiculares em distintos contextos e condições ambientais. Esses dados são essenciais para o treinamento e a validação dos modelos de detecção e reconhecimento de placas no projeto, isso contribui para uma maior generalização em múltiplos cenários.

### 3.1.2 Conjunto de dados próprio

O conjunto de dados próprio é composto por imagens de placas brasileiras (Cinza e Mercosul), isso pode ser útil para fortalecer a generalização do modelo e aprofundar seu aprendizado na detecção de placas de licenciamento veicular do Brasil. A Figura 12 ilustra uma amostra de imagens desse conjunto de dados.

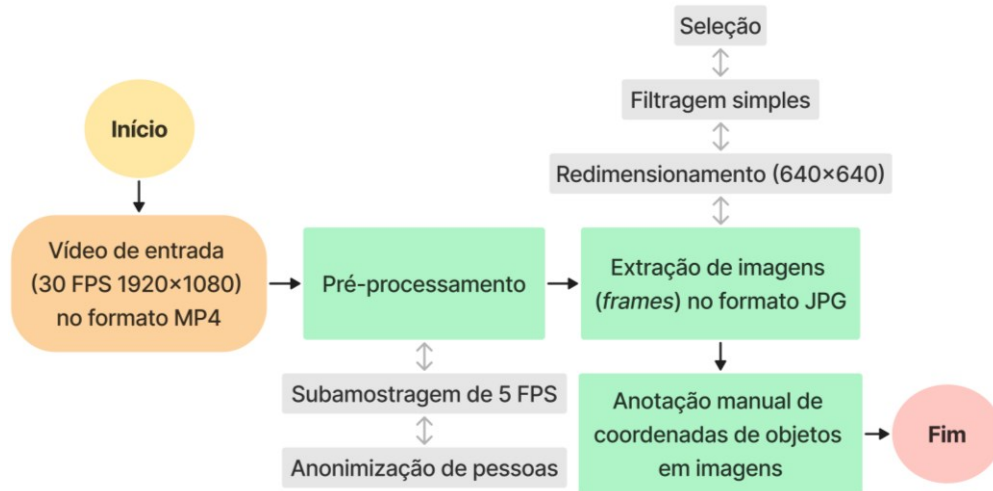
Figura 12 – Amostra do conjunto de dados próprio



Fonte: elaborado pelo autor (2024).

Na Figura 12, é possível observar a diversidade de veículos, ângulos e luminosidades dos locais de gravação para a construção desse conjunto de dados próprio. A Figura 13 expõe o fluxograma de construção desse conjunto de dados.

Figura 13 - Fluxograma de construção do conjunto de dados próprio



Fonte: elaborado pelo autor (2024).

A construção do conjunto de dados inicia-se com a gravação dos objetos de interesse da pesquisa, que são as placas de licenciamento veicular. Em seguida, no pré-processamento realiza-se a subamostragem do vídeo, originalmente gravado a 30 quadros por segundo (*frames per second* - FPS), para uma taxa de 5 FPS, isso visa reduzir a carga de trabalho nas etapas posteriores.

Ainda no pré-processamento, aplica-se a anonimização das pessoas presentes nas imagens, com uso do modelo YOLOv8l (padrão) para detectar e desfocar a região facial. Na extração de imagens do vídeo para o formato JPG, ocorre o redimensionamento de imagem para a resolução 640x640 pixels, essa etapa é crucial para reduzir o tempo de treinamento e melhorar a generalização, uma vez que as imagens do conjunto de dados generalista estão nessa resolução. Além disso, uma filtragem simples, com o uso desse mesmo modelo, é empregada para selecionar somente quadros onde há ao menos um veículo.

Na etapa de seleção, são manualmente escolhidas apenas as imagens que apresentam placas com todos os caracteres completamente visíveis. Somente após esses procedimentos, inicia-se a fase de anotação, que consiste em definir manualmente a região de localização das placas nas imagens selecionadas.

Por fim, realiza-se a divisão do conjunto de dados para distribuir as imagens entre os conjuntos de treinamento, validação e teste, de acordo com as finalidades do estudo.

### 3.1.2.1 Equipamentos e materiais de gravação

O Quadro 1 apresenta as especificações do dispositivo utilizado para a gravação das imagens do conjunto de dados próprio.

Quadro 1 – Características do dispositivo de gravação de imagens

<b>CARACTERÍSTICA</b>	<b>VALOR</b>
MARCA	Xiaomi
LINHA	Redmi
SÉRIE	Note 9 Pro
TIPO DE APARELHO	Smartphone
CÂMERA PRINCIPAL	64 megapixels
RESOLUÇÃO DA CÂMERA PRINCIPAL	4K 30 FPS, 1080p 30/60 FPS
FORMATO DE VÍDEO	MP4
MEMÓRIA RAM	6 GB LPDDR4X
MEMÓRIA ROM	128 GB
PROCESSADOR	64 bits Qualcomm Snapdragon 720G (SM7125) (8nm)
GPU	Qualcomm Adreno 618 (750MHz)
BATERIA	5020 mAh
PESO	209 gramas
SISTEMA OPERACIONAL	Android versão 12, Miui 14

Fonte: adaptado (XIAOMI, 2024).

A câmera e o processador Snapdragon 720G permitem capturar detalhes e evitar perdas de quadros. A bateria de longa duração e a memória de 128 GB suportam sessões extensas de gravação, isso assegura a criação de um conjunto de dados consistente. Embora não se trate do equipamento de mais alta performance, ele foi escolhido com base nas suas características e disponibilidade para o propósito da coleta de dados.

A Figura 14 ilustra o suporte utilizado para apoiar e estabilizar o aparelho de gravação durante a captura das imagens para o conjunto de dados próprio e a arte utilizada para sinalizar aos motoristas sobre a gravação das imagens.

Figura 14 - Suporte e alerta de gravação



Fonte: elaborado pelo autor (2024).

O uso desse suporte de gravação (a) reduz significativamente a ocorrência de tremores e desalinhamentos, comuns em gravações manuais, o que assegura a qualidade dos dados coletados, especialmente em condições de longa duração. Além disso, o uso do suporte permite manter o enquadramento e o ângulo constantes, isso aumenta a uniformidade das gravações e contribui para a confiabilidade do conjunto de dados final. Essa sinalização visual (b) foi projetada para possibilitar que os motoristas estejam cientes de que serão registrados. A arte promove a conscientização e assegura que o processo de captura de imagens seja realizado de maneira ética e transparente.

### 3.1.2.2 Gravação de imagens

Estas localizações ilustradas no Quadro 2, foram escolhidas estrategicamente para garantir a captura de imagens em diferentes áreas da mesma região.

Quadro 2 – Coordenadas dos locais de gravação

COORDENADAS	DESCRIÇÃO
-3.824555, -49.675044	<b>Local 1</b> – Vila Permanente, portaria principal das vilas da Eletronorte, Tucuruí, Pará, Brasil
-3.779717, -49.676377	<b>Local 2</b> – BR 422, frente a EMEF Odineia Leite Caminha, Tucuruí, Pará, Brasil

Fonte: adaptado (Google Maps, 2024)

Essas localidades distintas oferecem uma diversidade de condições ambientais, o que fortalece a robustez dos modelos de detecção de placas e aumenta sua capacidade de generalização para diferentes cenários. Essa variedade permite que os modelos enfrentem variações em iluminação e ângulos, isso resulta em uma maior eficiência ao serem aplicados em ambientes reais. A Figura 15 exibe um mapa que sinaliza esses locais onde as gravações das imagens foram realizadas.

Figura 15 - Locais de gravação no Google Maps



Fonte: adaptado (Google Maps, 2024).

Esse mapa ilustra a distribuição geográfica das diferentes áreas capturadas durante o projeto, oferece uma visão clara dos pontos específicos em que as placas de licenciamento foram filmadas.

Na etapa de gravação, foram definidas configurações cruciais para garantir a qualidade das imagens. O Quadro 3 ilustra essas configurações.

Quadro 3 – Configurações de gravação

CONFIGURAÇÃO	VALOR
VELOCIDADE DO OBTURADOR ( <i>shutter speed</i> )	1/4000 segundos
FOCO	Infinito
ISO	Automático
EXPOSIÇÃO	0
CÂMERA	Larga ( <i>wide</i> )
BALANÇO DE BRANCO ( <i>white balance - WB</i> )	Automático

RESOLUÇÃO	1080p (1920x1080 pixels)
IMAGENS POR SEGUNDO ( <i>frames per second</i> )	30 FPS
ANTI-FAIXAS ( <i>anti banding</i> )	60 Hz

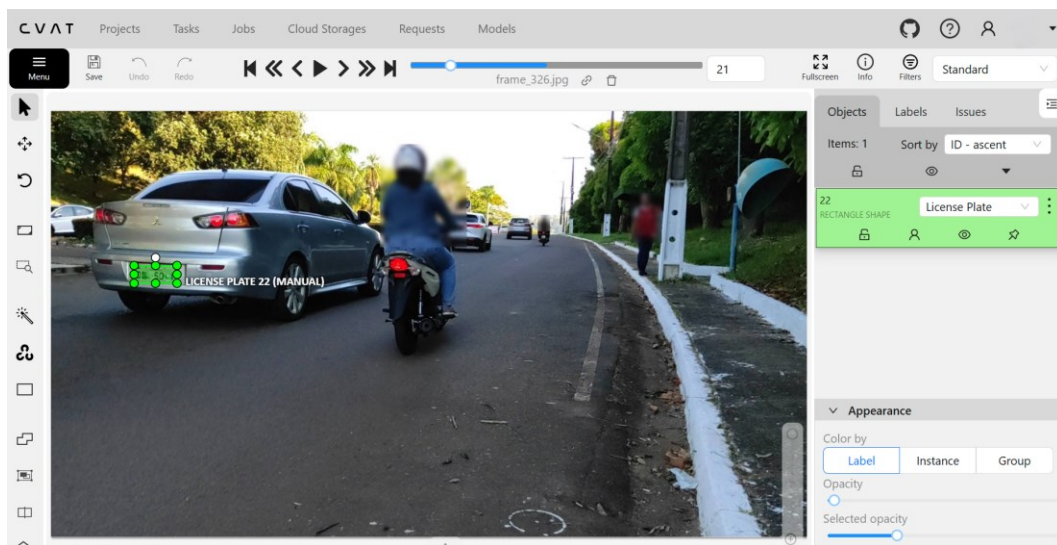
Fonte: elaborado pelo autor (2024).

A velocidade do obturador foi ajustada para 1/4000 segundos, o que foi fundamental para capturar imagens nítidas e evitar borrões, mesmo com os veículos em movimentos rápidos. O foco foi definido para o infinito, isso garante que as placas de licenciamento sejam claramente visíveis e bem definidas. Além disso, a resolução de 1080p assegurou uma alta qualidade de imagem, essencial para a detecção precisa das placas e para capturar detalhes finos necessários para um treinamento eficaz dos modelos.

### 3.1.2.3 Anotação de coordenadas de objetos

As imagens do conjunto de dados próprio foram manualmente anotadas. A Figura 16 ilustra a ferramenta utilizada para realizar a anotação de coordenadas de placas.

Figura 16 - Ferramenta de anotação de objetos



Fonte: adaptado (CVAT, 2024).

Esse processo de anotação gera arquivos de texto que contém as coordenadas das caixas delimitadoras (*bounding boxes*) ao redor das placas de veículos. Além disso, o CVAT (*computer vision annotation tool*, ferramenta de anotação de visão computacional) facilita e acelera o processo de anotação, garante consistência e qualidade nas coordenadas dos objetos, fator essencial para o treinamento eficiente dos modelos de visão computacional.

O processo de gravação, seleção e anotação das imagens que compuseram o conjunto de dados próprio foi extenso, isso exigiu um esforço manual significativo e um investimento considerável de tempo, mesmo com o suporte de ferramentas adequadas. Esse processo incluiu cerca de 5 horas de gravação, distribuídas ao longo de 6 dias distintos, e aproximadamente 360 horas dedicadas à anotação das coordenadas das placas veiculares.

### 3.1.3 Divisão dos conjuntos de dados

Neste projeto, foram utilizados três conjuntos de dados distintos: o conjunto de dados generalista, o conjunto de dados de treinamento (composto por imagens do conjunto de dados generalista e próprio) e o conjunto de dados de testes. Nas subseções seguintes, serão detalhados os conjuntos de dados de treinamento e de testes, a descrição de suas composições e características.

#### 3.1.3.1 Conjunto de dados de treinamento

O Quadro 4 apresenta a divisão do conjunto de dados utilizado para o treinamento dos modelos de detecção de objetos, esse conjunto inclui tanto imagens obtidas do conjunto de dados generalista quanto do conjunto de dados próprio.

Quadro 4 – Divisão do conjunto de dados de treinamento

<b>TIPO</b>	<b>QUANTIDADE (generalista + próprio)</b>	<b>TOTAL (%)</b>
TREINAMENTO	21173 + 1480	87,29%
VALIDAÇÃO	2046 + 232	8,78%
TESTE	1019 + 0	3,93%

Fonte: elaborado pelo autor (2024).

Essa estrutura de divisão é essencial para garantir que o modelo seja treinado de maneira robusta, mantenha a capacidade de generalização e a precisão em cenários do mundo real. A distribuição de imagens do conjunto de dados próprio para validação e treinamento seguiu uma proporção experimental, mas aproximada da original do conjunto de dados generalista. Além disso, não foram adicionadas imagens para teste nesta etapa, com o propósito de dedicar imagens para a etapa de reconhecimento de caracteres no conjunto de dados de testes.

### 3.1.3.2 Conjunto de dados de testes

O conjunto de dados de testes é utilizado para testar as capacidades dos modelos YOLO em detectar placas e dos modelos de OCR em reconhecer os caracteres de placas corretamente. Esse conjunto é composto por 450 imagens, também dos locais de gravação do conjunto de dados próprio, selecionadas e anotadas manualmente para garantir que cada imagem possui placas com todos os caracteres visíveis. Essas anotações seguem o padrão COCO (*common objects in context*, objetos comuns em contexto) de anotações, que é amplamente reconhecido na comunidade de Visão Computacional (Lin *et al.*, 2014). O uso de um formato padronizado como o COCO permite que as avaliações sejam realizadas de maneira consistente e comparável.

## 3.2 Ambiente de execução de algoritmos

O ambiente Kaggle foi utilizado para processar os algoritmos e dados necessários à pesquisa. Essa plataforma web oferece um espaço dedicado à comunidade de inteligência artificial (*artificial intelligence* - AI) e aprendizado de máquina (*machine learning* - ML). A comunidade desse ambiente é formada por estudantes, desenvolvedores, pesquisadores e entusiastas da programação. O Kaggle organiza-se em módulos que incluem conjuntos de dados, notebooks, modelos de AI, cursos e competições, isso promove o desenvolvimento colaborativo e o aprimoramento de habilidades de forma gratuita.

### 3.2.1 Especificações técnicas

O Quadro 5 detalha as especificações técnicas dos recursos disponíveis em notebooks do Kaggle. Esses recursos são fornecidos em sua plataforma web, sem a necessidade do usuário se preocupar com a infraestrutura necessária para o processamento de algoritmos.

Quadro 5 – Especificações gerais do Kaggle

ESPECIFICAÇÃO	DESCRIÇÃO
TEMPO DE EXECUÇÃO	12 horas para sessões com CPU e GPU, 9 horas para sessões com TPU
ESPAÇO EM DISCO (POR SESSÃO)	20 GB (/kaggle/working – caminho da pasta de trabalho)
ESPAÇO EM DISCO (PERMANENTE)	215 GB
TEMPO DE USO DE GPU	30 horas semanais

TEMPO DE USO DE TPU	20 horas semanais
TEMPO DE USO DE CPU	Limitado pelo tempo de execução por sessão (12 ou 9 horas)
MEMÓRIA RAM	30 GB

Fonte: adaptado (Kaggle, 2024)

As sessões que utilizam a TPU 1VM são equipadas com 96 núcleos de CPU e uma memória RAM robusta de 330 GB, isso garante maior capacidade de processamento para tarefas mais intensivas.

### 3.2.2 Linguagem e bibliotecas de algoritmos

No desenvolvimento deste trabalho, foi utilizado a linguagem de programação Python e diversas bibliotecas para a implementação e a análise dos resultados. A Ultralytics foi empregada para a utilização do modelo YOLO na detecção de placas de veículos. Para o reconhecimento óptico de caracteres, foram utilizados o EasyOCR e o PaddleOCR, que auxiliaram na extração de caracteres das placas detectadas. A Numpy facilitou operações matemáticas com listas, enquanto a Matplotlib e a Pandas foram essenciais na visualização e manipulação dos dados. A biblioteca Scipy contribuiu com funções matemáticas e estatísticas adicionais. A OpenCV foi utilizada para processamento de imagens e vídeos, e a Pycocotools para avaliação dos resultados de detecção. Além disso, a biblioteca PIL foi usada para manipulação de imagens, a Wandb para monitoramento de experimentos e treinamentos de modelos, a Json para leitura e escrita de arquivos no formato JSON.

## 3.3 Modelos de visão computacional

Esta seção aborda os modelos de visão computacional utilizados ao longo da pesquisa, abrange tanto os modelos para detecção de placas de licenciamento quanto os de reconhecimento óptico de caracteres (*optical character recognition* - OCR). Cada uma dessas abordagens foi essencial, ao garantir a integração eficiente entre a detecção precisa das placas e a extração confiável das informações contidas nelas.

### 3.3.1 Detecção de placas de licenciamento

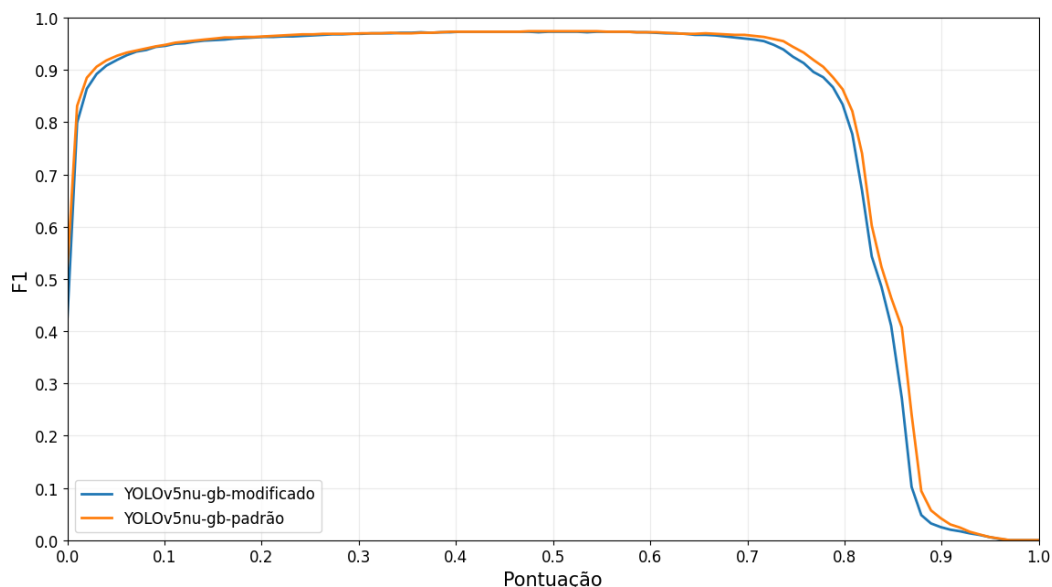
Para a detecção de placas de licenciamento, foram utilizadas variantes do modelo YOLO (*you only look once*, você só olha uma vez). Essa escolha levou em consideração a

ampla documentação e pesquisas disponíveis na literatura, bem como a sua simplicidade de implementação e treinamento.

### 3.3.1.1 Definição de hiperparâmetros

Na definição dos hiperparâmetros dos modelos YOLO, adotou-se uma estratégia iterativa baseada em experimentação. A Figura 17 demonstra que os hiperparâmetros em sua configuração padrão apresentaram uma melhoria suave na Pontuação-F1 em relação a configuração modificada no YOLOv5nu. Essa versão do YOLO foi escolhida devido à sua maturidade e à quantidade significativa de pesquisas na literatura, as quais aplicam essa versão, isso fortalece a escolha da variante ideal para o contexto deste trabalho.

Figura 17 – Pontuação-F1 em experimentação de hiperparâmetros YOLO



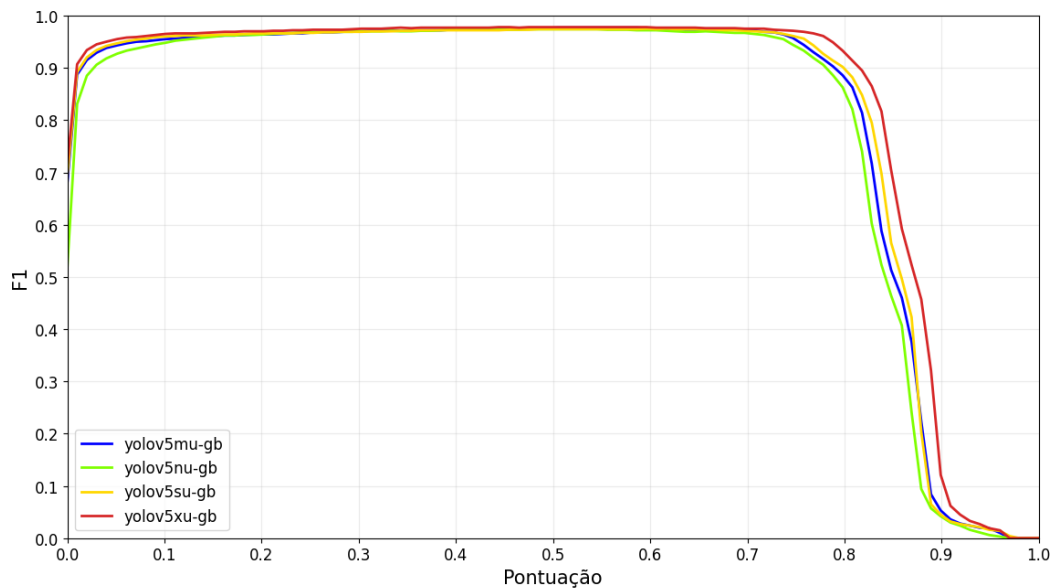
Fonte: elaborado pelo autor (2024).

Inicialmente, foram realizados testes com os valores padrão fornecidos pelos modelos, posteriormente foram ajustados alguns parâmetros, como a taxa de aprendizado, número de épocas, tamanho do lote (*batch size*) e o limite (*threshold*) de confiança para as detecções. Além disso, o monitoramento contínuo dos resultados com a ferramenta W&B (*Weights and Biases*) permitiu rastrear a evolução do modelo durante o treinamento, isso ajudou a identificar as configurações ideais que equilibrassem acurácia e eficiência de processamento para a detecção de placas de veículos.

### 3.3.1.2 Treinamento e validação de modelos

Durante o processo de treinamento e validação dos modelos, o conjunto de dados de treinamento foi utilizado. Cada modelo de detecção de objetos treinado com esse conjunto de dados é identificado pelo sufixo “-gb”. A Figura 18 ilustra a Pontuação-F1 de variantes do YOLOv5. Essa versão do YOLO foi escolhida devido à sua maturidade e à quantidade significativa de pesquisas na literatura. Esta etapa tem como objetivo principal identificar a variante do modelo YOLOv5 mais adequada para o contexto específico deste trabalho. As versões analisadas incluem a "n" (*nano*), "s" (*small*, pequeno), "m" (*medium*, médio) e "x" (*large*, largo). Cada uma dessas variantes apresenta características distintas em termos de complexidade, desempenho e requisitos computacionais. Por fim, a variante que apresentar o melhor desempenho será utilizada também para as demais versões do YOLO utilizadas neste trabalho.

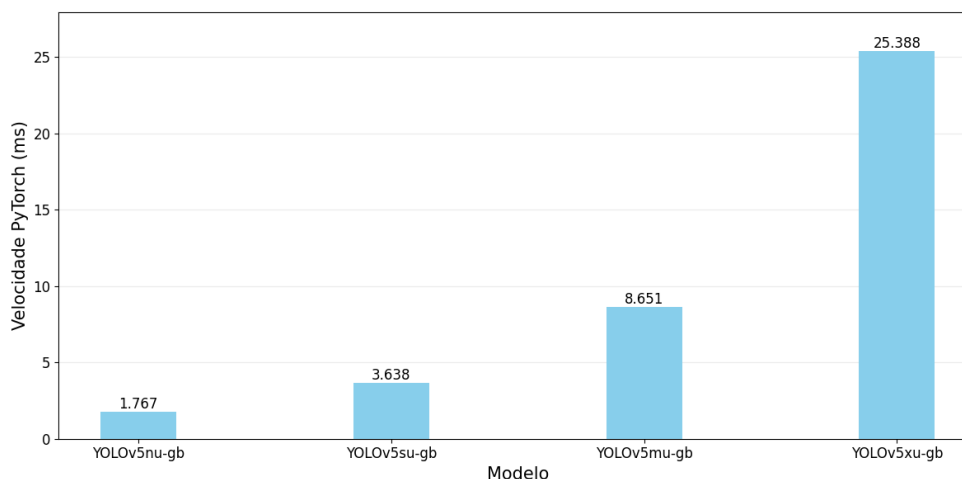
Figura 18 - Pontuação-F1 de variantes do YOLOv5



Fonte: elaborado pelo autor (2024).

O treinamento foi realizado com a configuração padrão dos modelos YOLO. É possível observar que o modelo YOLOv5xu-gb apresenta um desempenho levemente superior aos demais modelos, enquanto o modelo YOLOv5nu-gb apresenta o pior desempenho. Apesar disso, é necessário avaliar o tempo que o modelo leva em média para apresentar esse desempenho. Por isso, a Figura 19 ilustra a velocidade de processamento de lotes de dados (PyTorch) dessas variantes do YOLOv5.

Figura 19 – Velocidade de processamento de lotes de dados



Fonte: elaborado pelo autor (2024).

Nessa figura, é possível observar que o modelo YOLOv5xu-gb necessita de muito mais tempo para executar processos PyTorch, em relação aos demais modelos. Além disso, o modelo YOLOv5su-gb apresentou um desempenho levemente inferior ao YOLOv5xu-gb, como exposto na Figura 19. Portanto, o modelo YOLOv5su-gb destaca-se como o modelo mais promissor, pois equilibra um bom desempenho com um tempo de processamento relativamente curto.

### 3.3.2 Reconhecimento de caracteres

Para o reconhecimento dos caracteres presentes nas placas de veículos, foram empregadas as bibliotecas EasyOCR e PaddleOCR, selecionadas com base em suas características técnicas e adequação ao contexto do projeto. O EasyOCR destacou-se como uma ferramenta ágil e de fácil implementação, oferece suporte a uma ampla gama de idiomas e apresenta uma taxa de reconhecimento satisfatória em diversos cenários. Sua simplicidade de uso foi um fator decisivo para a sua escolha nesta pesquisa.

Em contraste, a PaddleOCR demonstrou um desempenho robusto, esse modelo não se confundiu tanto como o EasyOCR. Essa capacidade foi particularmente vantajosa em situações em que as imagens apresentavam baixa qualidade ou os caracteres estavam parcialmente desfocados. Assim, a PaddleOCR contribuiu significativamente para melhorar a precisão do reconhecimento em condições desafiadoras.

Essa abordagem, possibilitou avaliar qual biblioteca ofereceria o melhor reconhecimento de caracteres quando integrada aos modelos de detecção YOLO. A

comparação foi realizada com base na análise de acerto e erro de reconhecimento, isso proporciona uma visão abrangente sobre a eficácia de cada ferramenta no contexto do reconhecimento automático de placas de licenciamento.

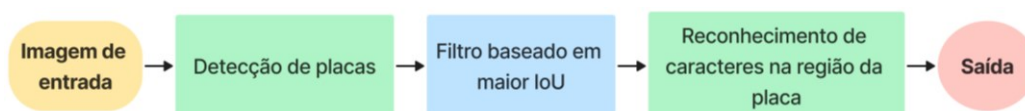
### 3.4 Processo de detecção e reconhecimento de placas

Na literatura, existem diversas técnicas e estratégias para o reconhecimento automático de placas de licenciamento, cuja escolha deve ser guiada pelo propósito e pelas limitações de cada aplicação. Em algumas situações, é necessário seguir uma sequência específica para obter maior precisão, a começar pela identificação dos veículos, seguida pela localização das placas e, finalmente, pelo reconhecimento dos caracteres. Essa abordagem em etapas permite reduzir erros e melhorar a eficácia geral do sistema.

Por outro lado, a depender do *hardware* (componentes físicos) que executa o sistema, cada etapa pode gerar um custo de tempo significativo, o que deve ser considerado no planejamento e na escolha da técnica. Em sistemas com recursos limitados, otimizar a eficiência de cada etapa torna-se essencial para manter o desempenho em tempo real.

A Figura 20, apresenta o processo adotado para o reconhecimento de placas neste trabalho, desde a imagem de entrada e a detecção da região da placa até o reconhecimento dos caracteres com OCR.

Figura 20 - Fluxograma de reconhecimento automático de placas de licenciamento



Fonte: elaborado pelo autor (2024).

Esse processo está limitado à detecção precisa da localização da placa de licenciamento, pois uma detecção imprecisa comprometerá o reconhecimento subsequente. Para evitar que os resultados do OCR sejam afetados por uma detecção incorreta, foi aplicado um filtro baseado no maior valor de IoU, isso assegura que apenas detecções que atendam ao limite mínimo necessário sejam consideradas corretas.

#### 3.4.1 Mapa de conversão de caracteres

Para mitigar erros de predição de caracteres, a aplicação de um mapa de conversão pode ser uma solução eficaz, especialmente no contexto específico do reconhecimento de

placas veiculares. Placas de licenciamento seguem padrões predefinidos, esses padrões permitem a conversão de caracteres confusos, como o número "0" para a letra "O" e vice-versa, de acordo com a posição do caractere na placa. Isso ocorre porque, em determinados índices, sabe-se que apenas números ou letras podem aparecer, e isso possibilita a correção de confusões no reconhecimento.

No entanto, essa estratégia de conversão é limitada a casos em que há uma confusão entre letras e números devido ao formato padrão da placa. Quando a confusão ocorre entre duas letras ou dois números, como "D" e "Q" ou "2" e "9", o mapa de conversão perde sua eficácia, pois esses caracteres podem ocupar a mesma posição na placa sem uma regra clara que os diferencie. Portanto, enquanto o mapa de conversão otimiza a taxa de acertos para caracteres específicos, sua aplicabilidade depende do contexto e das regras predefinidas associadas ao padrão das placas veiculares.

## 4 RESULTADOS E DISCUSSÕES

Esta seção aborda a análise de desempenho dos modelos YOLO (*you only look once*, você só olha uma vez) treinados e do reconhecimento de placas veiculares com os modelos de reconhecimento óptico de caracteres (*optical character recognition* - OCR).

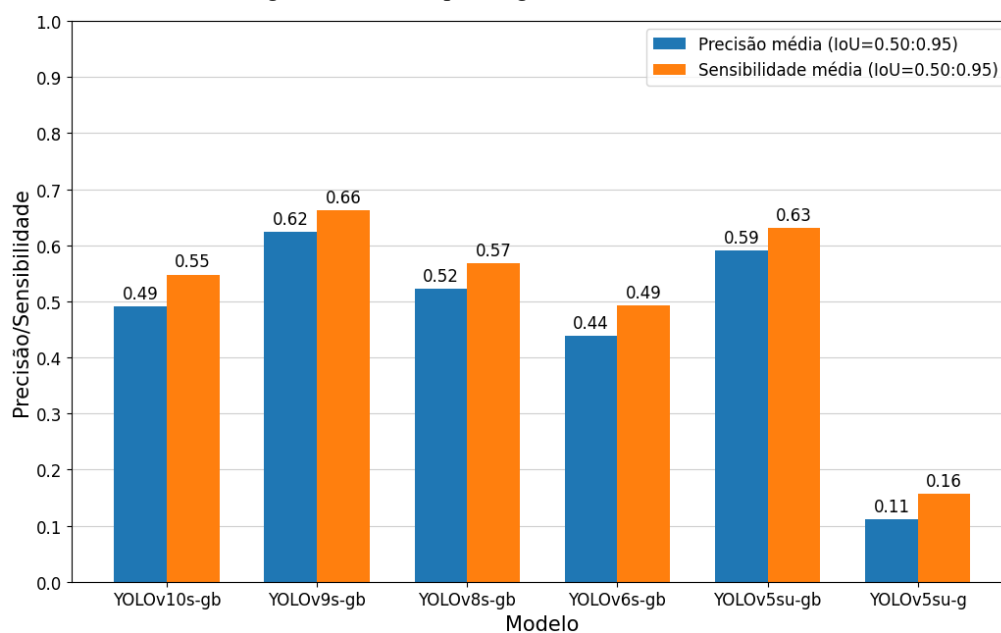
### 4.1 Detecção de placas de licenciamento

A detecção de placas de licenciamento é o primeiro passo para realizar o reconhecimento de caracteres e a identificação de veículos. As próximas subseções demonstram as métricas e análises de desempenho de variantes dos modelos YOLO nessa tarefa, com uma faixa de interseção sobre a união (*intersection over union* - IoU) superior ou igual a 0.50 (50%) e confiança mínima de 0.25 (25%) para definir uma predição como verdadeira. O conjunto de imagens de testes é o conjunto padrão utilizado para a análise de desempenho entre os diferentes modelos de detecção de objetos.

#### 4.1.1 Desempenho geral

O gráfico na Figura 21 compara a precisão média (*average precision* - AP) e a sensibilidade média (*average recall* - AR) dos modelos YOLO treinados com o conjunto de dados de treinamento (sufixo “-gb”) e conjunto de dados generalista (sufixo “-g”), aplicados ao conjunto de dados de testes, com uma faixa de IoU de 0.50 a 0.95.

Figura 21 - Desempenho geral de modelos YOLO



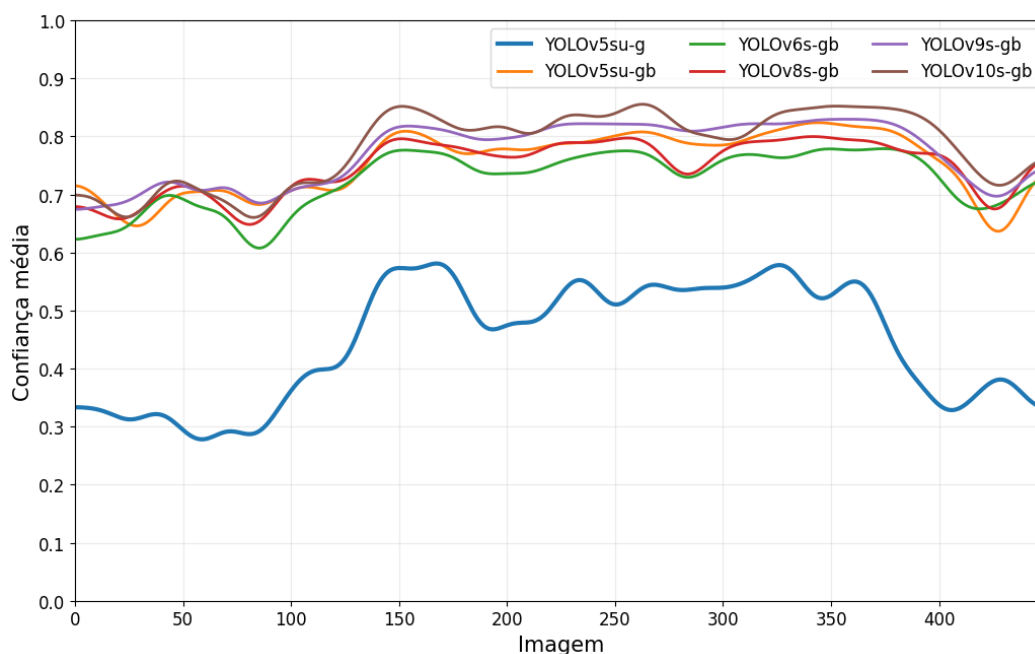
Fonte: elaborado pelo autor (2024).

Esses resultados demonstram que o modelo YOLOv9s-gb apresenta o melhor desempenho, com a maior AP (0.62) e AR (0.66). Por outro lado, o modelo YOLOv5su-g apresenta um desempenho consideravelmente inferior, com AP de 0.11 e AR de 0.16. Esse desempenho variável entre os modelos reforça a importância de utilizar conjuntos de dados específicos e variados para o treinamento de modelos de detecção de objetos.

#### 4.1.2 Confiância média

O gráfico da Figura 22 apresenta a confiância média dos modelos de detecção de placas de licenciamento veicular em cada imagem do conjunto de imagens de testes. Para suavizar as variações abruptas na confiância de detecção entre imagens, especialmente onde o modelo não foi capaz de detectar nenhuma placa, foi aplicada uma média gaussiana com  $\sigma = 10$ . Essa técnica consiste em um filtro que calcula uma média ponderada das detecções, atribui pesos maiores às confiâncias médias de imagens próximas na sequência de detecção e decresce esses pesos conforme a distância aumenta, conforme a distribuição gaussiana. O parâmetro  $\sigma$  controla a largura da curva gaussiana, isso determina o grau de suavização ao longo da sequência de imagens.

Figura 22 - Confiância média de detecção de placas de licenciamento



Fonte: elaborado pelo autor (2024).

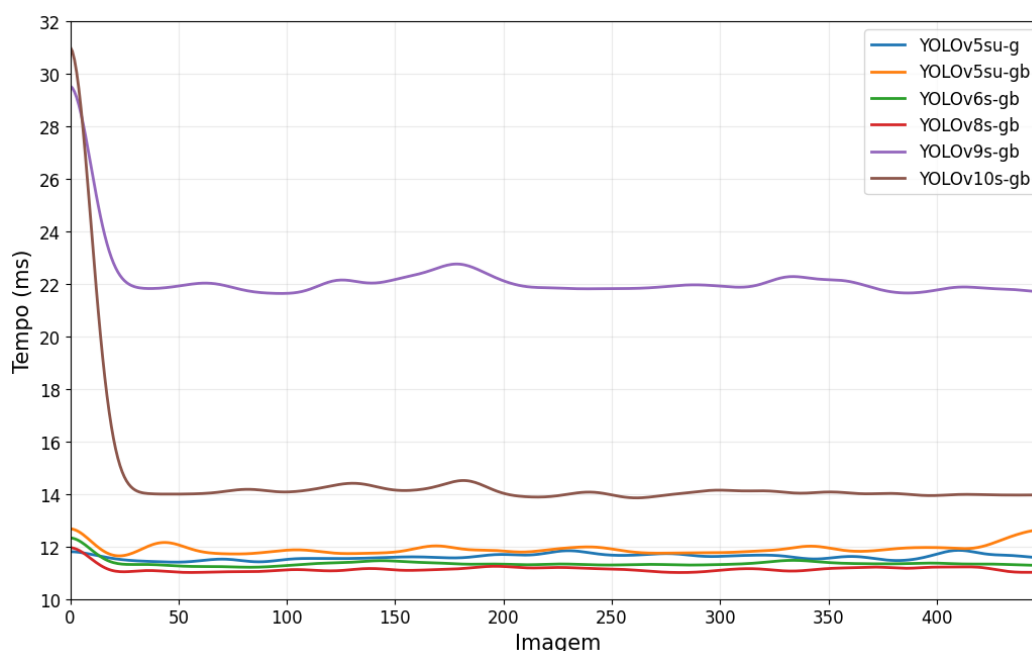
Os modelos YOLO treinados com o conjunto de dados de treinamento (sufixo “-gb”) exibem uma confiância média de inferência consistente, com variação de 0.6 (60%) a 0.9

(90%). Em contraste, o modelo YOLOv5su-g treinado exclusivamente com o conjunto de dados generalista, apresenta uma confiança de inferência significativamente menor em relação aos demais modelos. O baixo desempenho desse modelo observado no gráfico da Figura 21, é amplificado pela baixa confiança média observada, o que indica que a limitação do modelo se deve tanto à falta de diversidade nos dados de treinamento quanto à consequente queda na confiabilidade das inferências.

#### 4.1.3 Tempo total de predição

O gráfico da Figura 23 apresenta o tempo total de processamento de cada modelo na execução da tarefa de detecção de placas de licenciamento no conjunto de imagens de testes. Esse tempo total é composto pelo tempo de pré-processamento, inferência e pós-processamento de imagem. Para suavizar a variação e facilitar a análise geral, dado que alguns modelos apresentam resultados semelhantes, foi aplicada uma média gaussiana com  $\sigma = 10$ .

Figura 23 - Tempo total de predição



Fonte: elaborado pelo autor (2024).

Esse gráfico ilustra a robustez dos modelos YOLO na detecção de objetos, esses modelos apresentam um desempenho geral semelhante com uma variação de tempo próxima a 12 milissegundos. O modelo YOLOv9s-gb, no entanto, apresenta o maior tempo total, com uma variação de aproximadamente 22 milissegundos. Esses resultados são fundamentais para

assegurar que a detecção de placas de licenciamento veicular ocorra em tempo real, um requisito essencial para aplicações práticas e eficientes voltadas à fiscalização e ao monitoramento do trânsito.

A capacidade dos modelos YOLO de manterem uma variação de tempo reduzida entre suas inferências indica uma boa otimização para operações rápidas e consistentes. Esse aspecto é crucial em aplicações práticas, pois a detecção de placas de identificação veicular representa apenas uma etapa dentro de um processo mais amplo. Em sistemas desse tipo, é comum a necessidade de reconhecer os caracteres das placas, consultar sistemas e bancos de dados por meio de diversos tipos de comunicação entre computadores.

#### 4.1.4 Exemplos de detecções

A Figura 24 ilustra a detecção de placas de licenciamento veicular com o modelo YOLOv8s-gb no local 1, esse é um dos locais onde a gravação do conjunto de imagens para treinamento e testes foi realizada. A imagem destaca a capacidade desse modelo em identificar e localizar com precisão as placas desse local.

Figura 24 - Exemplo de detecção de placa no local 1



Fonte: elaborado pelo autor (2024).

Nesse exemplo, o modelo é capaz de detectar uma placa de licenciamento veicular com precisão de 0.80 (80%) e IoU de 0.82 (82%). Nesse local, a gravação foi realizada com a câmera posicionada na lateral direita da pista, a cerca de 1 metro de altura com o auxílio de um tripé de modo a sempre capturar a região traseira dos veículos. O ambiente geralmente é sombreado, o que adiciona um desafio à detecção de placas de licenciamento veicular.

A Figura 25 ilustra a detecção de placas de licenciamento veicular no local 2, em condições ambientais diferentes em relação ao local 1. A gravação foi realizada com a câmera montada em um tripé ajustado em sua altura mínima (50cm), posicionada sob uma passarela a aproximadamente 4,5 metros de altura e centralizada em relação à pista de rodagem.

Figura 25 – Exemplo de detecção de placas no local 2



Fonte: elaborado pelo autor (2024).

Nessa figura, é possível perceber que distância entre a câmera e os veículos é significativamente maior em comparação ao local 1, devido à altura da passarela. Apesar dessa maior distância, as placas de licenciamento de dois veículos que circulam na pista são detectadas com uma confiança de 0.75 (75%) para ambos. Essa taxa de confiança de detecção destaca a eficácia do modelo de detecção. No entanto, o veículo mais distante em relação à câmera apresenta um valor de IoU inferior ao do veículo mais próximo, com valores de 0.69 (69%) e 0.79 (79%), respectivamente. Essa diferença sugere que, à medida que a distância aumenta, a qualidade da detecção pode ser comprometida, isso enfatiza a importância de ajustar a posição da câmera ou melhorar a resolução de imagem para garantir uma detecção mais precisa em situações similares.

## 4.2 Reconhecimento de placas de licenciamento

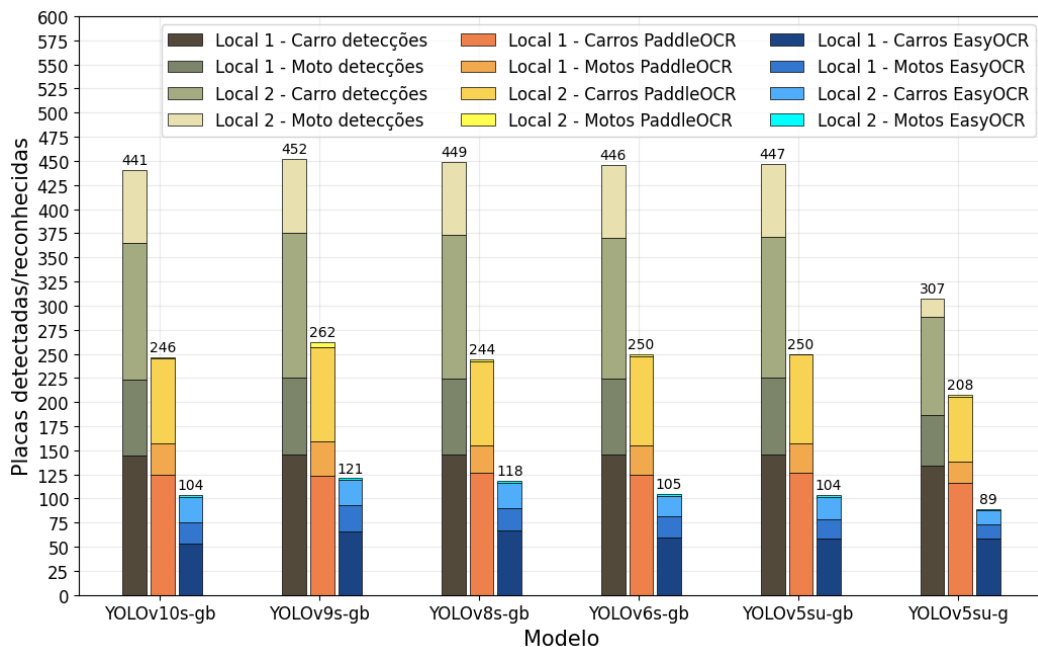
O reconhecimento de placas de licenciamento é baseado em OCR. Nas próximas subseções, serão apresentados os resultados e a análise do desempenho geral dos modelos EasyOCR e PaddleOCR, aplicados às placas detectadas pelos modelos YOLO no conjunto de imagens de testes.

#### 4.2.1 Desempenho geral

O conjunto de imagens de testes contém um total de 460 placas de identificação veicular, dividido em 225 imagens para cada local de gravação. As imagens do local 1 contém 146 placas de carros e 79 de motos, enquanto as imagens do local 2 contém 158 placas de carros e 77 de motos.

O gráfico da Figura 26 apresenta a quantidade de placas de licenciamento veicular reconhecidas corretamente pelos modelos de OCR. Os dados desse gráfico estão categorizados em local de gravação, tipo de placa, modelo de OCR e o modelo YOLO utilizado para a detecção das placas.

Figura 26 - Total de reconhecimento de placas



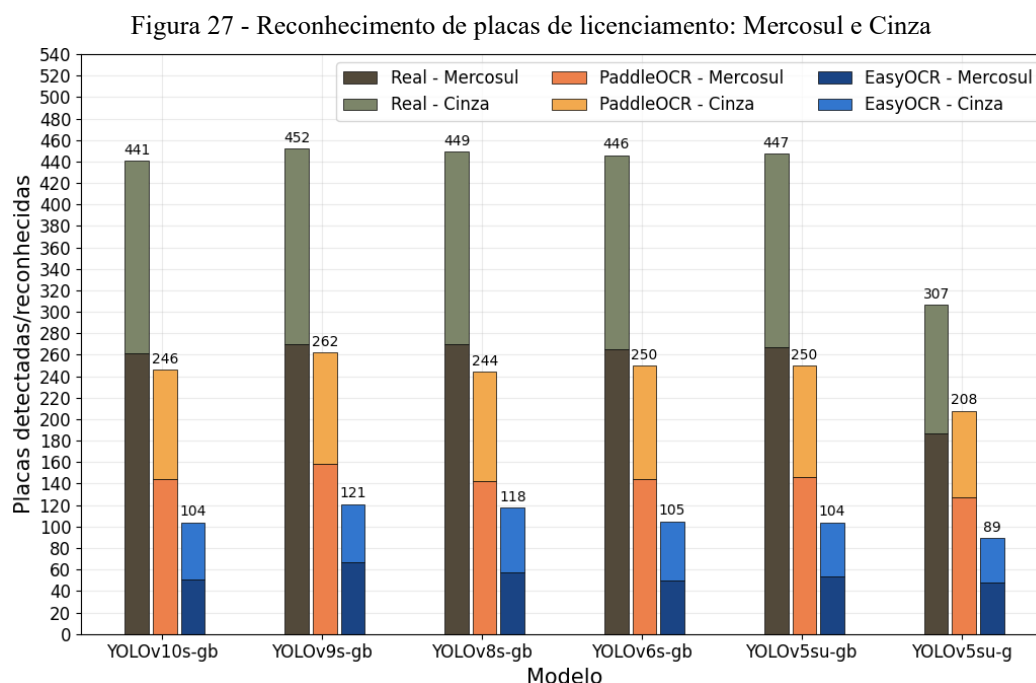
Fonte: elaborado pelo autor (2024).

Nesse gráfico, é possível observar que o modelo de reconhecimento óptico de caracteres PaddleOCR supera, consideravelmente, o modelo EasyOCR em todas as métricas e para todos os modelos de detecção YOLO, isso pode estar relacionado a uma preferência do EasyOCR para textos contínuos, como em livros ou fachadas de estabelecimentos comerciais. Ambos os modelos de OCR apresentam um desempenho superior no local 1 em relação ao local 2. Essa diferença de desempenho é explicada pela maior distância da câmera em relação aos veículos no local 2, o que pode ter impactado negativamente na qualidade de imagem das detecções e, conseqüentemente, o reconhecimento óptico dos caracteres das placas. Além disso, é possível perceber que nos modelos YOLO treinados com o conjunto de dados de

treinamento (sufixo “-gb”), quase todas as placas do conjunto de imagens de testes foram detectadas corretamente. Em contraste, no modelo treinado com conjunto de dados generalista (YOLOv5su-g), a quantidade de placas detectadas é significativamente reduzida.

No entanto, mesmo com essa redução, as placas que foram detectadas pelo modelo YOLOv5su-g em sua maioria eram as mais evidentes e com boa qualidade de imagem, conseqüentemente, mais favoráveis a serem reconhecidas pelos modelos de OCR. Esse fato é corroborado pela similaridade na quantidade de placas reconhecidas, tanto de motos quanto de carros, no local 1 em comparação com os outros modelos YOLO. Isso sugere que, apesar das limitações do modelo YOLOv5su-g, causadas pelo treinamento com conjunto de dados generalista, ainda foi possível detectar com precisão as placas mais evidentes, especialmente no local 1, onde as condições de gravação podem ter sido mais favoráveis.

O gráfico da Figura 27 ilustra a quantidade de placas reconhecidas corretamente pelos modelos PaddleOCR e EasyOCR, categorizadas por Mercosul e Cinza. Esses tipos de placa de licenciamento são distintos em cores, padrão de caracteres, fontes e tamanho. Além disso, como na Figura 26, os resultados são divididos em diferentes versões do YOLO, onde a barra real representa a quantidade e a categoria de placas detectadas.



Fonte: elaborado pelo autor (2024).

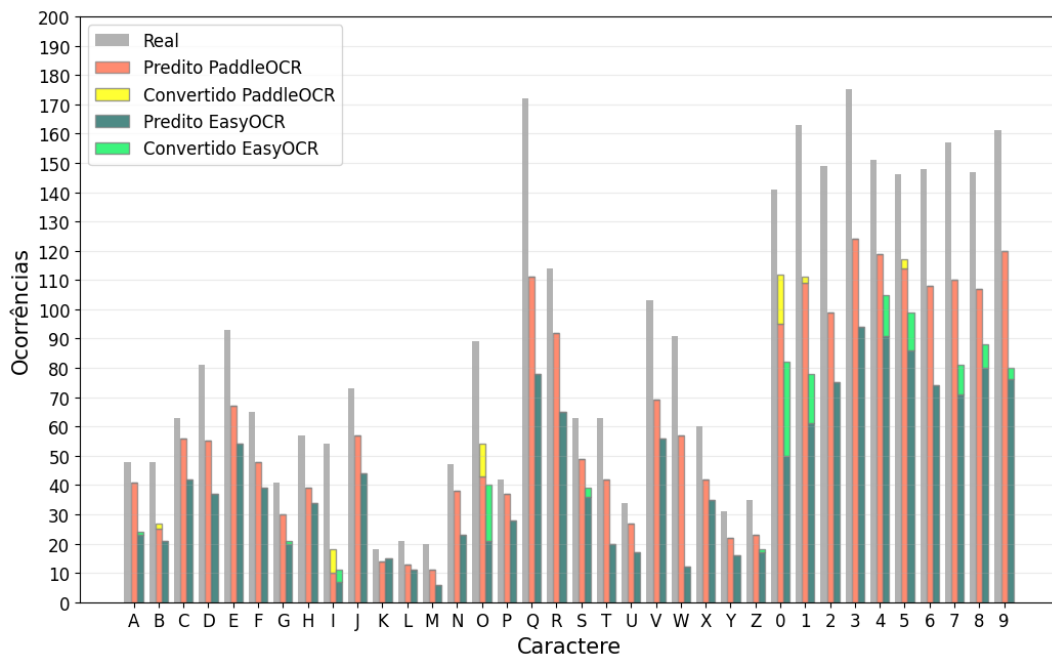
Nesse gráfico, é possível observar que mesmo com diferenças físicas significativas entre as placas de licenciamento Mercosul e Cinza, não há uma tendência clara de um dos modelos de OCR ser mais eficiente em reconhecer placas de um tipo específico. Ao analisar a

quantidade de placas reconhecidas, é possível notar que tanto o PaddleOCR quanto o EasyOCR mantêm um desempenho equilibrado entre os dois tipos de placa, com variação causada pelas limitações desses modelos. Para cada modelo do YOLO treinado com o conjunto de dados misto (\*-gb), o número de placas detectadas dos tipos Cinza e Mercosul é bastante semelhante, com variações mínimas. Embora o YOLO tenha detectado uma quantidade maior de placas do modelo Mercosul em comparação com o modelo Cinza, a quantidade de placas reconhecidas do tipo Mercosul também foi superior, isso reflete diretamente essa maior quantidade de detecções. Isso sugere que o desempenho do reconhecimento está fortemente correlacionado a quantidade de detecções iniciais, sem indicar uma preferência intrínseca dos modelos de OCR por um tipo de placa específico.

#### 4.2.2 Desempenho por caractere

O gráfico da Figura 28 apresenta o desempenho dos modelos de OCR aplicados em placas detectadas pelo modelo YOLOv8s-gb, ao considerar os acertos de caracteres alfanuméricos. Esse modelo escolhido não é o modelo que apresenta o melhor desempenho na Figura 26, mas é similar ao YOLOv9s-gb e é o modelo mais rápido observado na Figura 23.

Figura 28 - Reconhecimento de caracteres – YOLOv8s-gb



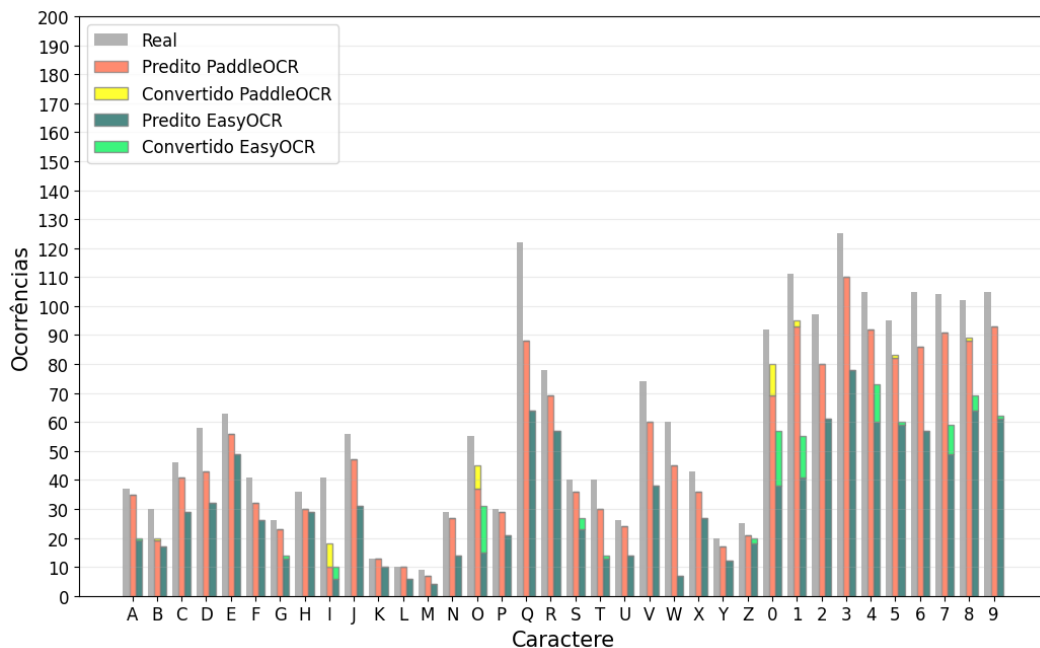
Fonte: elaborado pelo autor (2024).

Esse gráfico utiliza o empilhamento de barras para exibir a contribuição das predições corretas e das conversões de caracteres, na formação do total de acertos por modelo. Essa visualização permite uma análise detalhada da eficácia dos modelos de OCR em reconhecer

corretamente cada caractere individual das placas veiculares, isso destaca as diferenças de desempenho entre os modelos analisados. Além disso, demonstra a superioridade do modelo PaddleOCR na tarefa de reconhecimento óptico de caracteres, uma vez que, apresenta mais acertos de caracteres de forma geral e isso impacta diretamente no reconhecimento total de placas de licenciamento veiculares. Porém, ambos os modelos analisados apresentam maior dificuldade em diferenciar os caracteres “O” e “0” devido a suas similaridades visuais, condições ambientais de imagem e ângulos de captura.

O gráfico da Figura 29 é similar ao gráfico da Figura 28, porém este considera o modelo YOLOv5su-g na detecção de placas de licenciamento veiculares. Esse modelo apresentou o menor desempenho geral nessa tarefa, porém, é possível observar que ainda assim os resultados são similares aos da Figura 28, em termos de caracteres corretos. Isso é confirmado na Figura 26, onde esse modelo apresenta apenas uma leve variação para baixo no reconhecimento total de placas de licenciamento, em relação aos outros modelos. Além disso, é possível perceber que a quantidade de caracteres reais é inferior, justamente, devido a não detecção das placas mais distantes e com qualidade de imagem inferior.

Figura 29 - Reconhecimento de caracteres – YOLOv5su-g



Fonte: elaborado pelo autor (2024).

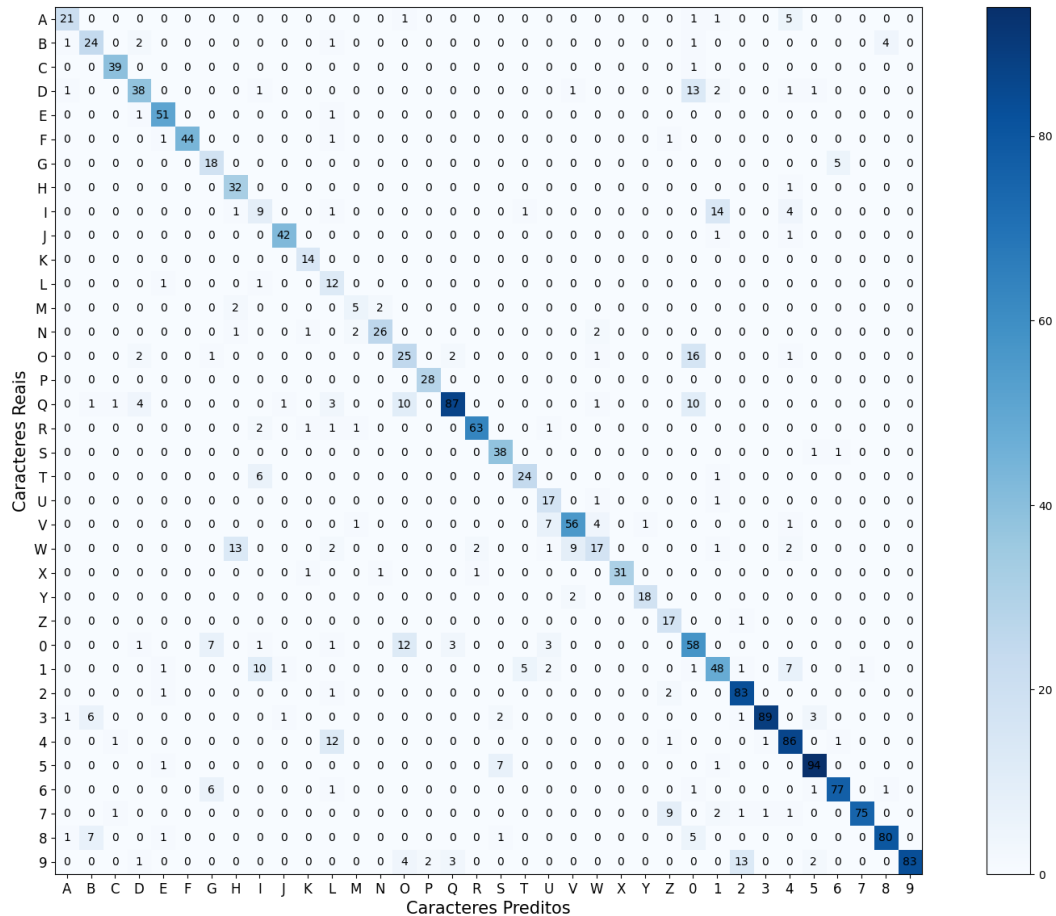
Nesse gráfico, é possível observar que as barras de conversão de caracteres apresentam uma acentuação menos pronunciada, embora ainda apresentem semelhança com as barras de conversão exibidas no gráfico da Figura 29. Esse fato evidencia que o mapa de

conversão possui limitações intrínsecas e que, mesmo com o aumento no número de placas detectadas, a qualidade da imagem continua a ser um fator crucial. Se as imagens não apresentarem uma qualidade satisfatória, os modelos de OCR não conseguem fornecer uma margem de confiança suficientemente elevada, o que compromete a acurácia dos resultados e consequentemente a eficácia do mapa de conversão de caracteres. Além disso, essa observação reforça a importância de se utilizar técnicas avançadas de processamento de imagem para garantir que as entradas submetidas ao OCR estejam dentro de parâmetros ideais de qualidade, de modo a obter uma melhor performance dos modelos e maior precisão na conversão de caracteres que são comumente confundidos pelos modelos de OCR.

### 4.2.3 Matriz de confusão de caracteres

A Figura 30 ilustra uma matriz de confusão, com a performance do modelo EasyOCR na tarefa de reconhecimento de caracteres em placas detectadas pelo modelo YOLOv8s-gb.

Figura 30 - Matriz de confusão de caracteres EasyOCR

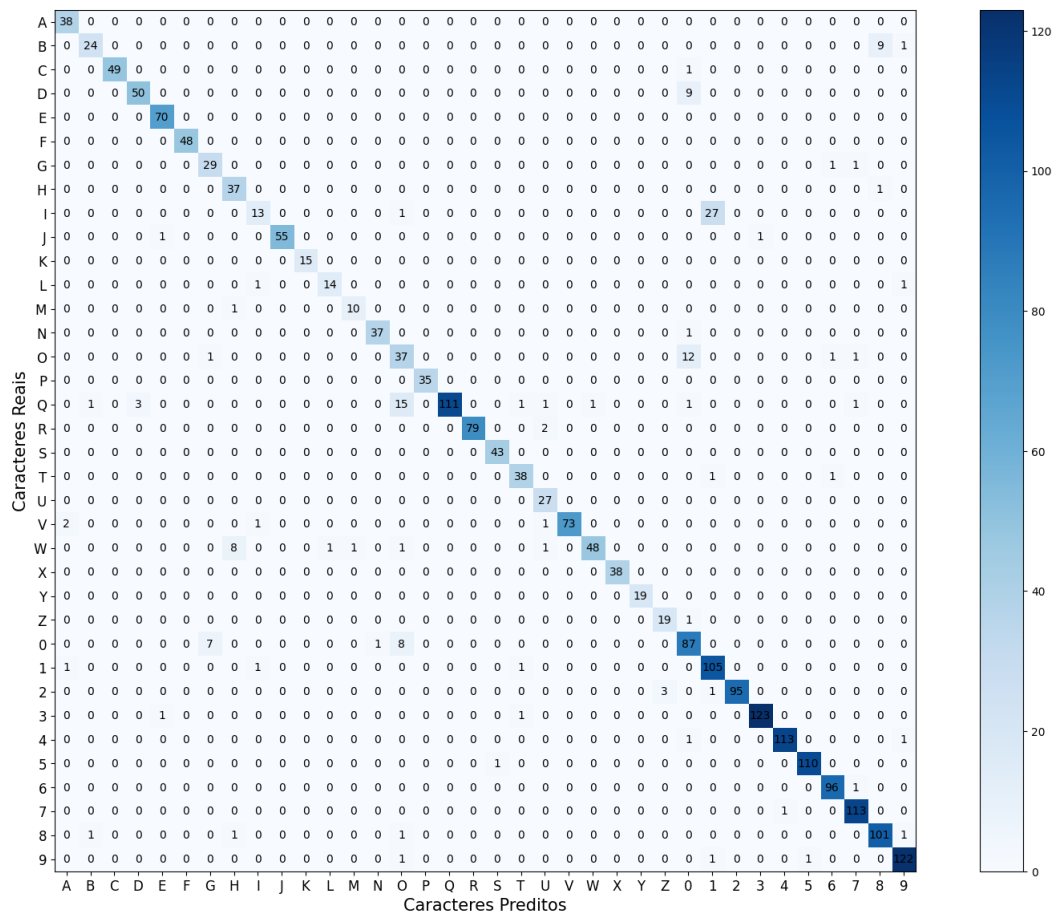


Fonte: elaborado pelo autor (2024).

Os caracteres reais estão listados no eixo vertical, enquanto os caracteres previstos pelo modelo EasyOCR estão no eixo horizontal. Cada célula da matriz indica o número de vezes que um determinado caractere real foi previsto como um caractere específico pelo modelo de OCR. A diagonal principal (descendente) da matriz representa as previsões corretas, onde os caracteres reais coincidem com os caracteres previstos. Quanto mais escura a célula na diagonal, maior a precisão na identificação daquele caractere específico.

Ao comparar a matriz de confusão do EasyOCR com a do PaddleOCR na Figura 31, é possível observar diferenças na performance de cada modelo na tarefa de reconhecimento de caracteres.

Figura 31 - Matriz de confusão de caracteres PaddleOCR



Fonte: elaborado pelo autor (2024).

Enquanto o EasyOCR apresentou uma quantidade considerável de confusões, o PaddleOCR demonstra uma maior precisão, refletida em uma densidade maior nas células da diagonal principal. Isso sugere que o PaddleOCR é mais eficaz na identificação correta dos caracteres.

Por outro lado, embora o PaddleOCR apresente uma performance superior, ainda há áreas onde melhorias são necessárias. Caracteres, como '0' e '1', continuam a apresentar confusões, isso indica que mesmo modelos mais avançados ainda enfrentam desafios em situações complexas de reconhecimento. Em resumo, o PaddleOCR oferece uma maior precisão e confiabilidade em comparação ao EasyOCR, mas o desenvolvimento contínuo e o ajuste fino desses modelos são essenciais para mitigar os erros restantes e garantir uma taxa de acerto ainda mais alta em aplicações do mundo real.

## 5 CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi realizada a construção de um conjunto de dados composto por placas de identificação veicular brasileiras, integrado a um conjunto de dados generalista, o qual foi utilizado no treinamento de variantes do modelo YOLO. Essa abordagem permitiu uma análise detalhada do desempenho desses modelos em um conjunto de dados de teste e viabilizou a aplicação de técnicas de reconhecimento óptico de caracteres (OCR) por meio dos modelos EasyOCR e PaddleOCR.

Os experimentos demonstraram que o modelo YOLOv8s-gb, treinado com o conjunto de dados de treinamento, apresentou um desempenho significativamente superior em termos de confiança média e tempo de processamento quando comparado ao modelo YOLOv5su-g, que foi treinado exclusivamente com um conjunto de dados generalista. Apesar de suas limitações, o YOLOv5su-g mostrou-se eficaz na detecção de placas mais evidentes em condições favoráveis de captura de imagem, como aquelas observadas no local 1. Além disso, de um total de 460 placas do conjunto de dados de teste, o modelo YOLOv9s-gb foi capaz de detectar corretamente 452 placas, enquanto o YOLOv8s-gb e YOLOv5su-g detectaram, respectivamente, 449 e 307.

Nos testes realizados com OCR, o PaddleOCR obteve um desempenho consistentemente superior ao EasyOCR em todas as métricas avaliadas, tanto em precisão quanto na taxa de acertos dos caracteres reconhecidos. As matrizes de confusão evidenciaram a menor ocorrência de erros de confusão e a implementação de um mapa de conversão de caracteres mostrou-se eficaz para mitigar erros recorrentes, isso contribuiu para o aumento da taxa de acertos no reconhecimento das placas. A combinação do PaddleOCR com o YOLOv8s-gb e YOLOv5su-g aplicados a um total de 460 placas, resultou no reconhecimento de 244 e 208, respectivamente, enquanto o EasyOCR reconheceu 118 e 89 placas nos mesmos cenários.

De maneira geral, os resultados obtidos comprovam a viabilidade do reconhecimento de placas de veículos brasileiros com uso dos métodos e modelos empregados. Os modelos demonstraram eficiência no processamento de imagens e na extração precisa dos caracteres das placas, isso evidencia o potencial para aplicações práticas. No entanto, para implementação em escala real, é indispensável o acesso a um banco de dados com informações públicas ou privadas de veículos, o que permitiria a realização de verificações automatizadas.

Embora o estudo tenha demonstrado a eficácia do PaddleOCR em relação ao EasyOCR, ressaltado a relevância da escolha do conjunto de dados de treinamento, bem como a escolha do modelo YOLO, ainda existem desafios e oportunidades de aprimoramento a serem explorados em trabalhos futuros. Entre as possibilidades, está a integração com sistemas de imagens de trânsito em tempo real, isso pode ser desafiador ao incluir variáveis como o tempo de transferência de dados da câmera que capta as imagens e o sistema que realiza o processamento. O acoplamento a bases de dados para consultas de veículos, esse acesso pode ser possível através de parcerias com instituições públicas ou privadas que possuem autorização legal para posse e compartilhamento de informações de veículos nacionais. O refinamento dos modelos YOLO, através da transferência de aprendizagem e uso de novas imagens.

## REFERÊNCIAS

ALIFERIS, Constantin, *et al.* "Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI". **Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls**. Cham: Springer International Publishing. 2024.

BANSAL, Swati, *et al.* "Enhancing ALPR: a two stage YOLO model with data augmentation for improved accuracy and robustness". **Multimedia Tools and Applications**. 2024.

CONTRAN. **ESPECIFICAÇÕES DO SISTEMA DE PLACAS DE IDENTIFICAÇÃO VEICULAR (PIV)**. Disponível em: <https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-contran/resolucoes/resolucao9692022anexos.pdf>. Acesso em: 06 dez. 2024.

CVAT. **Open Data Annotation Platform**. Disponível em: <https://www.cvat.ai/>. Acesso em: 26 .out 2024.

DIAS, Marcelo, *et al.* "DRAT: A semi-supervised tool for automatic annotation of lesions caused by diabetic retinopathy". **Proceedings of the 20th Brazilian Symposium on Information Systems**. 2024.

DU, Yuning, *et al.* "Pp-ocr: A practical ultra lightweight ocr system". 2020.

EVERINGHAM, Mark, *et al.* "The pascal visual object classes (voc) challenge". **International journal of computer vision 88 (2010)**. 2010.

FILHO, Ogê M. e NETO, Hugo V. "Processamento digital de imagens". 1999.

GONZALEZ, Rafael C. e WOODS, Richard E. "Digital image processing". **Pearson education, global edition**. 2018.

GHOSH, Anirudha, *et al.* "Fundamental concepts of convolutional neural network". **Recent trends and advances in artificial intelligence and Internet of Things**. 2020.

Google Maps. **Locais de gravação**. Disponível em: [https://www.google.com/maps/dir/-3.824555,-49.675044/-3.779717,-49.676377/@-3.8060481,-49.6904577,13.75z?entry=tту&g\\_ep=EgoyMDI0MTAyMy4wIKXMDS0ASAFQAw%3D%3D](https://www.google.com/maps/dir/-3.824555,-49.675044/-3.779717,-49.676377/@-3.8060481,-49.6904577,13.75z?entry=tту&g_ep=EgoyMDI0MTAyMy4wIKXMDS0ASAFQAw%3D%3D). Acesso em: 21 out. 2024.

GRAY, David, *et al.* "Further thoughts on precision". **15th Annual Conference on Evaluation & Assessment in Software Engineering (EASE 2011)**. 2011.

HU, Liqi, *et al.* "Research and Implementation of a License Plate Detection Algorithm for Complex Environments". **2023 8th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)**. 2023.

ISMAIL, Amir, *et al.* PGTLP: A Dataset for Tunisian License Plate Detection and Recognition. In: **2021 18th International Multi-Conference on Systems, Signals & Devices (SSD)**. IEEE, 2021.

JAIDEDAI. EasyOCR. Disponível em: <https://github.com/JaidedAI/EasyOCR>. Acesso em: 21 out. 2024.

KAGGLE. **Technical Specifications**. Disponível em: <https://www.kaggle.com/docs/notebooks#technical-specifications>. Acesso em: 26 out. 2024.

KESUMAH, Farhan Tirta, *et al.* "YOLO-Based Vehicle Plate Number Recognition with Unconstrained Conditions". **2024 International Conference on Data Science and Its Applications (ICoDSA)**. IEEE, 2024.

LECUN, Yann, *et al.* "Deep learning". **Nature**. 2015.

LIN, Tsung-Yi, *et al.* "Microsoft COCO: Common Objects in Context". **European Conference on Computer Vision (ECCV)**. 2014.

MANE, Deepak, *et al.* "A Metaphor Analysis on Vehicle License Plate Detection using Yolo-NAS and Yolov8." **Journal of Electrical Systems** **20.1s**. 2024.

MORESI, Eduardo. "Metodologia da pesquisa". **Brasília: Universidade Católica de Brasília**. 2003.

RAHMAN, Rashik, *et al.* "A real-time end-to-end Bangladeshi license plate detection and recognition system for all situations including challenging environmental scenarios". **2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)**. 2021.

REDMON, Joseph, *et al.* "You only look once: Unified, real-time object detection". *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

ROBOFLOW. **License Plate Recognition > resized640\_aug3x-ACCURATE**. Disponível em: <https://universe.roboflow.com/roboflow-universe-projects/license-plate-recognition-rxg4e/dataset/4>. Acesso em: 30 maio 2024.

RUSSELL, Bryan, *et al.* "Object recognition by scene alignment". **Advances in Neural Information Processing Systems**, **20**. 2007.

SENATRAN, Secretaria Nacional de Trânsito. **Frota de veículos por UF e Tipo de Veículo - Agosto de 2024**. Disponível em: <https://www.gov.br/transportes/pt-br/assuntos/transito/conteudo-Senatran/FrotaporUFETipodeVeculoAgosto2024.xlsx>. Acesso em: 29 set. 2024.

SERPRO, Serviço Federal de Processamento de Dados. **Inteligência Artificial é utilizada na gestão do trânsito brasileiro**. Disponível em: <https://www.serpro.gov.br/menu/noticias/noticias-2018/inteligencia-artificial-e-utilizada-na-gestao-do-transito-brasileiro>. Acesso em: 29 set. 2024.

SRINIVASAN, R., *et al.* "Real-Time Vehicle Classification and License Plate Recognition via Deformable Convolution based Yolo v8 Network". **IEEE Sensors Journal**. 2024.

ULTRALYTICS. **Models Supported by Ultralytics**. Disponível em: <https://docs.ultralytics.com/models/>. Acesso em: 06 dez. 2024.

VEDHAVIYASSH, D. R., *et al.* "Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm". **2022 6th International Conference on Electronics, Communication and Aerospace Technology**. 2022.

XIAOMI. **United Kingdom**. Disponível em: <https://www.mi.com/uk/redmi-note-9-pro/specs/>. Acesso em: 22 nov. 2024.

## APÊNDICE A – REPOSITÓRIOS DE CÓDIGO-FONTE UTILIZADOS

Treinamento de modelos YOLO: <https://github.com/brenodahora/YOLO-training>

Reconhecimento de placas: <https://github.com/brenodahora/YOLO-recognition>