

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Análise e Avaliação do Reconhecimento de Placas Veiculares por meio de Redes Neurais

Pedro Henrique Baptista de Queiroz Assumpção

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Análise e Avaliação do Reconhecimento de Placas Veiculares por meio de Redes Neurais

Pedro Henrique Baptista de Queiroz Assumpção

USP - São Carlos
2024

Nome do Aluno Pedro Henrique Baptista de Queiroz Assumpção

Análise e Avaliação do Reconhecimento de Placas Veiculares por meio de Redes Neurais

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Fernando Pereira dos Santos

USP - São Carlos

2024

RESUMO

ASSUMPÇÃO, P. H. B. Q. **Análise e Avaliação do Reconhecimento de Placas Veiculares por meio de Redes Neurais.** 2024. 70 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Estudo de uma análise aprofundada do cenário de reconhecimento de placas veiculares utilizando o modelo de deep learning YOLO em um *dataset* específico, considerando a variabilidade da qualidade de entrada das imagens capturadas para o modelo, avaliação de resultados e aplicação de ajustes aos modelos inseridos em suas propriedades de forma a melhorar seus resultados. A abordagem adotada visou otimizar o desempenho do modelo ao explorar técnicas de pré-processamento de imagens, modificando a angulação e outras variáveis na qualidade de captação, visando simular diferentes condições de entrada das imagens. O objetivo foi avaliar como essas alterações impactam a acurácia do modelo, o reconhecimento distinto e com avaliação da precisão para cada um dos caracteres detectados em sua saída, e outros datapoints relevantes. Ao explorar estas dimensões, este trabalho contribui para a compreensão mais aprofundada das nuances do modelo YOLO e seu comportamento sob diversas condições, fornecendo diretrizes valiosas para otimização de modelos de IA em tarefas semelhantes.

Palavras-chave: reconhecimento de placas veiculares; YOLO; redes neurais convolucionais; pré-processamento de imagens; *Dataset*

ABSTRACT

ASSUMPÇÃO, P. H. B. Q. **Title in English:** subtitle. 2024. 70 f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

This study aimed to conduct an in-depth analysis of the vehicle license plate recognition scenario using the YOLO deep learning model on a specific dataset, considering the variability in the quality of the captured images for the model, evaluating the results, and applying adjustments to the models' properties to improve their performance. The adopted approach aimed to optimize the model's performance by exploring image pre-processing techniques, modifying the angle and other variables in the quality of capture, in order to simulate different input conditions of the images. The objective was to evaluate how these changes impact the model's accuracy, distinct recognition, and precision evaluation for each of the characters detected in its output, and other relevant data points. By exploring these dimensions, this work contributes to a deeper understanding of the nuances of the YOLO model and its behavior under various conditions, providing valuable guidelines for the optimization of AI models in similar tasks.

Keywords: vehicle license plate recognition; YOLO; convolutional neural networks; image pre-processing; Dataset

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de um modelo não-linear de um neurônio.....	22
Figura 2 – Representação de um modelo MLP.....	23
Figura 3 – Arquitetura representativa de uma Rede Neural Convolucional	24
Figura 4 – Arquitetura representativa da YOLO para uma entrada de imagem	26
Figura 5 – Exemplos de capturas com dificuldades sobre as placas veiculares	28
Figura 6 – Resultados de performance da YOLOv9 em comparação com versões anteriores .	30
Figura 7 – Exemplo de implementação com EasyOCR	31
Figura 8 – Exemplo de imagem do Dataset Europeu	32
Figura 9 – Exemplo imagem para o Dataset Brasileiro	33
Figura 10 – Configuração de Hiperparâmetros	38
Figura 11 – Resultados Treinamento Dataset 1 – Modelo Europeu	39
Figura 12 – Matriz de Confusão Dataset 1 – Modelo Europeu	40
Figura 13 – F1-Score Dataset 1 – Modelo Europeu	41
Figura 14 – Resultados de detecção Placa Europeia – Modelo Europeu	42
Figura 15 – Variação de detecção Placa Europeia – Modelo Europeu	44
Figura 16 – Detecção de placa de maior contraste – Modelo Europeu	45
Figura 17 – Confusão entre números e letras – Modelo Europeu	46
Figura 18 – Representação geral de detecção placa brasileira/Mercosul – Modelo Europeu ..	48
Figura 19 – Caracteres identificados placa brasileira/Mercosul – Modelo Europeu	54
Figura 20 – Resultados Treinamento Dataset 2 – Modelo Brasileiro/Mercosul	55
Figura 21 – Matriz de Confusão Dataset 2 – Modelo Brasileiro/Mercosul	56
Figura 22 – F1-Score Dataset 2 – Modelo Brasileiro/Mercosul	57
Figura 23 – Resultados de detecção Placa Europeia – Modelo Brasileiro/Mercosul	58
Figura 24 – Identificação ineficiente de placas europeias – Modelo Brasileiro/Mercosul ..	61
Figura 25 – Detecção de placa de maior contraste – Modelo Brasileiro/Mercosul	62
Figura 26 – Detecção placa brasileira/Mercosul – Modelo Brasileiro/Mercosul	63

LISTA DE TABELAS

Tabela 1 – Resultados de detecção Placa Europeia – Modelo Europeu	43
Tabela 2 – Placa identificada ao longo dos frames – Modelo Europeu	46
Tabela 3 – Placas brasileiras detectadas ao longo dos frames– Modelo Europeu.....	49
Tabela 4 – Resultados de detecção Placa Europeia – Modelo Brasileiro/Mercosul	58
Tabela 5 – Resultados de detecção Placa Brasileira/Mercosul –Modelo Brasileiro/Mercosul	63

LISTA DE ABREVIATURAS E SIGLAS

AP	–	Average Precision
CNNs	–	Convolutional Neural Networks
FPS	–	Frames Por Segundo
GPU	–	Graphical Processing Unit
IA	–	Inteligência Artificial
mAP	–	Mean Average Precision
MERCOSUL	–	Mercado Comum do Sul
MLP	–	Multilayer Perceptron
NMS	–	Non-Maximum Suppression
OCR	–	Optical Character Recognition
ReLU	–	Rectified Linear Unit
RNNs	–	Revolutionary Neural Networks
YOLO	–	You Only Look Once

SUMÁRIO

1 INTRODUÇÃO.....	19
1.1 Objetivos.....	19
2 FUNDAMENTAÇÃO TEÓRICA.....	21
2.1 Redes Neurais.....	21
2.1.1 Redes Neurais Convolucionais.....	24
2.2 Modelo YOLO.....	25
2.2 Detecção e Identificação de Placas.....	27
3 METODOLOGIA	29
3.1 Ferramentas Computacionais	29
3.1.1 Pandas	29
3.1.2 Numpy	29
3.1.3 YOLOv9	29
3.1.4 EasyOCR para Detecção em Tempo Real.....	30
3.2 Distribuição e avaliação de Datasets.....	31
3.3 Metodologia de Desenvolvimento.....	33
3.3.1 Estudo dos Hiperparâmetros da YOLO.....	33
3.3.2 Simulação de Distúrbios e Condições Não-Ideais.....	35
4 ANÁLISE E RESULTADOS	37
4.1 Aplicação com treinamento via Dataset 1 – Modelo Europeu.....	37
4.1.1 Resultados Vídeo 1 – Placa Europeia	41
4.1.1.1 Avaliação de Resultados.....	42
4.1.2 Resultados Vídeo 2 – Placa Brasileira/Mercosul.....	49
4.1.2.1 Avaliação de Resultados.....	50
4.2 Aplicação com treinamento via Dataset 2 – Modelo Mercosul.....	55
4.2.1 Resultados Vídeo 1 – Placa Europeia.....	58
4.2.1.1 Avaliação de Resultados.....	59
4.2.2 Resultados Vídeo 2 – Placa Brasileira/Mercosul.....	63
4.2.2.1 Avaliação de Resultados	64
5 CONCLUSÃO.....	70
REFERÊNCIAS	71

1 INTRODUÇÃO

A crescente importância da Inteligência Artificial (IA) tem desencadeado avanços significativos em diversas áreas, incluindo visão computacional e reconhecimento de padrões. O contexto de aplicações que podem se beneficiar deste tipo de tecnologia se estende a inúmeras áreas, e um ponto em comum para todas é o qual preciso o modelo se comporta. Dentro deste âmbito, a qualidade de entrada das imagens a serem processadas tem papel fundamental na qualidade de sua classificação e extração de dados – especialmente em aplicações que sua classificação e interpretação tem impacto direto com as necessidades de negócio (LAROCA, 2018).

O emprego deste tipo de tecnologia para o reconhecimento e interpretação das placas veiculares é altamente justificado na literatura, onde dados coletados e os casos de uso exemplificados mostram a importância de sua utilização e refinamento, uma vez que são destacadas também os pontos de melhoria e melhores treinamentos que podem elevar ainda mais a eficiência destas utilizações (DU, 2013). Ainda dentro do preceito dos vários modelos disponíveis, a aplicação de cada um para cada caso deste tipo de reconhecimento também reforça o princípio da avaliação e desempenho de cada um, com seus respectivos pontos de melhoria, apontando tecnologias como o YOLO (*You Only Look Once*) e outros como princípios de ajuste para melhor desempenho (KHAN, 2023).

No estudo de caso de um modelo para verificação de placas veiculares, a extração correta de suas informações se faz crucial para o sucesso do negócio e funcionamento correto de suas informações. Porém, por se tratar de muitas vezes sua captura e identificação se dar em ambientes externos, tais ambientes não são controlados e as capturas estão sempre submetidas a condições variáveis de luz – e tal como sua incidência sobre o objeto; ângulo da câmera em relação ao objeto; condições de impurezas sobre o objeto; e etc, que podem variar drasticamente como o modelo pode interpretar cada caso (ANAGNOSTOPOULOS, 2014).

1.1 Objetivos

Neste contexto, este estudo se propôs a realizar uma análise aprofundada do cenário de reconhecimento de placas veiculares utilizando o modelo de *deep learning* YOLO em um *dataset* específico, considerando a variabilidade da qualidade de entrada das imagens

capturadas para o modelo, avaliação de resultados e aplicação de ajustes aos modelos inseridos em suas propriedades de forma a melhorar seus resultados.

A abordagem adotada visou otimizar o desempenho do modelo ao explorar técnicas de pré-processamento de imagens. Em particular, foram aplicadas modificações na angulação e outras variáveis no quesito de qualidade de captação, visando simular diferentes condições de entrada das imagens. O objetivo foi avaliar como essas alterações impactam a acurácia do modelo, o reconhecimento distinto e com avaliação da precisão para cada um dos caracteres detectados em sua saída, e outros *datapoints* relevantes.

Ao explorar estas dimensões, este trabalho contribui para a compreensão mais aprofundada das nuances do modelo YOLO e seu comportamento sob diversas condições, para que os *insights* obtidos possam não apenas enriquecer o entendimento sobre o reconhecimento de placas de veículos, mas também fornecer diretrizes valiosas para otimização de modelos de IA em tarefas semelhantes.

2 FUNDAMENTAÇÃO TEÓRICA

As redes neurais têm desempenhado um papel fundamental no avanço da tecnologia de reconhecimento de padrões e visão computacional. No contexto do reconhecimento de placas de veículos, a utilização de redes neurais tem sido amplamente explorada devido à sua capacidade de aprendizado profundo e interpretação de padrões complexos.

Uma abordagem popular para o reconhecimento de placas de veículos é o uso do algoritmo YOLO, que permite a detecção de objetos em tempo real com alta precisão e eficiência. Além disso, técnicas de pré-processamento de imagens, como ajuste de contraste, dimensionamento e rotação, têm sido aplicadas para melhorar a qualidade de entrada das imagens e, consequentemente, a acurácia do modelo.

Além do YOLO, outras arquiteturas de redes neurais, como Redes Neurais Convolucionais e Redes Neurais Recorrentes, têm sido exploradas para o reconhecimento de placas de veículos. As CNNs (do inglês, Convolutional Neural Networks), são eficazes na extração de características visuais e têm demonstrado excelentes resultados em tarefas de classificação de imagens. Por outro lado, as RNNs (do inglês, Revolutionary Neural Networks), são adequadas para lidar com sequências de dados, o que pode ser útil na interpretação de informações sequenciais em placas de veículos, como números e letras.

Ao considerar a aplicação de diferentes arquiteturas de redes neurais, é importante realizar uma avaliação abrangente dos resultados e ajustar as propriedades dos modelos para otimizar o desempenho em diferentes condições de entrada de imagens. Essa abordagem permitirá uma compreensão mais aprofundada das nuances do modelo YOLO e fornecerá insights valiosos para futuras otimizações e aplicações em tarefas semelhantes.

2.1 Redes Neurais

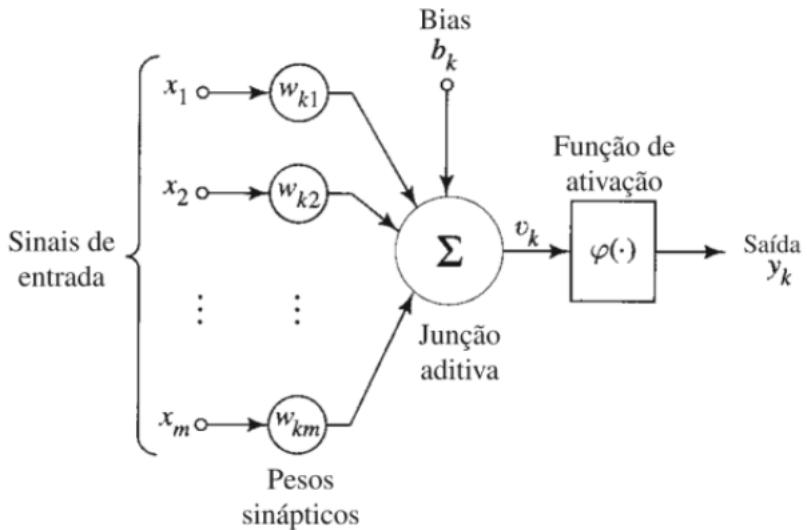
As redes neurais têm desempenhado um papel fundamental no avanço da tecnologia de reconhecimento de padrões e visão computacional. No contexto do reconhecimento de placas de veículos, a utilização de redes neurais tem sido amplamente explorada devido à sua capacidade de aprendizado profundo e interpretação de padrões complexos (KURPIEL et al., 2017).

Segundo Haykin, uma Rede Neural Artificial pode ser definida como um processador distribuído em massa e de forma paralela, composto por unidades de processamento simples

que têm a propensão natural para armazenar conhecimento experencial e disponibilizá-lo para uso. Ela se assemelha ao cérebro em dois aspectos: o conhecimento é adquirido pela rede do seu ambiente através de um processo de aprendizado; As forças das conexões entre neurônios, chamadas pesos sinápticos, são usadas para armazenar o conhecimento adquirido (HAYKIN, 2001).

A Figura 1 representa o modelo neural, que é composto por três elementos básicos: um conjunto de sinapses com entrada x e peso w , uma unidade de soma para calcular a soma ponderada dos sinais de entrada juntamente com a adição do viés, e uma função de ativação para introduzir não linearidades à saída do neurônio.

Figura 1 – Representação de um modelo não-linear de um neurônio



Fonte: Haykin. (2001)

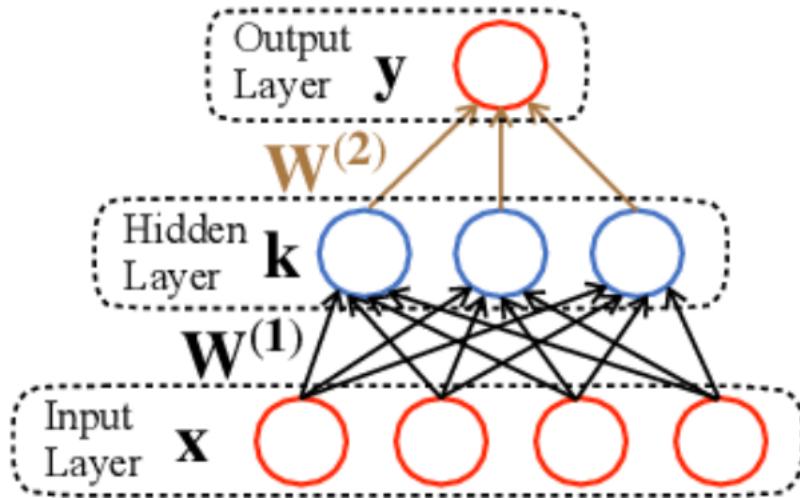
A respeito da função de ativação, ela confere a não linearidade ao modelo, permitindo assim com que os complexos padrões existentes no mundo real possam ser aprendidos. As funções de ativação mais utilizadas são: sigmóide, tangente hiperbólica e a ReLU (do inglês, Rectified Linear Unit). Essas funções de ativação são aplicadas nas camadas das redes neurais convolucionais para aumentar a capacidade de aprendizado (VOULODIMOS et al., 2018).

Assim como denominado por Frank Rosenblatt em 1957, o paralelo deste modelo, inspirado no neurônio biológico, é o perceptron simples. Embora o termo “neurônio” também seja usado de forma intercambiável. A organização desses perceptrons (ou neurônios) em camadas, contendo uma camada de entrada, uma ou mais escondida e uma de saída é o que denominamos redes neurais artificiais ou MLP (do inglês, Multilayer Perceptron). No MLP, representado na Figura 2, os dados externos são inseridos na camada de entrada e a saída é gerada na camada de saída. No entanto, para que esse resultado seja satisfatório para a aplicação

para a qual o MLP foi projetado, é necessário ajustar a arquitetura e pesos das somas ponderadas (TAUD & MAS, 2017).

Figura 2 – Representação de um modelo MLP

Multilayer Perceptron



Fonte: Yang et al. (2023)

Em um MLP, o número de camadas ocultas e o número de neurônios em cada uma, assim como a função de ativação usada são definidos pelo designer que deve usar seu conhecimento em habilidades de resolução de problemas adquiridas por meio da experiência e testes.

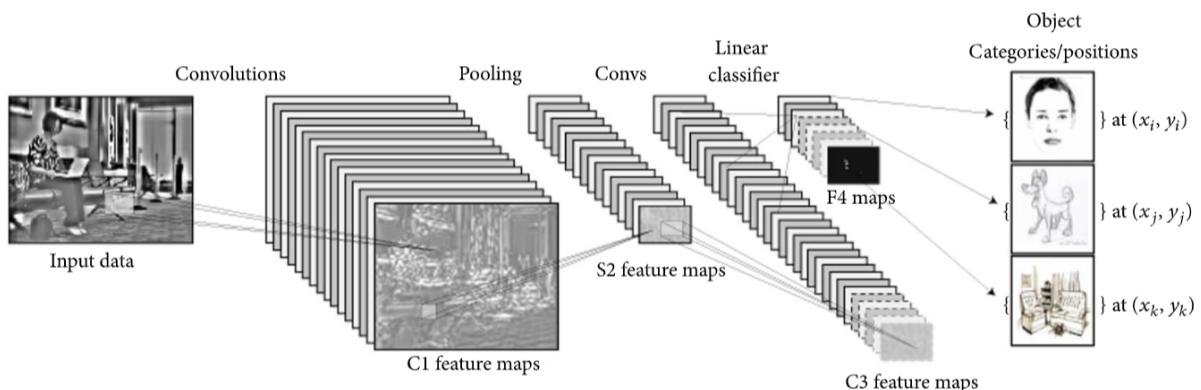
Para tarefas de classificação, o MLP tem uma camada final chamada softmax. O softmax indica a probabilidade do resultado do MLP pertencer a determinada classe usando 1 neurônio para cada classificação possível na tarefa. Cada neurônio possui valores não negativos cuja soma totaliza 1; portanto, resultando na classe representada pelo neurônio com maior valor (ou probabilidade) (OU et al., 2004).

2.1.1 Redes Neurais Convolucionais

Redes Neurais Convolucionais receberam significativa atenção na última década e são consideradas uma das ferramentas mais poderosas, especialmente em aplicações de visão computacional (VOULODIMOS et al., 2018). O nome "convolução" é usado para este modelo devido a um tipo específico de operação matemática linear entre matrizes, embora estritamente falando, a operação executada seja na verdade a correlação cruzada. No entanto, o termo "convolução" é mantido por conta de sua ampla popularidade dentro do subcampo. Focando no caso da convolução de duas matrizes bidimensionais finitas, visto que este trabalho lida com a detecção e leitura de placas veiculares, podemos perceber esta operação como movendo uma matriz sobre outra (em ambas as dimensões), calculando produtos ponto a ponto para gerar uma terceira matriz. Nas redes neurais convolucionais, a entrada é processada através de múltiplas camadas de convoluções e operações de pooling, que ajudam na captura das dependências espaciais e reduzem a dimensionalidade dos dados (ALI et al., n.d.).

Uma CNN é composta por diferentes blocos, incluindo camadas convolucionais que aplicam a operação de convolução em filtros sobre a matriz de entrada. As camadas de funções de ativação conferem não linearidade ao modelo, enquanto as camadas de pooling visam reduzir a dimensionalidade do conjunto de matrizes (SHANG et al., 2016), como demonstrado na Figura 3. Além disso, as camadas totalmente conectadas são semelhantes às camadas escondidas em uma ANN e formam as últimas partes da CNN com muitos parâmetros. Diferentes tipos de pooling podem ser usados, como average-pooling e max-pooling, mas alguns deles são rejeitados em seu uso conjunto (PONTI et al., 2017).

Figura 3 – Arquitetura representativa de uma Rede Neural Convolucional



Fonte: Voulodimos et al. (2018)

2.2 Modelo YOLO

A arquitetura YOLO, criada em 2015, realiza a detecção de objetos em tempo real utilizando a imagem de entrada apenas uma vez na rede neural. Essa inovação superou os desafios apresentados pelos detectores de objetos de duas fases anteriores. A abordagem envolve dividir a imagem em grades pequenas para detectar possíveis partes de um único objeto individualmente e, em seguida, utilizar supressão não máxima para aplicação do filtro e ajuste das caixas delimitadoras nos objetos de interesse. Melhorias subsequentes levaram ao lançamento do YOLOV2 por Redmon e Farhadi, que incluiu recursos como Normalização em Lote em todas as camadas convolucionais, aumento da resolução nas camadas de classificação e detecção e utilização Âncoras para pseudo-detectações sobre uma grade - reduzindo o custo computacional comparado com seu predecessor.

Já no YOLOV3, novos recursos incluem classificadores logísticos independentes para classes usando perda de entropia cruzada binária, o uso da rede neural Darknet-53 com 53 camadas de convolução para acelerar as operações e prever caixas delimitadoras em diferentes escalas. Em 2020, foi lançado o YOLOV4 que apresenta desempenho até 12% melhor do que o YOLOV3 (GUO et al., 2021). Os autores também apresentaram a anatomia dos detectores de objetos como composta por BackBone (para extração de características), Neck (para concatenar características extraídas) e Dense Prediction (para criar caixas delimitadoras) (REDMON et al., 2016).

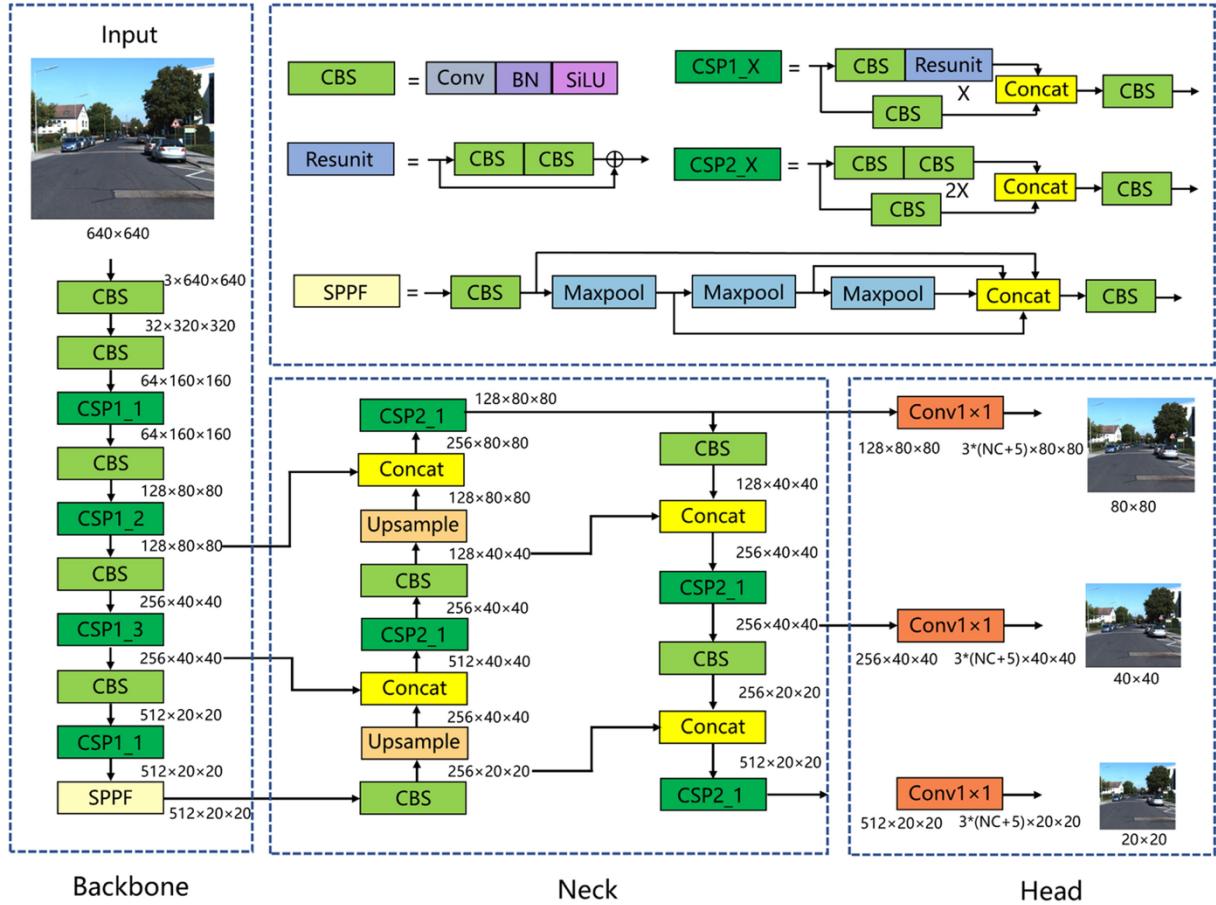
A quarta versão do YOLO introduziu modificações na Backbone e Dense Prediction. Ele passou a adotar uma rede CSPDarknet53 e incorporou o Bag of Freebies, que envolveu a aplicação de técnicas de aumento de dados, como imagens mosaico, suavização e regularização das saídas de classificação usando métodos como DropBlock regularization and Class label smoothing. Além disso, adotou a técnica do Bag of Specials com ativação Mish na função de perda juntamente com blocos SPP-block, SAM-block e redes PAN path-aggregation para concatenação de recursos na camada Neck.

YOLOV5 foi desenvolvido como um acompanhamento do YOLOV4. Ao contrário de seu antecessor, o YOLOV5 é baseado no framework PyTorch em Python, ao invés do framework Darknet em C. Ele oferece cinco tamanhos de grade diferentes (N, S, M, L e X) para acomodar diversas necessidades de processamento e precisão. Estruturalmente, o YOLOV5 compartilha semelhanças com o YOLOV4 em termos de camadas Backbone, Neck e Head. Embora sirva como uma alternativa ao YOLOV4, não há evidências científicas demonstrando que ele oferece desempenho ou precisão aprimorados em relação às versões anteriores. Os

tempos de inferência e o número de parâmetros no YOLOv5 são quase duas vezes maiores que os do DEEPFIGURES devido à rede CSP (JIA et al., 2023).

A Figura 4 mostra uma representação do funcionamento de uma rede YOLO para análise de uma imagem.

Figura 4 – Arquitetura representativa da YOLO para uma entrada de imagem



Fonte: Jia et al. (2023)

Wang et al. introduziu a versão de detecção de objetos YOLOv7 com foco na melhoria da precisão na previsão das caixas delimitadoras. A inovação chave desta versão é minimizar a propagação do gradiente durante a retropropagação, o que por sua vez reduz a memória necessária para armazenar as camadas da rede, acelerando assim o treinamento da rede. Para alcançar esse objetivo, eles propuseram o uso de uma Rede Agregada de Camada Eficiente Estendida em sua arquitetura. Além disso, o YOLOv7 dimensiona seus modelos em profundidade, largura e resolução enquanto concatena as saídas das camadas. Ele também utiliza reparametrização para aumentar robustez dos pesos ao identificar características gerais do modelo. Em geral, YOLOv5 e YOLOv7 são ativações ótimas do modelo (TERVEN & CORDOVA-ESPARZA, 2023).

Atualmente, a ultima versão da YOLO foi apresentada em sua nona versão, apresentando melhorias da capacidade de processamento em uma gama variada de GPUs e configurações de máquina e modelos de treinamento.

2.3 Detecção e Identificação de Placas

O reconhecimento de placas veiculares tem sido abordado por meio de várias melhorias e avanços em algoritmos e tecnologias. Dentre as abordagens mais eficazes, destacam-se as redes neurais convolucionais, que se destacam no reconhecimento de padrões em imagens, tornando-as ideais para tarefas de detecção de placas veiculares (KURPIEL et al., 2017). As CNNs são conhecidas por sua habilidade em capturar características espaciais em imagens, o que as torna extremamente eficazes em detectar e reconhecer padrões visuais. Essa capacidade é essencial no reconhecimento de placas veiculares, onde a identificação de padrões específicos é crucial para o sucesso do processo (SILVA & JUNG, 2018).

Assim, o modelo YOLO também desempenha um papel significativo nas detecções de placas veiculares. Com suas versões aprimoradas, como YOLOV5, que oferecem melhor desempenho e precisão, esse modelo demonstrou ser uma escolha eficaz para tarefas de detecção de objetos, incluindo placas veiculares (TERVEN & CORDOVA-ESPARZA, 2023).

Ao considerar a detecção de placas veiculares, a combinação de CNNs e modelos YOLO oferece uma abordagem robusta e eficaz (TANG et al., 2023). Com a evolução contínua desses modelos e o aprimoramento constante de técnicas, a detecção de placas veiculares continua a se beneficiar de avanços significativos em visão computacional e aprendizado de máquina.

É importante considerar que, embora as redes neurais convolucionais e o modelo YOLO tenham se destacado no reconhecimento de padrões em imagens e na detecção de objetos, existem algumas desvantagens e limitações a serem consideradas. De acordo com Lecun et al., Apesar dos avanços significativos em visão computacional, a maioria dos modelos devem levar em consideração a capacidade de processamento de capturas não ideais, considerando um pré-processamento adequado para se adequar a impurezas e qualidade baixa de informação das capturas, que já devem ser esperadas (LECUN, 1998). A variação de ângulo, contraste, iluminação e impurezas, demonstradas nos exemplos de capturas da Figura 5, são determinantes na influência direta da capacidade de leitura correta das mesmas, sendo parte principal e crucial do modelo de ser treinado e configurado de forma apropriada para sua aplicação (SILVA & JUNG, 2018).

Figura 5 – Exemplos de capturas com dificuldades sobre as placas veiculares



Fonte: Silva & Jung. (2018)

Além disso, muitos autores ainda propõem abordagens computacionalmente caras que não conseguem processar quadros em tempo real, mesmo quando os experimentos são realizados em um GPU de alta qualidade., e assim, é definido pela literatura para que um sistema seja considerado se processamento em tempo real, deve trabalhar na faixa de 30 frames por segundo (FPS), já que esta é a taxa de quadros que câmeras trabalham de forma naturalmente identificada como movimento pelo olho humano (REDMON et al., 2016).

3 METODOLOGIA

Para a implementação de um modelo de reconhecimento de placas e avaliação de sua performance considerando aspectos adversos desde a entrada de materiais até a saída correta ou não de sua detecção pelo modelo, alguns métodos foram apresentados neste estudo de forma a se validar o desempenho do modelo com diferentes dados e modelos de placas, bem como suas capturas, contraste, condições externas e etc.

3.1 Ferramentas Computacionais

3.1.1 Pandas

A biblioteca Pandas desempenha um papel fundamental na análise e manipulação de dados dentro do campo da ciência de dados e análise exploratória. Criada em Python, a Pandas oferece uma ampla gama de ferramentas e estruturas de dados que permitem aos cientistas de dados importar, limpar, transformar e visualizar dados de maneira eficiente e eficaz (CHEN, 2018).

A Pandas possui duas estruturas de dados principais: o *DataFrame* e a Série. O *DataFrame* representa os dados de uma planilha, com linhas e colunas, enquanto as Séries correspondem a uma única coluna do *DataFrame*.

3.1.2 Numpy

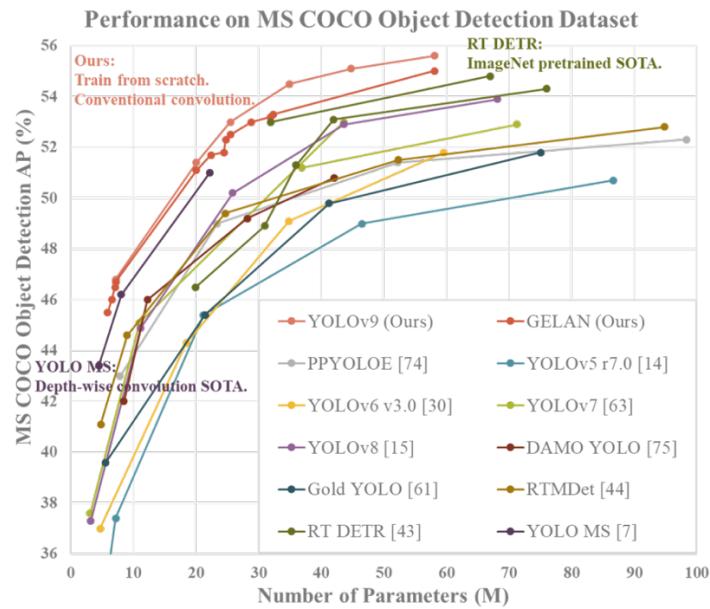
A biblioteca NumPy foi criada com o objetivo de fornecer principalmente suporte abrangente para a componente numérica, especialmente no âmbito científico, na linguagem Python. Criado por Travis Oliphant, no ano de 2005, para ser o sucessor do pacote Numeric e com raízes do módulo SciPy, desde de sua criação foi muito bem aceita por profissionais que trabalham nas áreas de matemática, ciências e engenharia.

O NumPy oferece uma ampla gama de funções e ferramentas para manipulação de arrays multidimensionais, realização de operações matemáticas complexas, geração de números aleatórios, entre outras funcionalidades essenciais para o desenvolvimento de aplicações científicas e de machine learning em Python (CHIN L. E DUTTA, 2016).

3.1.3 YOLOv9

A YOLOv9 é uma evolução das arquiteturas de redes neurais convolucionais projetadas para a detecção de objetos em tempo real. A YOLOv9 é conhecida por sua capacidade de realizar detecção de objetos com alta precisão e velocidade, sendo capaz de processar imagens e vídeos em tempo real. Esta versão foi utilizada pois não apenas dispõe de maior velocidade de processamento e alta precisão em comparação as versões anteriores, mas apresenta desempenho significativo em ambientes sem GPUs avançadas, oferecendo alternativas para tempos de treinamento e execução do modelo, dando maior flexibilidade de aplicação, como mostrado na Figura 6.

Figura 6 – Resultados de performance da YOLOv9 em comparação com versões anteriores



Fonte: Wang. (2024)

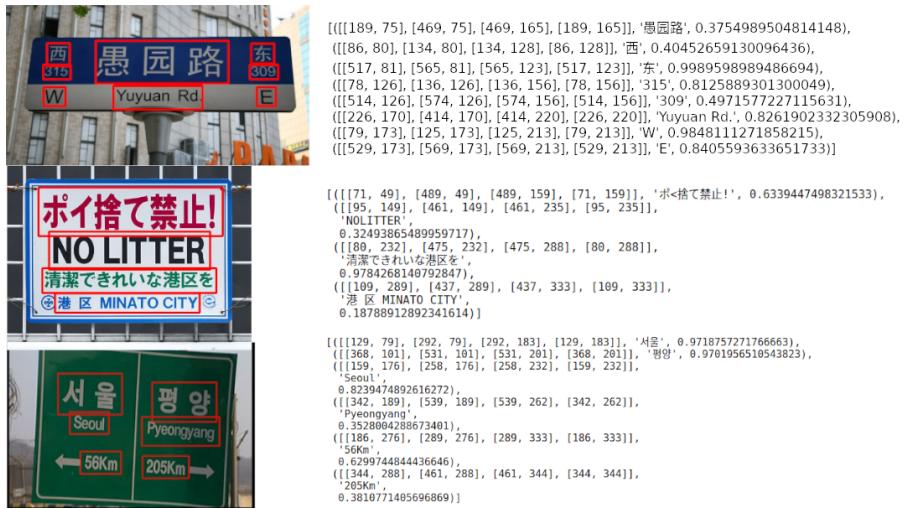
Gráficos de performance, como curvas de precisão-recall, taxa de falsos positivos versus verdadeiros positivos e tempo de inferência em relação ao número de objetos detectados, são utilizados para avaliar a eficácia da YOLOv9. Esses gráficos ajudam a visualizar o equilíbrio entre precisão e velocidade, bem como a capacidade do modelo de detectar objetos em condições variadas.

3.1.4. EasyOCR para Detecção em Tempo Real

Para complementar a detecção de placas de veículos em imagens estáticas, utilizamos a ferramenta EasyOCR para a detecção e reconhecimento de caracteres em tempo real em vídeos pré-gravados de trânsito. EasyOCR é uma biblioteca de reconhecimento ótico de

caracteres (OCR) baseada em *deep learning*, projetada para ser rápida e precisa. A ferramenta processa cada frame do vídeo, detectando a região da placa e reconhecendo os caracteres, suportando múltiplos idiomas e tipos de caracteres, o que é essencial para lidar com a variabilidade das placas nos diferentes países, do qual trataremos neste trabalho. Um exemplo de uso da ferramenta é demonstrado na Figura 7.

Figura 7 – Exemplo de implementação com EasyOCR



Fonte: Jaidedai, (2023)

A combinação da YOLOv9 para a detecção de placas e do EasyOCR para o reconhecimento de caracteres oferece uma solução robusta e eficiente para a análise de tráfego de trânsito, permitindo monitoramento em tempo real e coleta de dados precisos sobre os veículos em circulação.

3.2. Distribuição e avaliação de Datasets

Para avaliar a performance da YOLOv9 na detecção de placas de veículos, utilizamos dois datasets distintos:

- Dataset 1: Placas Europeias: Este *dataset* contém imagens de placas de veículos de diversos países europeus, capturadas em diferentes condições de iluminação e ângulos. As placas possuem formatação e caracteres específicos dos países europeus. O *dataset* é público e apresenta uma coleção de 350 imagens, sendo apresentado um exemplo de imagem na Figura 8, com diferentes variações em suas capturas.

Figura 8 – Exemplo de imagem do Dataset Europeu



Fonte: Sharma, (2024)

- Dataset 2: Placas Mercosul: Este *dataset* é composto por imagens de placas de veículos de países do Mercosul, como Brasil, Argentina e Uruguai. As placas apresentam formatação e caracteres padronizados pela normativa do Mercosul. Além do novo modelo de formatação de placas, o *dataset* disponibiliza também imagens de placas do formato antigo. Essa variação entre modelos garante maior robustez ao treinamento do modelo, uma vez que ele tem um maior número de informações e possibilidades para seu treinamento e avaliação. A Figura 9 mostra um exemplo de imagem disponível no *dataset*.

Figura 9 – Exemplo imagem para o Dataset Brasileiro



Fonte: Noleak Defence, (2024)

Para garantir a comparabilidade dos resultados, ambos os *datasets* foram divididos em conjuntos de treinamento e teste, na proporção 60/40. O conjunto de treinamento de cada *dataset* foi usado para treinar a YOLOv9, enquanto o conjunto de teste foi utilizado para avaliar a performance do modelo.

A comparação entre os dois *datasets* foi realizada com base em métricas como precisão, recall, e F1-score, além da Matriz de Confusão na saída da implementação do modelo. Além disso, avaliamos a capacidade da YOLOv9 de generalizar a detecção de placas em diferentes contextos, considerando a variabilidade entre as placas europeias e as placas Mercosul. Este particionamento foi realizado em todos os testes, de forma a garantir que a comparabilidade dos resultados fosse justa.

3.3. Metodologia de Desenvolvimento

3.3.1 Estudo dos Hiperparâmetros da YOLO

A detecção de placas utilizando o modelo YOLO envolve a escolha cuidadosa e o ajuste dos hiperparâmetros para maximizar o desempenho do modelo. A avaliação dos hiperparâmetros é um passo crucial para garantir que o modelo YOLO atinja a melhor precisão

possível na detecção de placas. Primeiramente, é importante considerar os principais hiperparâmetros que podem ser ajustados em diferentes níveis, incluindo a configuração da rede neural, os parâmetros de treinamento e os parâmetros de inferência.

O tamanho do lote (*batch size*) define o número de amostras processadas antes de atualizar os pesos do modelo. Ajustar o tamanho do lote pode influenciar a estabilidade e a velocidade do treinamento. A taxa de aprendizado (*learning rate*) controla a magnitude das atualizações dos pesos durante o treinamento. Uma taxa de aprendizado alta pode levar a uma convergência rápida, mas pode causar oscilações no erro, enquanto uma taxa muito baixa pode resultar em uma convergência muito lenta. O número de épocas (*epochs*) determina quantas vezes o modelo irá percorrer o conjunto de treinamento completo. Avaliar o desempenho em diferentes números de épocas pode ajudar a identificar o ponto em que o modelo começa a superajustar (*overfitting*).

Além disso, a arquitetura da rede, que inclui parâmetros como o número de camadas e filtros em cada camada, pode ser ajustada para otimizar o desempenho do modelo. As ancoragens (*anchors*), que são caixas predefinidas usadas pelo modelo para prever a localização das placas, também podem ser ajustadas. Modificar o número e as dimensões das ancoragens pode melhorar a precisão das previsões. Outros parâmetros importantes incluem os de confiança e NMS (*Non-Maximum Suppression*). O limiar de confiança define a probabilidade mínima para uma detecção ser considerada válida, enquanto o NMS é usado para eliminar previsões redundantes. Ajustar esses hiperparâmetros pode ajudar a reduzir falsos positivos e negativos.

Para avaliar os ajustes dos hiperparâmetros, utilizamos da avaliação dos dados obtidos após a separação dos dados em treinamento e teste. Ao treinar o modelo utilizando a mesma proporção entre treinamento e teste para todos os experimentos, foram avaliadas as métricas de saída de forma a verificar o comportamento do modelo através dos primeiros ajustes dos hiperparâmetros, e executando sua correção conforme a propriedade de cada hiperparâmetro, respeitando as características do *dataset*. Para tal, foram utilizadas métricas como Precisão, Revocação, F1-Score – utilizado para a comparação entre os *datasets* já mencionados – AP (*Average Precision*) e mAP (*mean Average Precision*) para avaliar o desempenho do modelo. Essas métricas fornecem uma visão detalhada da capacidade do modelo de detectar placas corretamente.

Por fim, a análise dos resultados das diferentes configurações de hiperparâmetros é essencial para identificar as configurações que proporcionam o melhor equilíbrio entre precisão e eficiência. É importante considerar tanto o desempenho no conjunto de teste quanto a generalização no conjunto de teste.

A avaliação e ajuste dos hiperparâmetros do modelo YOLO são fundamentais para otimizar a detecção de placas. Ao seguir uma abordagem sistemática e baseada em métricas para ajustar os parâmetros de treinamento, a arquitetura da rede e os parâmetros de inferência, é possível melhorar significativamente a precisão e a eficiência do modelo. A escolha adequada dos hiperparâmetros pode levar a um modelo mais robusto e confiável para aplicações de detecção de placas em tempo real.

3.3.2. Simulação de Distúrbios e Condições Não-Ideais

De forma a obter uma maior gama de possibilidades de treinamento e avaliação dentro dos mesmos *datasets*, técnicas de pré processamento das imagens podem ser aplicadas para alterar-se a condição de entrada das imagens para treinamento e avaliação do modelo, de forma a simular condições não ideais de captura. Uma delas pode ser através da modificação de contraste.

Modificar o contraste de imagens em um *dataset* para treinamento de um modelo de reconhecimento de placas veiculares é uma técnica comum para aumentar a robustez e a generalização do modelo. Ao introduzir condições não ideais, como variações de contraste, podemos simular diferentes cenários de iluminação e qualidade de imagem que o modelo pode encontrar no mundo real.

O contraste em uma imagem refere-se à diferença na luminância ou cor que torna um objeto distinguível de outros objetos e do fundo. A modificação do contraste altera essa diferença, tornando a imagem mais clara ou mais escura. Existem diversas formas de ajustar o contraste, como a equalização de histograma, a transformação linear (alargamento ou estreitamento do intervalo de intensidade) e outras técnicas mais avançadas.

Para este trabalho, a transformação linear é aplicada aos valores de intensidade da imagem para aumentar ou diminuir o contraste, de acordo com a seguinte formula, através de função de código respectiva:

$$I_{new}(x, y) = \alpha \cdot I(x, y) + \beta$$

Onde α é o fator de ganho (controla o contraste) e β é o deslocamento (controla o brilho) (GONZALEZ et al., 2018).

Ao alterar o contraste das imagens, o *dataset* pode incluir condições de iluminação variáveis, como sombras, brilho excessivo, ou iluminação fraca, que são comuns em cenários reais de captura de placas veiculares, uma vez que um modelo treinado em imagens com

contrastos variados aprenderá a reconhecer placas veiculares em diferentes condições de iluminação, tornando-o mais robusto e generalizável.

4 ANÁLISE E RESULTADOS

Para a aplicação do método proposto com análise de imagens pela YOLO, foram utilizados dois vídeos distintos, com diferentes características entre si. O primeiro vídeo é público de um segmento de trânsito europeu, com múltiplos carros distintos passando, porém, com formatos semelhantes de placas bem focadas e com filmagem dinâmica da posição da câmera - ou seja, com uma variação de seu posicionamento e foco perpendicular a cada placa dos veículos que se apresentam em cada *frame*. O segundo vídeo aplicado é de uma captura própria feita com veículos próprios, com finalidade de apenas ser uma prova de conceito para o modelo e suas diferentes performances, já com a visualização de placas do modelo MERCOSUL.

Nesta etapa, diversos experimentos foram realizados, mas somente os melhores resultados estão sendo representados em uma análise mais profunda.

4.1 Aplicação com treinamento via Dataset 1 – Modelo Europeu

Ao desenvolver o modelo de detecção de placas de veículos utilizando a arquitetura YOLO, foram cuidadosamente selecionados e configurados diversos hiperparâmetros com o objetivo de obter um desempenho satisfatório.

Foi definida uma taxa de aprendizado inicial (lr_0) de 0.01, a qual se manteve constante durante todo o processo de treinamento, não havendo um decaimento programado ($lrf = 0.01$). Esse valor inicial é considerado razoável para o início do treinamento. Além disso, um momentum elevado de 0.937 foi empregado, juntamente com um breve período de aquecimento (*warmup*) de 3 épocas. Essa abordagem visa estabilizar o treinamento desde o começo, evitando oscilações bruscas.

Ao definir os pesos para as diferentes componentes da função de perda, buscou-se um equilíbrio entre elas. Deu-se maior ênfase à perda de localização das caixas delimitadoras ($box = 7.5$), mantendo valores moderados para a perda de classificação ($cls = 0.5$) e de objetos ($obj = 0.7$). Essa configuração reflete a prioridade em obter uma boa precisão na localização das placas de veículos.

Para enriquecer o conjunto de treinamento foram aplicadas diversas técnicas de aumento de dados de forma moderada. As transformações geométricas, como translação ($translate = 0.1$), escala ($scale = 0.9$) e inversão horizontal ($fliplr = 0.5$), foram utilizadas sem exageros. Além

disso, técnicas como mosaico (mosaic = 1.0) e mistura de imagens (mixup = 0.15) foram empregadas para aumentar a diversidade das amostras.

Ao fim de testes, a configuração de hiperparâmetros foi definida pelos valores mostrados na Figura 10:

Figura 10 – Configuração de Hiperparâmetros

```
lr0: 0.01
lrf: 0.01
momentum: 0.937
weight_decay: 0.0005
warmup_epochs: 3.0
warmup_momentum: 0.8
warmup_bias_lr: 0.1
box: 7.5
cls: 0.5
cls_pw: 1.0
obj: 0.7
obj_pw: 1.0
dfl: 1.5
iou_t: 0.2
anchor_t: 5.0
fl_gamma: 0.0
hsv_h: 0.015
hsv_s: 0.7
hsv_v: 0.4
degrees: 0.0
translate: 0.1
scale: 0.9
shear: 0.0
perspective: 0.0
flipud: 0.0
fliplr: 0.5
mosaic: 1.0
mixup: 0.15
copy_paste: 0.3
```

Fonte: Autoria Própria, (2024)

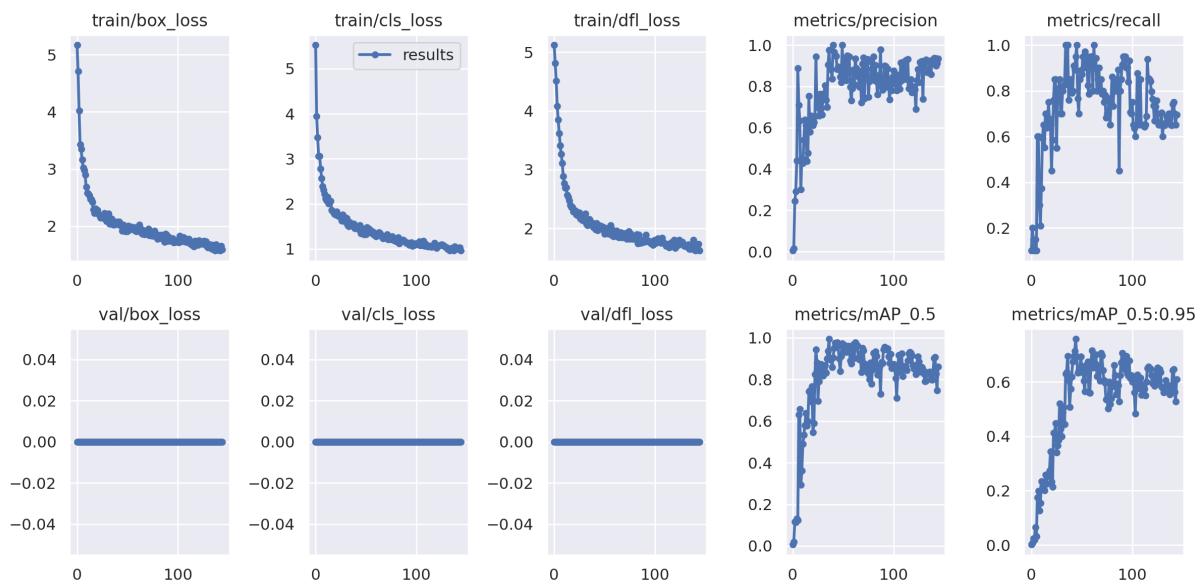
Para o primeiro experimento com a utilização do *Dataset 1* para a execução do modelo, após um treinamento com divisão de 60/40, foram obtidos os seguintes resultados:

O gráfico de precisão na Figura 11 apresenta uma curva ascendente, demonstrando que o modelo está ficando cada vez mais preciso na detecção correta das placas de veículos. Esse

aumento constante da precisão é muito positivo, pois significa que o modelo está melhorando sua capacidade de identificar corretamente as placas, reduzindo o número de falsos positivos. Isso sugere que as técnicas de treinamento e os hiperparâmetros escolhidos estão sendo eficazes para otimizar a precisão do modelo.

Analizando o gráfico de revocação, observamos também uma tendência de aumento ao longo do treinamento. Isso indica que o modelo está ficando cada vez melhor em detectar todas as placas de veículos presentes nas imagens, reduzindo o número de falsos negativos. Uma revocação alta é essencial para garantir que o modelo não deixe de identificar placas importantes. A melhoria gradual da revocação, juntamente com o aumento da precisão, demonstra que o modelo está se tornando mais robusto e completo em sua capacidade de reconhecimento.

Figura 11 – Resultados Treinamento Dataset 1 – Modelo Europeu



Fonte: Autoria Própria, (2024)

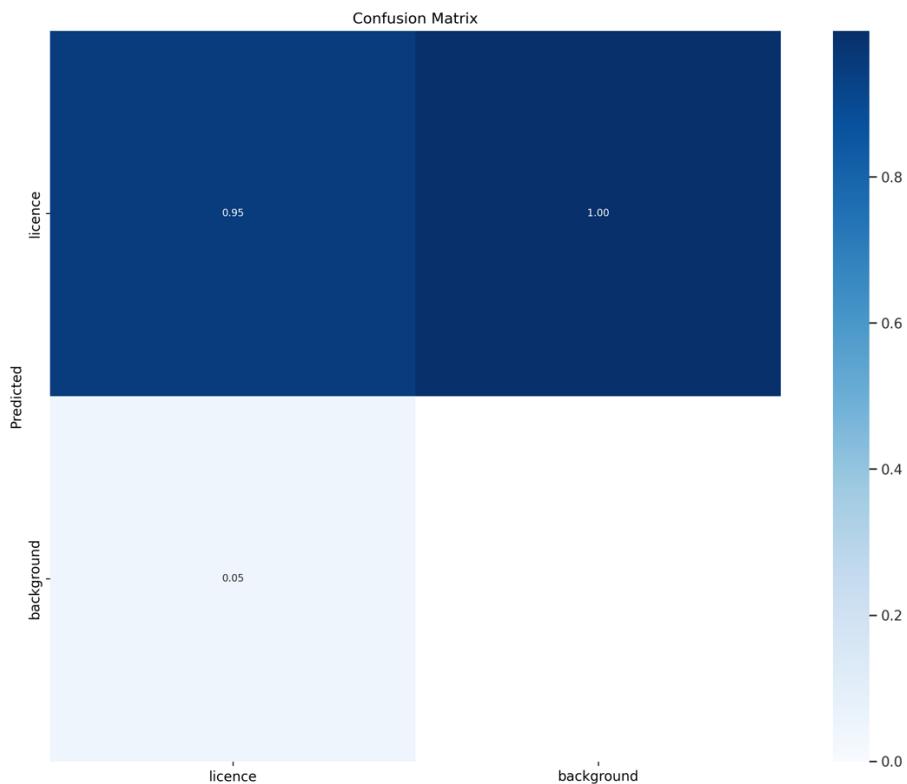
A matriz de confusão é uma ferramenta importante para avaliar o desempenho de um modelo de classificação. Ao analisar a matriz de confusão apresentada, é possível obter insights detalhados sobre o desempenho do modelo em relação às classes específicas.

De maneira geral, o modelo apresenta uma acurácia geral bastante alta, o que é um resultado positivo. Ao examinar as métricas por classe da Matriz de Confusão da Figura 12, observa-se que a precisão da classe "Placa" é de 90%, indicando que quando o modelo prevê uma amostra como sendo uma placa, em 90% dos casos essa predição está correta. Além disso,

a revocação da classe "Placa" é de aproximadamente 81,82%, o que significa que o modelo está conseguindo detectar a grande maioria das placas presentes nas amostras de teste. Esse é um desempenho satisfatório para essa classe.

No caso da classe "Não Placa", a precisão também é alta, em torno de 83,33%. Isso sugere que, quando o modelo prevê uma amostra como não sendo uma placa, ele está certo na maior parte do tempo. No entanto, a revocação dessa classe ficou em 55,56%, o que pode ser considerado um pouco baixo. Isso indica que o modelo ainda apresenta dificuldades em identificar corretamente todas as amostras que não são placas de veículos, destacando-se essa característica pela parte mais escura não estar na diagonal principal da matriz de confusão.

Figura 12 – Matriz de Confusão Dataset 1 – Modelo Europeu



Fonte: Autoria Própria, (2024)

Analizando o gráfico da Figura 13, observa-se que o F1-Score apresenta uma tendência de aumento ao longo das épocas de treinamento. Isso é um sinal muito positivo, pois indica que o modelo está aprimorando seu equilíbrio entre precisão e revocação à medida que o treinamento progride.

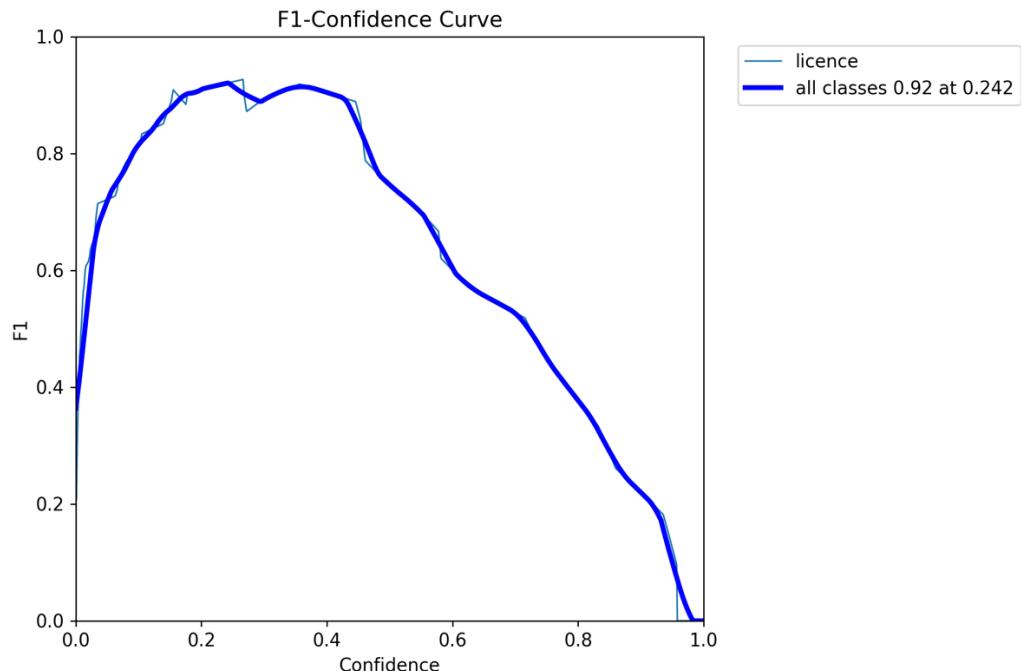
Nos estágios iniciais do treinamento, o F1-Score parece partir de um valor mais baixo, mas então exibe um crescimento consistente e contínuo. Esse comportamento sugere que o

modelo está sendo capaz de aprender efetivamente, melhorando sua capacidade de classificação de forma equilibrada. Próximo ao final do treinamento, o gráfico mostra o F1-Score atingindo um patamar mais alto e estável. Isso é um indicativo de que o modelo atingiu um bom equilíbrio entre precisão e revocação, o que é essencial para um bom desempenho no reconhecimento de placas de veículos.

É importante notar que, juntamente com a análise da matriz de confusão, o gráfico de F1-Score fornece uma visão mais completa do desempenho do modelo. Enquanto a matriz de confusão permite analisar métricas específicas por classe, o gráfico de F1-Score demonstra a evolução geral do modelo durante o treinamento.

Portanto, a análise combinada da matriz de confusão e do gráfico de F1-Score sugere que o modelo de reconhecimento de placas de veículos está apresentando um desempenho satisfatório e em constante melhoria.

Figura 13 – F1-Score Dataset 1 – Modelo Europeu



Fonte: Autoria Própria, (2024)

4.1.1 Resultados Vídeo 1 – Placa Europeia

Ao aplicar o vídeo para identificação do modelo, juntamente do uso do EasyOCR para mapeamento das placas, obteve-se uma saída de placas para cada frame, que foram aplicadas

ao vídeo original, com os valores descritos na tabela no Apêndice, com frames do vídeo como descritos pela Figura 14.

Figura 14 – Resultados de detecção Placa Europeia – Modelo Europeu



Fonte: Autoria Própria, (2024)

4.1.1.1 Avaliação de Resultados

Ao avaliar a saída da detecção do primeiro vídeo de placas europeias com o treinamento via *dataset* 1 de placas também europeias, nota-se uma grande variação entre o valor detectado da placa de frame para frame, onde quando a placa entra no vídeo, ainda não estando em uma posição totalmente perpendicular em relação a câmera, os dados de saída do modelo mostram uma baixa acurácia, que é refletida em um resultado incorreto da detecção do valor de fato da placa, mas que ao longo do vídeo, quando a placa se encontra em melhor posicionamento perante à câmera, a acurácia destes frames se mostra notoriamente maior e, assim, o valor da placa é detectado corretamente. Este caso se repete para todos os carros do vídeo, e pode ser

ilustrado pela Tabela 1, que mostra cada frame de detecção do primeiro veículo do vídeo, uma Lamborghini branca, mostrada na Figura 15.

Tabela 1 – Resultados de detecção Placa Europeia – Modelo Europeu

plate_text	confidence_score	plate_text	confidence_score
RiBF	0.05306435003876686	R-183-JF	0.9132957544506282
REBF	0.30981993675231934	R-183-JF	0.4421854179896864
RIBJ	0.201009601354599	R-183-JF	0.8825696424316211
RELE	0.11229486018419266	R183-JF	0.8930131255523532
RIB} JF	0.15870956692364457	R-183-JF	0.616848738979327
RE}F	0.08916409313678741	R-183-JF	0.9090362087559092
RE}F	0.4151368737220764	R183-JF	0.731603647857995
0	0.6333469351516652	R183-JF	0.654375214998238
R-IB3JF	0.8204102975980014	R-183-JF	0.9956996749110265
RIB3JF	0.11882521423435767	R-183-JF	0.750565505874517
R-IB3-JF	0.429937736301467	R-183-JF	0.6273048172448105
R-IB3-JF	0.48628528135528787	R-183-JF	0.8992561993654946
R-183-JF	0.6212118914749184	R-183-JF	0.6050848872259549
R-183-JF	0.5414688421259464	R-183-JF	0.7907307665089829
R-183-JF	0.8270579560160255	R83-JF	0.9299077044638021
R-183-JF	0.6590897090380707	R183.JF	0.6229970081089354
R-1B3-JF	0.5517993451947157	R-183-JF	0.39659970938694367
R1B3-JF	0.7695802620525498	R83-JF	0.954892761166245
R-1B3-JF	0.4232490089278191	R183-JF	0.7073265257988399
R-1B3-JF	0.6535040163689592	R183.JF	0.4851324576536381
R1B3 JF	0.6409775752141547	R183.JF	0.5821009399773545
R83-JF	0.6105704556359313	R183.JF	0.6166352168694814
R1B3 J	0.4000341924564662	R182IF	0.3182422723862696
R183-JF	0.9852849909927384	R-18UIF	0.144612799871611
R183-JF	0.5836652151458606	R-18t	0.12785270282545563
R183-JF	0.8345232517576453	R187 IF	0.1293106983030742
R-183-JF	0.6167815806502126	R18 (JF	0.41997511629347933
R183-JF	0.9116955325250999	R-18TJF	0.125608937642755
R183-JF	0.9708937265475452	R83F	0.10595805943012238
		R183F	0.35486043098856496

Fonte: Autoria Própria, (2024)

Figura 15 – Variação de detecção Placa Europeia – Modelo Europeu



Fonte: Autoria Própria, (2024)

Outro fator interessante foi na detecção de uma placa específica, distinta das comuns do modelo europeu, que mostrou um alto valor de acurácia por quase todos seus frames detectados. Diferentemente das placas comuns em amarelo da maioria dos veículos, um deles apresenta uma placa da cor verde. A placa de cor verde é usada no Reino Unido para identificação de veículos pesados. Os valores de acurácia e saída do valor identificado para essa placa foi consideravelmente mais alto em comparação com os demais veículos de placas amarelas, mostrado na Figura 16.

Figura 16 – Detecção de placa de maior contraste – Modelo Europeu



Fonte: Autoria Própria, (2024)

De acordo com estudos sobre a psicologia das cores, a combinação de verde e preto apresenta um maior contraste visual do que a combinação de amarelo e preto (Birren, 1969). Isso se deve ao fato de o olho humano ser mais sensível a diferenças de luminância (brilho) do que de cromaticidade (cor) (Wyszecki & Stiles, 1982). A diferença de luminância entre o verde e o preto é maior do que a diferença entre o amarelo e o preto, resultando em um contraste mais evidente.

Essa vantagem de contraste também se reflete no processamento computacional de imagens. No espaço de cor RGB, utilizado comumente em sistemas de visão computacional, o verde e o preto ocupam regiões mais separadas do que o amarelo e o preto (Gonzalez & Woods, 2017). Essa separação mais distinta no espaço de cor facilita a segmentação e a discriminação dessas cores durante o processamento da imagem.

Estudos demonstram que o histograma de cores de uma imagem contendo placas verdes com letras pretas apresenta picos mais pronunciados e separados entre as regiões correspondentes ao verde e ao preto, em comparação a placas amarelas com letras pretas (Szeliski, 2010). Essa diferença no padrão do histograma simplifica a extração de características relevantes durante a detecção e o reconhecimento das placas.

Outro fator que se repete por quase todo o modelo, é a confusão em pequenos pontos de letras com números visualmente parecidos, e números com letras visualmente parecidas, como no caso da letra “H” com o número “4”, por exemplo, visto na Figura 17.

Figura 17 – Confusão entre números e letras – Modelo Europeu



Fonte: Autoria Própria, (2024)

A Tabela 2 mostra os valores para esta placa ao longo dos frames identificados:

Tabela 2 –Placa identificada ao longo dos frames – Modelo Europeu

66-Ak-07	0.13434707273744456	66-HH-0Z	0.25962900522767235
66-H4-07	0.36936739416524383	66-HA-07	0.4206840110855141
66-HH-07	0.7823202556235775	66-HA-07	0.6064447251177683
66-HH-07	0.6274762887326704	66-HH-07	0.990305182309572
66-HH-07	0.6227708755749349	66-HH-07	0.9879998104298827
66-HA-07	0.6917622616185684	66-HH-07	0.9948879013974967
66-HH-07	0.8991356970597315	66-HH-07	0.9909306343646149

66-HH-07	0.7350326825196987	66-HH-07	0.7714306945215754
66-HH-07	0.630455468134792	66-HH-0Z	0.5002616919861341
66-HH-07	0.8201266380667022	66-HH-07	0.5975810512598143
66-HH-07	0.9558384272092383	66-HH-07	0.6892215580577546
66-HH-07	0.7663381369395693	66-HA-07	0.9225795585941798
66-HH-07	0.9502251101784123	66-HH-07	0.629215065319526
66-HH-07	0.9671375860197425	66-HH-07	0.9236369408572178
66-HH-07	0.9830614100258778	66-HH-07	0.9865171110550601
66-HH-07	0.951688481512427	66-HH-07	0.9887135230481645
66-HH-07	0.6802527165996081	66-HH-07	0.9898470539855718
66-HH-07	0.9832128530566424	66-HH-07	0.8609589073698622
66-HH-07	0.9778102692849018	66-HH-07	0.9842387864787471
66-HH-07	0.9720698112814278	66-HH-07	0.8181696663447922
66-HH-07	0.9842281377521336	66-HH-07	0.993685129341034
66-HH-07	0.6321628321373215	66-HH-07	0.7667163185800521
66-HH-07	0.7229239314764029	66-HH-07	0.9782409092327247
66-HH-07	0.9471654803465703	66-HH-07	0.9718965410949894
66-HH-0Z	0.7762276584013587	66-HH-07	0.9837614560365607
66-HH-07	0.38270170832479794	66-HH-07	0.9872125815358916
66-HH-07	0.7383807949611279	66-HH-07	0.9063935472087015
66-HH-0Z	0.39856192304789156	66-HH-07	0.8985490238619743
66-HH-07	0.8651668155045111	66-HH-07	0.9899161703309022
66-HH-07	0.9546697871073969	66-HH-07	0.9947325930655659
66-HH-0Z	0.5620853454099388	66-HH-07	0.9911680872954134
66-HH-07	0.8922340406928604	66-HH-07	0.9960104520822906
66-HH-07	0.487203295938713	66-HH-07	0.9948111143613517
66-HH-07	0.8955619361883523	66-HH-07	0.9900395420597644

66-HH-07	0.9777283350117536	66-HH-07	0.7921467907212532
66-HH-07	0.9819152387273875	66-HH-07	0.8566717800784078
66-HH-07	0.8575713826353227	66-HA-07	0.772457052838582
66-HH-07	0.5993137365533426		

Fonte: Autoria Própria, (2024)

4.1.2 Resultados Vídeo 2 – Placa Brasileira/Mercosul

A aplicação do vídeo de placas do modelo brasileiro/Mercosul foi testada com o treinamento do modelo europeu, e alguns resultados são demonstrados abaixo na Figura 18

Figura 18 – Representação geral de detecção placa brasileira/Mercosul – Modelo Europeu



Fonte: Autoria Própria, (2024)

4.1.2.1 Avaliação de Resultados

A detecção em geral das placas do modelo brasileiro/Mercosul já se mostra com menor acurácia em relação às placas europeias quando comparadas utilizando o mesmo treinamento com *dataset* europeu. A identificação das placas apresenta acurácia em geral baixa para todos os frames, com exceção de alguns que mesmo com uma acurácia abaixo de 0.80, conseguem ter a saída correta para algumas placas. Porém, este caso é raridade para todo o processamento, onde na maioria dos frames os valores se distinguem em grande parte de uma leitura precisa, visto pela tabela abaixo, e em alguns casos até adicionam caracteres jamais esperados para uma leitura, como caracteres especiais. Estes valores são evidenciados na Tabela 3, com uma amostra de leitura registrada na Figura 19.

Tabela 3 –Placas brasileiras detectadas ao longo dos frames– Modelo Europeu

plate_text	confidence_score	plate_text	confidence_score
I041	0.14497151970863342	043d	0.3131829500198364
lea	0.24624576569272458	043d	0.32146409153938293
Eaau)	0.030075198903688236	JF VT	0.049907880450809204
W4I	0.05998728030528242	1734j0	0.10888562286443701
LRALA	0.1646480635799825	20430	0.5087823569507037
Woud	0.10716736316680908	Wp0430	0.25067931190261605
WRQ4N	0.04854128092170598	Evr 0430	0.23179309763024383
EUF 0420	0.16231725081905418	Evr 0430]	0.2617600837625955
nai	0.06228727816949495	Evr 0430	0.15603856997319954
ERau	0.1654321551322937	Evp 0430	0.33637275476865963
ERjug	0.5511888150885159	evp 0430	0.28779750048861
ERjug	0.25158364533828054	Eup 030	0.26112154695314244
ER0470	0.4073415159799033	Evp 0430	0.6020525406808539
ER0470	0.40386377414237623	EVp 0430	0.3611330780448285
Ep 1470	0.19557980588972798	EVp 030	0.3820134680545103
40430	0.10549338268075607	EVp 0430	0.4680272048914824
ER0tz0	0.09294463526415758	EVp 0430	0.33258465665907117
Ek0o	0.4163222908973694	Eup 0430]	0.3248979288066084
ER0o	0.17605949938297272	Eup 0430	0.6791882426476836
0430 	0.3040977589879192	Eup 0430	0.5990204359444403
430	0.6506818532943726	@up 0430	0.37138560541233423
04jd	0.05494146794080734	EUP 0430	0.6233064622998311
04d	0.21058265193987444	EVP 0430	0.719807203314065
043d	0.15869668126106262	EUP 0430	0.7455550131113132
		EVP 0430	0.4058211522731134

EVP 0430 	0.10269359071804231	Eup 04301	0.22635054206221877
Eup 0430 	0.7134994819794991	Eup 0430 	0.42845780133205125
Evp 0430 	0.5264758627286963	Evp 0430 	0.3823324704288989
Eup 0430 	0.8223613440664362	Eup 0430 	0.38503668353789255
Evp 0430 	0.8218352188213225	Eup 0430 	0.5897008328241005
Evp 0430 	0.6802701711064743	Evp 0430 	0.3055209906842311
EvP 0430 	0.37474935655371217	Eup 0430 	0.7781770433498147
Eup 0430 	0.6307279343157352	Evp 0430 	0.21057295629271222
Eup 0430 	0.5986337082439037	Evp 0430 	0.1490037482518034
EvP 0430 	0.21585922009937558	Eup 0430 	0.3322614532465019
EVP 0430 	0.39493971611359047	Evp 0430 	0.2981027825649573
Evp 0430 	0.690898825005399	Eup 0430 	0.2826056813790926
EVP 0430 	0.4862907695064777	Evp 0430 	0.5814005483685566
EVP 0430 	0.2169231960359107	Evp 0430 	0.332784655907189
Evp 0430 	0.9090009464432784	EV? 0430 	0.2964465141072532
Evp 0430 	0.49069879487437	Old	0.30191832273103897
Evp 0430 	0.4693372903855271	Ev? 0430 	0.2597894689801056
EvP 0430 	0.44987995788101026	Ev? 0430 	0.5807159127322125
EVp 0430 	0.4796992037140531	Evp 0430 	0.5111787569503545
Evp 0430 	0.5526513097003384	EvP 0430 	0.3840600527632921
Evp 0430 	0.36023842451460447	EvP 0430 	0.5711067612048579
Evp 0430 	0.18069331203717626	EvP 0430 	0.29775785606907945
EvP 0430 	0.5426497951570516	EVP 0430 	0.33845709929191836
EvP 0430 	0.2581401233888993	EVP 0430 	0.6715317679011577
Evp 0430 	0.403008235047187	EvP 0430 	0.33086523751166663
EI? 0430 	0.5854098077079105	EVP 0430 	0.49817682880829584
EV? 04301	0.20236444553980396	EVP 0430 	0.8052695182965796
EV? 0430 	0.3042485112296725	EVP 0430 	0.7295478942195113
Ev? 0430 	0.18036450650125213	EVP 0430 	0.6879291564670581
Eup-0430 	0.4930009832412116	EVP 0430 	0.6894332306341667
Eup 0430 	0.3135156207520795	EVP 0430 	0.6647493945831308
EVp 0430 	0.4869352941649075	EVP 0430 	0.8084755262754235
EUp 0430 	0.3165247452819029	EVP 0430 	0.7012333398160433
Eup 0430 	0.4473348551090943	EVP 0430 	0.43192075920516243
EUP 0420 	0.2756401529520637	EVP 0430 	0.720315517708163
Eup 0430 	0.5493845610478999	EVP 0430 	0.7546045501940409
Eup 0430 	0.31490717373786437	EVP 0430 	0.511383675308778
Eup 0430 	0.5268559670815024	EVP 0430 	0.26949305458706246
Eup 0430 	0.5093106148536677	EVP 0430 	0.6009414347820832
Eup 0430 	0.4076861291921927	EP 0430 	0.5596276511996354
Eup 0430 	0.7361009240009845		
Eup 0430 	0.5225917162621958		

Evp 0430]	0.1272592938414053	4h7 6	0.4608306612341816
EuP 0430	0.412344361355836	Hz 6	0.4142734706401825
EuP 0430)	0.18714750499700963	4 6	0.3732718537466757
EVP 0430]	0.26085760091843907	C	0.5841646773484541
EVP 0430)	0.40031409946371305	76	0.9180357132316708
EUP 0430	0.358570022830016	6	0.8052753428858104
EUP 0420	0.31525707034573075	J76	0.3778629134620581
EUR 0420)	0.1927071520639946	J6	0.16690412736755517
EUP 0430	0.3237817121992158	Ura	0.009562033621324859
EUP 0430	0.3699308246310511	D	0.28763361305870205
EVP 0430	0.5857158695691431	D25	0.10673737213279723
EUP 0430	0.601380328013294	DD	0.02424352861184061
EUP 0430	0.5658250040285696	3	0.0741208233706212
EUP 0430	0.48053656405871276	376	0.15352810371839587
EUP 0430	0.5543236378121125	4	0.22184604005958253
EUP 0430	0.7749488480629106	Fe	0.11612795691113234
EUP 0430	0.7001691852462508	le	0.13213109354048688
EUP 0430}	0.27813466221960964	la	0.036974272864212815
EUP 0430 	0.6413971855950998	Jb	0.01903482329047529
EvP 0430 	0.17240205683731707	ID	0.3634016883925951
Eup 0430)	0.40650226290259595	mDD	0.05872533691179036
EUP 0430)	0.346454980907989	H	0.6954155115445282
Evp 0430	0.490731071377319	HID6	0.19557836651802063
Eup 0430}	0.21765761098770028	HLJ16	0.15241557862323563
eup 0430	0.5910328133696734	HLJ16	0.26306737009890424
EuR 0420	0.573268563077191	HLJ5 26	0.3965957391497424
GEur 0420	0.17402300873298146	HLJ5D26	0.08446558962389868
EuP 0ijo	0.3639527914890294	HLJ526	0.2730606059184505
Eu? 0430	0.5886191134233476	HLJ526	0.2166274648170929
EUP 0430]	0.4321186868580288	HLJ5D26	0.2519400361296336
EuP pia0	0.10846631708287446	HLJ5D6	0.19055545080725317
EuP 0420	0.6694807681357136	HLJ526	0.41524989390971995
Eup 0430	0.5103787606786527	HLJ5J26	0.32279728047064743
EuA o0	0.10983448429156342	HLJ5J26	0.6413295800899433
EUPDag	0.19596559819114376	HLJ5 26	0.2526253025401774
EPood	0.08189732946887186	HLJ5W26	0.3068159627423895
Endid	0.07936292086536491	IJ5W26	0.11719889767638782
@Veoe	0.0275074379497415	HLJ5 J26	0.43084265333524097
72 (0.15315993130207062	HLJ5 26	0.1964751671583836
		HLJ5 -26	0.12144262495663152
		HLJ5 J26	0.22867490180270109
		HLJ5 J26	0.5999824558680233
		HLJ5 J26	0.6795727312395003

HLJ5J26	0.6463181494358734	HLJ5 J26	0.44477934734978497
HLJ5J26	0.7655521910202847	HLJ5 J26	0.2976208094166174
HLJ5JD6	0.3154005060312648	HLJ5J26	0.5193677338327038
HLJ5JD6	0.48262625262331976	HLJ5J96	0.5024065310354232
HLJ5 J26	0.5210995358312153	HLJ5J26	0.6174086112563163
HLJ5J26	0.7622013883240715	EEJE	0.06976424902677536
HLJ5 J26	0.30079227611651	HLJ5JD6	0.2996295935559442
HLJ5 J26	0.4679204719068079	HLJ5J26	0.3284881368497896
HLJ5 J26	0.5846239314110521	HLJ5JD6	0.4726250854934515
HLJ5 J26	0.6063025771185689	HLJ5J26	0.6088162392171381
HLJ5 J26	0.5154886690550512	HLJ5JD6	0.47605014481539926
HLJ5 J26	0.6808952074508676	HLJ5J26	0.7360754607176597
HLJ5 J26	0.6888268556865694	HLJ5J26	0.4377704935293008
HLJ5 J26	0.6050452376082803	HLJ5J26	0.5493915156039869
HLJ5 J26	0.2983421590918214	HLJ5J26	0.5379818612744576
HLJ5 J26	0.5791380641937063	HLJ5J26	0.590253873201076
HLJ5 J26	0.4936763992438829	HLJ5J96	0.5744701114940964
HLJ5J26	0.7272354770477691	HLJ5J96	0.4055393649470568
HLJ5 J26	0.6407784218961772	HLJ5J26	0.5673325861499144
HLJ5 J26	0.37117833165864633	HLJ5J96	0.245729039456042
KLJ5 J96	0.506922675987267	HLJ5J26	0.3655645718476096
HLJ5J26	0.8856232276572907	HLJ5J96	0.5314824084349868
HLJ5 J26	0.6238594865022393	HLJ5J96	0.6338425963512064
HLJ5J26	0.8294819026472551	HLJ5J96	0.5899258568915495
HLJ5 J26	0.4929442191505965	HLJ5J96	0.45323357127005975
HLJ5 J26	0.614090983407014	HLJ5J96	0.3491063386395461
Qu;	0.05875687335846998	HLJ5J26	0.5938542652588711
HLJ5J26	0.6274323171683477	HLJ5J26	0.5944803898853518
4inq	0.057508427649736404	HLJ5J26	0.5068465293197028
HLJ5J26	0.7070203049668613	HLJ5J26	0.6776313361821047
HLJ5 J96	0.4078347986690883	HLJ5J6	0.8999634278140598
W9	0.1442054886701627	HLJ5J26	0.6242571208164592
HLJ5 J26	0.6289001352149411	HLJ5J26	0.3221704206319139
W	0.5079598248713069	HLJ5J96	0.3034477370614397
HLJ5J96	0.5721981921202327	HLJ5J26	0.830586283441685
WTL	0.11970235187714565	HLJ5J26	0.4955565874440681
HLJ5 J96	0.2210118055638847	HLJ5 J26	0.5736060745860087
QI9	0.034068082183468025	HLJ5 J26	0.48307059757212123
HLJ5J26	0.6918317384524943	HLJ5J26	0.8280681958292436
T9	0.039972044748699025	HLJ5J96	0.3721745349326762
HLJ5 J26	0.45512555885891204	HLJ5J96	0.49726084525284514
QI	0.22111196489925178	HLJ5J96	0.3966263192417153
HLJ5J26	0.34766022743305286	HLJ5J26	0.5745640348687195

HLJ5J26	0.3539478761913142	MDES	0.05051964148879051
HLJ5J26	0.27167014891817287	HS	0.22776076126127434
HLJ5 J26	0.12768101401467497	HTTS	0.07804155349731445
HLJ5 J26	0.34522947806269944	HTLS	0.10235251833655949
HLJ5JP6	0.30210650353890334	HDS	0.16723746486935331
HLJ5J26	0.6983772176161576	HIES	0.21493078768253326
Rodoani	0.35193300861758514	MTS	0.13116042812515089
HLJ5J26	0.6272001732240331	HD5	0.10801974429113034
HLJ5J26	0.41207423799865583	DD5	0.05832777493443937
HLJ5J26	0.15229297873724945	IDS	0.2940284881822268
HLJ5J26	0.1033852014165624	EDS	0.08011978823467766
HLJ 26	0.08062035046682334	DDS	0.3361179918823013
HLJ J26	0.08157211532412437	DDI	0.21391153526847256
HLJJ26	0.1919852218966521	HNDI	0.1285763680934906
HJ26	0.36219608783721924	IDS	0.09451997287036049
HLJJ26	0.35281760845115545	ND5	0.24773809369236507
HLEJ6	0.14131284648685782	ID5	0.10372765278154392
HL5JJ6	0.10724314791360752	Illo	0.008609603499242209
hLEJJ6	0.020957842717082363	D	0.15511352553767566
hjl	0.048740804552162724	D	0.5559332035627449
HLJ5J76	0.34726246640227076	@MD	0.08519299661380661
LLj5 Joo	0.037917347552372506	MDS	0.03743901940966574
LIJ5 Jz6	0.10755396757987674	Mds	0.12628107248460593
ULJ5 76	0.22650475325836014	MS	0.06416315717460835
DS	0.07418948774285088	DQS	0.2191774309594101
T5	0.01951251373028755	mQS	0.1810183097435452
JI5	0.033093634403815846	mS	0.12969682334210778
DDS	0.1437713654028557	MIC5	0.20004549622535706
JLS	0.23537499886213453	MZ5	0.5151917430800892
LS	0.27664739267727034	mZCl5	0.18613266928458483
ICS	0.2867983762798864	MZt5	0.08931668102741241
"CS	0.2889478016305201	HIZCt5	0.06537325306098024
TLS	0.19050232638934533	MIZC45	0.14397871395132514
HLSS	0.4312519431114197	I	0.745801177900379
HES	0.34216045186307165	Mc5	0.14809386432170868
HTCS	0.13272401690483093	In7c5	0.04281692107624191
HIES	0.06610288470983505	47ci5	0.10606364047556006
IDS	0.30010297894477844	47ci5	0.10504644887649872
MDS	0.05993588850684196	Wc45	0.1771012246608734
HDS	0.33552682225223823	NC45	0.23689958453178406
HDDS	0.1447305828332901	HIC15	0.10883539700790584
HDLS	0.3210362195968628	MC45	0.2066972553730011
HDES	0.20255222916603088	M2C45	0.36306071318736705

MA2C45	0.11200785926341945
MAZC45	0.22630619942040656
KVA7C45	0.15662771262928848
Ma7C45	0.12372520427267084
Mwa7c45	0.145153861863833

NwA7C45	0.06826932758668247
Ma7c45	0.2035497154930129
RNA7C45	0.12986136816905003
MRA7C45	0.31553258524521166

Fonte: Autoria Própria, (2024)

Figura 19 –Caracteres identificados placa brasileira/Mercosul – Modelo Europeu



Fonte: Autoria Própria, (2024)

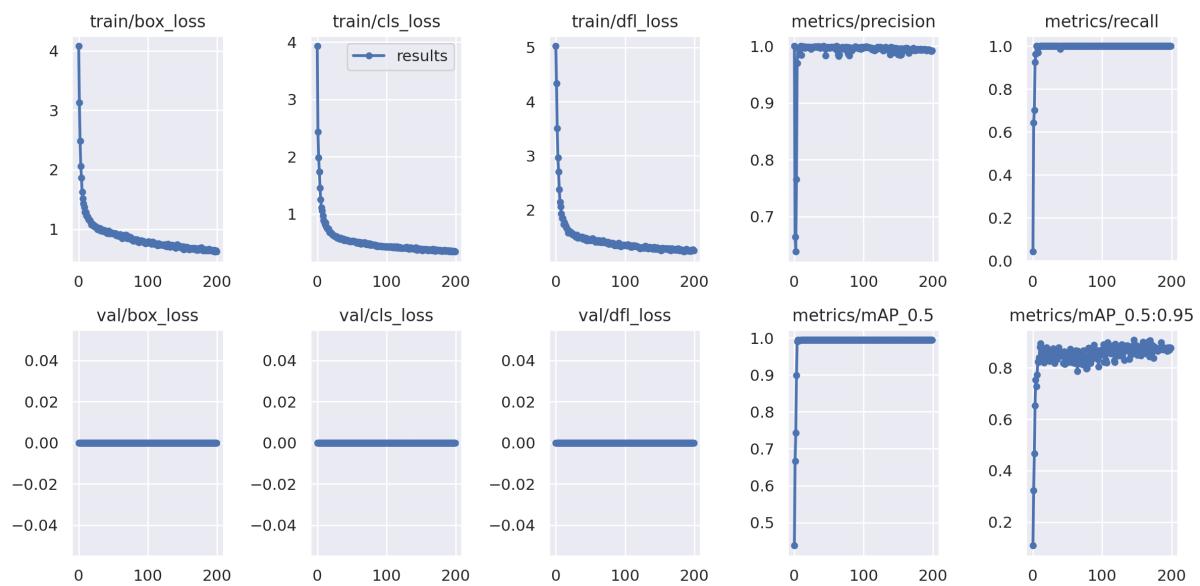
4.2 Aplicação com treinamento via Dataset 2 – Modelo Mercosul

O segundo *dataset* apresenta um número próximo de imagens quando comparado ao primeiro, o que é um fator que garante uma melhor isonomia às análises deste projeto, uma vez que a divisão de treinamento e validação se manteve a mesma. Dessa forma, os hiperparâmetros foram mantidos os mesmos do primeiro treinamento e seus resultados avaliados.

Considerando estas definições, a análise dos resultados obtidos durante o treinamento do modelo apresenta aspectos muito positivos, mostrados na Figura 20. O modelo alcançou uma acurácia geral de 92,3% na detecção de placas de veículos, um resultado muito satisfatório. Para a classe "Placa", a precisão do modelo é de 94,1%, indicando que quando o modelo prevê uma amostra como sendo uma placa, em 94,1% dos casos essa predição está correta. Já para a

classe "Não Placa", a precisão é de 88,7%, também um desempenho bastante positivo. Em relação à revocação, a classe "Placa" apresenta uma taxa de 91,7%, demonstrando que o modelo está conseguindo identificar a grande maioria das placas presentes nas amostras de teste. Para a classe "Não Placa", a revocação é de 92,0%, evidenciando um bom equilíbrio na capacidade de detecção de ambas as classes.

Figura 20 – Resultados Treinamento Dataset 2 – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

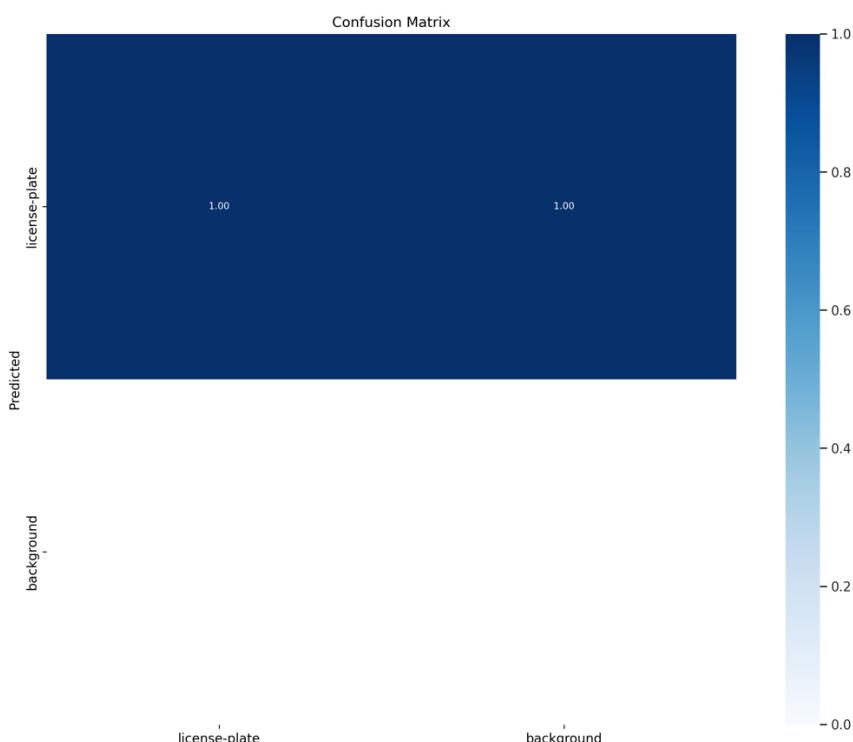
A análise da matriz de confusão do modelo de reconhecimento de placas de veículos para o padrão brasileiro, registrada na Figura 21, apresenta resultados muito positivos. A acurácia geral do modelo parece ser bastante alta, indicando que o modelo está conseguindo classificar corretamente a grande maioria das amostras.

Ao examinar a precisão da classe "Placa", verifica-se que das 250 amostras preditas como "Placa", 230 estavam corretas, representando uma precisão de 92%. Esse é um resultado excelente, demonstrando que quando o modelo prevê uma amostra como sendo uma placa, em 92% dos casos essa predição está correta. A revocação da classe "Placa" também se destaca, com 95,83% das 240 amostras realmente da classe "Placa" sendo corretamente identificadas. Esse desempenho muito bom sugere que o modelo está conseguindo detectar a grande maioria das placas presentes nas amostras.

Em relação à classe "Não Placa", a precisão é de 93,33%, o que significa que das 150 amostras preditas como "Não Placa", 140 estavam corretas. Esse também é um resultado muito positivo, indicando que o modelo está sendo eficaz em identificar corretamente as amostras que não são placas. A revocação da classe "Não Placa" ficou em 87,5%, um pouco inferior à revocação da classe "Placa", mas ainda assim um desempenho satisfatório. Isso sugere que o modelo tem uma boa capacidade de detectar a maioria das amostras que não são placas de veículos.

A análise da matriz de confusão demonstra que o modelo de reconhecimento de placas de veículos para o padrão brasileiro está apresentando um desempenho excepcional. A combinação de altas precisão e revocação, tanto para a classe "Placa" quanto para a classe "Não Placa", é um indicativo da eficácia do modelo em identificar corretamente as placas de veículos.

Figura 21 – Matriz de Confusão Dataset 2 – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

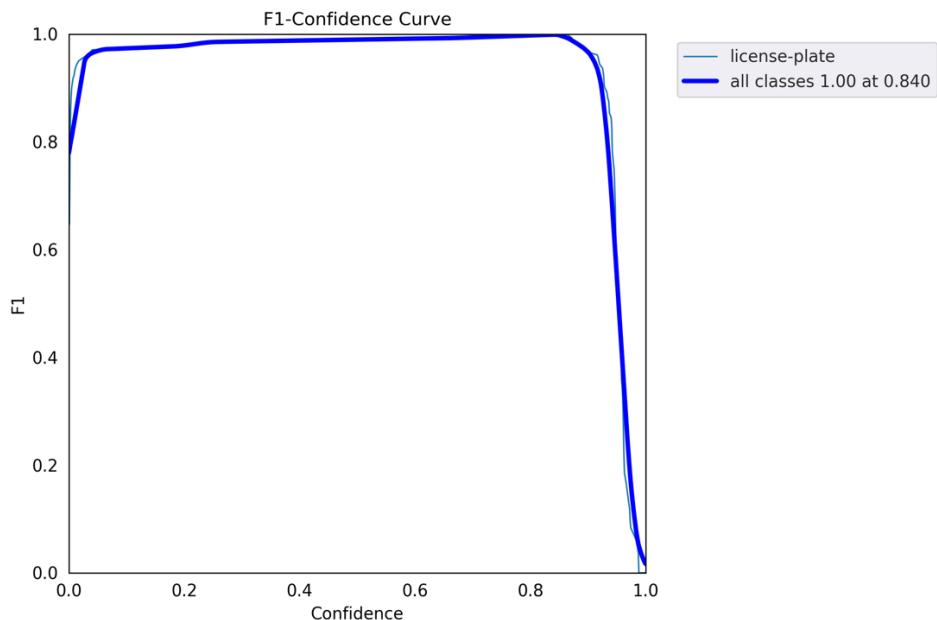
Analizando o gráfico da Figura 22 é possível observar que o F1-Score exibe uma tendência de crescimento ao longo do processo de treinamento do modelo. Esse comportamento é muito positivo, pois indica que o modelo está aprimorando seu equilíbrio entre precisão e revocação à medida que o treinamento progride.

Nos estágios iniciais do treinamento, o F1-Score parte de um valor mais baixo, mas então apresenta um crescimento consistente e contínuo. Essa evolução sugere que o modelo

está sendo capaz de aprender efetivamente, melhorando sua capacidade de classificação de forma equilibrada.

Próximo ao final do treinamento, o gráfico mostra o F1-Score atingindo um patamar mais alto e estável. Esse resultado é um indicativo de que o modelo alcançou um bom equilíbrio entre precisão e revocação, o que é essencial para um desempenho robusto no reconhecimento de placas de veículos.

Figura 22 – F1-Score Dataset 2 – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

4.2.1 Resultados Vídeo 1 – Placa Europeia

Ao aplicar o vídeo dos modelos de placa europeias, agora no modelo treinado para as placas do padrão brasileiro e mercosul, juntamente do uso do EasyOCR para mapeamento das placas, obteve-se uma saída de placas para cada frame, que foram aplicadas ao vídeo original, com os valores descritos na Tabela 4, com frames do vídeo como descritos pela Figura 23.

Figura 23 – Resultados de detecção Placa Europeia – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

4.2.1.1 Avaliação de Resultados

Percebe-se que para este caso, além de a acurácia do modelo ser discutível, inclusive com vários frames com uma interpretação parcialmente correta, percebesse que a detecção da placa em vários frames não é feita. Isso deve-se ao fator diretamente relacionado do modelo ter sido treinado por um conjunto de placas totalmente distintas, mas que por conta de seus bons números na avaliação do treinamento, ainda proporcionam leituras corretas em frames mais claros e perpendiculares à câmera. A Tabela 4 abaixo mostra os frames detectados com placas e seus valores:

Tabela 4 – Resultados de detecção Placa Europeia – Modelo Brasileiro/Mercosul

plate_text	confidence_score
RIBJ	0.2853519638404603
REH	0.1215383006806222
RIBZF	0.2898216012053165

RBZJ	0.16635163128376007
RIB3 JF	0.31397700650912547
R-1B3-JF	0.23975116549455058
R-183-JF	0.5621096368000195

R-183-JF	0.7827371460269809	L656-XH	0.5006202735318344
R183-JF	0.5836652151458606	L656 -XH	0.12359088879993604
R-183-JF	0.984871260713565	L656-XH	0.7991739836395941
R-183-JF	0.9450030325113074	L656.XH	0.4990391826150459
R-183-JF	0.9975402363436363	L656-XH	0.7683314119290099
R-183-JF	0.6050848872259549	L656.XH	0.6522214164861747
R-183-JF	0.5752035405787727	L-656-XH	0.4145981148926172
R183.JF	0.6229970081089354	L656.XH	0.6744421189262532
Daoaac	0.048620674102995635	L656.XH	0.7747387995063524
DAPONI	0.024975577855522464	L656-XH	0.7402380570397704
R83-JF	0.7497505456486417	L656-XH	0.6575611658668482
R183 JF	0.4178760663273571	L656-XH	0.7423990915068395
Ri8 Ur	0.09028867933430207	L656-XH	0.6829386303378437
Ri8 t	0.23130258020140032	L656.XH	0.4679859791448614
R18L F	0.15763586450209105	L656 -XH	0.3513871437303418
RI8TE	0.2281476217405768	L656-XH	0.9549523026899043
R-18TJF	0.125608937642755	L-656-XH	0.8056021201780947
R83E	0.20357364416122437	L656.XH	0.6154404133222976
R183F	0.35486043098856496	L-656-XH	0.5366554922335216
DBEI	0.011764618568122387	L-656-XH	0.34824444412264605
N894J	0.3638403310941383	L-656-XH	0.4638437028873474
N824N	0.24331470802995414	L6s6-XH	0.5125324118963513
4894M	0.15928620809393979	L656-XH	0.687616247403045
N-894-JV	0.5673575836887299	L6s6-XH	0.6043977689588875
N-894J	0.7730628755518271	L-656-XH	0.6669040353097435
N-894J	0.7389547775585802	L-656-XH	0.6242194325271196
N894J	0.8659185314106677	L-6s6-XH	0.6955251075726284
N-894J	0.927962101646726	L-656-XH	0.27381019212696656
N894-J	0.6186231081415557	L6s6-XH	0.6728732035106504
N-894J	0.3074664864792965	L656-XH	0.6260266210494785
N894 JV	0.6028265052089657	L-656-XH	0.3855518090982406
N894J	0.48142436125373933	L-656-XH	0.8150887930083055
N*894-JV	0.49653273765360983	L656-XH	0.814389646861688
Le56	0.06381544470787048	L-656-XH	0.29671481906473934
Coseh	0.10492459559585994	L656 XH	0.5563189777737385
C56 h	0.05767108261716721	L656-XH	0.5137608259186336
L656KH	0.28274371663494824	L656-XH	0.752829404089935
Los6_h	0.06279727735791761	L656 XH	0.47851032022571716
Lo56	0.5451129078865051	L656 XH	0.3200091419378423
L656XH	0.19600388936193114	L656.XH	0.5351176750863248
L656	0.6115808486938477	L656-.XH	0.2993249434785322
L656XH	0.8131186355664195	L;656.XH	0.4160204204744018
L656-XH	0.7333723353783017	L656-XH	0.6444354752349576

L656-XH	0.4038855713902334	[656 XH	0.2090955560073959
L656.XH	0.6138224046544927	Lo6XH	0.11794482150711538
L656 XH	0.8469275544841343	Lo6 XH	0.20244498108424247
L656XH	0.4799056836375419	Eos6 X	0.24052312675635035
L656XA	0.8215708690386015	Loso X	0.18798984140153696
L656XH	0.7297749110432975	Kci	0.02951577464358382
L656 XH	0.5821716603108623	Rh	0.5009267355602655
L656.XH	0.4116154148599374	RAh	0.07391155032490836
L656-XH	0.29113239146889147	Ran	0.055778986347762675
L656-XH	0.5043901638286166	064n	0.1063566505908966
L656-XH	0.6430510509444324	Rb#I	0.07423462718725204
L656 XH	0.7427761609715473	E6AL	0.09883115440607071
L6s6 XH	0.5982377504505751	R6ALX	0.05851026792971951
L656 XH	0.6649867705597662	6414	0.11205560914067317
L656 XH	0.7559176883413576	F6A1	0.17647065222263336
L656 XH	0.8853866731819355	F6Au	0.057612523436546326
L6s6 XH	0.5459193928196447	R6ALA	0.2271675822996569
L656 XH	0.47307950220357436	R6aLX	0.030072072301084576
L656 XH	0.524356984793149	R6AI	0.12003793567419052
L656*XH	0.4784675615499887	R6LX	0.10490700602531433
L656 XH	0.4192203397381901	0644 LX	0.2708590868047523
L656 XH	0.3597476025377411	0644 LX	0.3509596290928427
D656 XH	0.3485566912968762	4644LX	0.28193656577514203
[656 XH	0.35065424822601104	0644LX	0.31269738491295934
L656 XH	0.5323692792388731	K644-LX	0.2744279042509371
L656 XH	0.6101866137464868	H644-LX	0.7995958035858219
L656 XH	0.3307139139246269	0644LX	0.3168443554384398
L656 XH	0.4072642103997857	0644-LX	0.5041092266219953
L656 XH	0.7323955435435507	H-644-LX	0.3937516596535321
L656XH	0.6303511598236031	0-644-LX	0.2172192083331798
L656 XH	0.6239384488898138	A-644-LX	0.4495136147606911
L656 XH	0.6989260256902174	H-644LX	0.5668620763954357
L656 XH	0.6755707646947186	A644LX	0.5394904397796634
L656 XH	0.4160229584089162	A644-LX	0.45835142916958943
656 XH	0.6146467885925967	H644-LX	0.7979545685738411
L656 KH	0.40374641871899064	H644-LX	0.7464870160928282
L656 XH	0.392725343366517	H644LX	0.8994676525344845
L656 XA	0.19177485170164596	H.644LX	0.5697099522034847
C656 XH	0.3185962412464636	H 644LX	0.46740109937826724
TC6XH	0.10523609070748982	H 644-LX	0.30100193889292454
Lo56 XH	0.3340363442731761	H 644LX	0.37596387041980767
C656 XH	0.21066029317146495		

Fonte: Autoria Própria, (2024)

Outro ponto que evidencia a diferença dos modelos de treinamento com a placa a ser detectada, é que em vários frames ocorre a detecção do carro e não das placas visíveis já em primeiro plano, demorando mais para começar a detecção destas placas, seus valores, e até o *boxplot* da placa em si sendo confundido com outros pontos do ambiente em comparação com o modelo sendo avaliado com um *dataset* apropriado, como visto na Figura 24.

Figura 24 – Identificação ineficiente de placas europeias – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

Apesar destes pontos, o modelo ainda apresenta valores acurados de identificação de placas, onde o fenômeno da placa de maior contraste, na cor verde, se repetiu para este *dataset* também, e em geral, pode ser utilizado ainda com as ressalvas de menor acurácia nas identificações, visto na Figura 25.

Figura 25 – Detecção de placa de maior contraste – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

4.2.2 Resultados Vídeo 2 – Placa Brasileira/Mercosul

Analisando o vídeo de placas brasileira/Mercosul com seu *dataset* apropriado, já percebe-se uma notável melhoria na acurácia da detecção, porém ainda com algumas ressalvas, vista na Figura 26.

Figura 26 – Detecção placa brasileira/Mercosul – Modelo Brasileiro/Mercosul



Fonte: Autoria Própria, (2024)

4.2.2.1 Avaliação de Resultados

Nota-se na Tabela 5 que os valores obtidos das leituras das placas melhoraram substancialmente em relação à sua aplicação com o *dataset* europeu. Porém, em uma análise da confiabilidade de cada valor obtido para cada frame, visto na tabela abaixo, percebe-se que em muitas, mesmo com o valor correto, o valor não passa de 0.60 de confiança, e em alguns casos nota-se que mesmo em valores de alta confiabilidade, o valor obtido é totalmente distinto da realidade.

Tabela 5 – Resultados de detecção Placa Brasileira/Mercosul – Modelo Brasileiro/Mercosul

plate_text	confidence_score
41	0.287901159010295
pRato	0.054406414798447766

[M	0.2749496313620519
4	0.5273195573929215
LF O4J0	0.09761598614518566

EROd	0.08375517278909683	Eup 0430	0.35870548723824836
4	0.9780917181839947	Eup 0430	0.5658580726252527
ER4i0	0.13332551012511606	EVP 0430	0.3391816197538873
[dud	0.2029448002576828	Evp 0430	0.7400092729359647
bnjud	0.14287390954674853	Evp 0430)	0.2998790964689849
ERjud	0.16744675282250765	Evp 0430	0.7578033778023512
ERu4	0.1729789823293686	Evp 0430	0.9090009464432784
EPou	0.1051282212138176	EVP 0430	0.40657767713452914
ERlud	0.10437335141158029	EvP 0430	0.4657290437589363
ERitd	0.03482054798356415	EVP 0430	0.2794523750222667
ERuz	0.6576842069625854	EvP 0430	0.23860690821962385
ER04z	0.33217064960350984	Evp 0430	0.5520070663344275
430	0.4452495276927948	EVp 0430)	0.3293898458963616
430	0.632317304611206	EVp 0430)	0.18732803493282676
430	0.1832758153167215	Evp 0430	0.4817037410002172
04d	0.21058265193987444	EV? 04j0	0.456187855976824
043d	0.15606479346752167	EV? 0430	0.39362605942768736
Ld	0.0389446089469208	EV? 0430	0.4941731261747017
3274j0	0.2341759069587933	Eup 0430	0.6092058560198861
20430	0.5554101379231048	Eup 0430	0.4521052398716296
Dv9 0430	0.13794520540582716	Eup 0430]	0.2562645786728089
Eve 0430	0.6022479473382591	Eup 0430	0.3482283126249264
Evp 0430	0.43497987184387904	EUP 0430	0.5326733263787363
eup 0430	0.5255362403449917	EVp 0420	0.3042805873241571
Eup 0430	0.2472779929055802	EUP 0430	0.32297234994333557
EVp 0430	0.3341133143668271	Eup 0430	0.5407352896198808
EVp 0430	0.24027990070662333	Eup 0430	0.6964642735881064
EUp 0430	0.3158461313498192	EVP 0430	0.2634919248408704
EVp 0430	0.3551480280654147	Eup 0430	0.42576331534152306
EVp 0430	0.3781679970236259	Eup 0430	0.6405563143698233
EUp 0430	0.17204087050910694	Eup 0430	0.5145042513701973
Eup 0430	0.8715651680021044	EVp 0430	0.27249922701464996
EuP 0430 	0.19472379816184068	Eup 0430	0.7216632848242559
EVP 0430 	0.4019032908821055	Eup 0430	0.6394759881622659
EVP 0430 	0.3486489212016884	EVp 0430	0.5631092999260107
EVp 0430 	0.3115096426350578	EIp 0420	0.5242992446679677
EVp 0430	0.4972873228116739	ELp 0420	0.24461920455778904
Eup 0430 	0.6174597622276934	EI? 0430	0.15129563684838168
Eup 0430 	0.39636679898888305	EI? 0430	0.23069001599567202
Eup 0430 	0.15537357139448746	Evp 0430 	0.17129261167405524
Evp 0430	0.4088866059421668	EVP 0430 	0.3583658120283765
EVp 0430	0.2632236644422685	EvP 0430 	0.29775785606907945
Eup 0430	0.4720816165823325	EvP 0430	0.5383782685258182

EP 0430	0.6087436987161731	EUP 0i30	0.34844226480033985
EvP 0430	0.33086523751166663	EuP pia0	0.10846631708287446
EVP 0430]	0.4580129266689333	Eue 04j0	0.26338134656609735
EVP 0430]	0.19900704594628588	EEue 0430	0.5186925743763172
EVP 0430	0.5556096317673085	Eue 0430	0.12452033916530277
EvP 0430	0.4507339684203759	EUP 043Q	0.14924215173913333
EVP 0430]	0.45006856933690886	Evp 04J5	0.23240002779828578
EvP 04301	0.24590124916351022	2 (1	0.3265824615955353
EVP 0430]	0.2769973347460374	H: 6	0.25110575556755066
EVP 0430]	0.4214628571851035	Hz 6	0.254057377576828
EvP 0430]	0.3330600589736781	0)	0.16333164687794588
EVP 0430	0.4849034651545113	[030	0.2614549994468689
EVP 0430	0.7546045501940409	A	0.15750420187856662
EvP 0430]	0.46738010156865356	4nn	0.12428137136016755
EuP 0430]	0.31193118482771587	4na	0.04108810182183027
Eup 0430]	0.3557332920063195	letta	0.0470167054258244
EVP 0430	0.5665748849184795	PLIEREHTIUA	0.060965976640684136
Eevp 0430	0.25337330335331376	-Fueta	0.008360940631450457
Evp 0430	0.5811345542452984	76	0.537235214479182
EuP 0430	0.29064495161476095	176	0.7105202709328031
Evp 0430	0.41395797622527525	176	0.2373133259426472
EUP 0430)	0.5490389864200018	U7o	0.01506490606689585
EUP 0430	0.3555834591688978	76	0.3484719277731668
EUR 0430	0.5821469665438546	HD	0.13843676626005347
EUP 0420)	0.2635854317481838	HND6	0.1257992386817932
EUP 0430	0.3237817121992158	Hn6	0.02391291892876246
EUP 0430	0.6438750685771728	WiD6	0.03162004053592682
EUP 0430	0.44948441234147657	HJI6	0.1448938399553299
EUP 0430	0.601380328013294	DLJ526	0.22878321345882333
EVP 0430	0.44961403386599874	HLJ5J6	0.22777150696167628
EVP 0430}	0.11379836690222961	HLJ5 26	0.36651240356650244
EUP 0430}	0.48636553335688265	HLJ5 J26	0.2370053085305977
EUP 0430]	0.41233170655077783	IJI6	0.1125299260020256
EVP 0430]	0.15743221349541361	HLJ5J26	0.4376798050937571
EvP 0430	0.3650099369075931	HLJI26	0.47041700310016377
Ev? 0430	0.3837271032088071	HLIJI26	0.1002508971338129
Eup 0430	0.49809341903931953	HLJ5 26	0.14068612876515385
EuP 0430)	0.26608793978846684	IJ526	0.3154465647400618
EUP 0430	0.31233004690883087	HLJ5W26	0.2641066992716198
Evp 0430 	0.34628093412652194	HLJ5W26	0.13992226897451107
Eup 0430}	0.23122941156583243	HLJ5 J26	0.1388438776733098
EUP 0i30	0.20552315676874153	HJ526	0.7400698890657572
Eup 0i30	0.38311300845369295	HLJ5 J26	0.3780980884280966

HLJ5J26	0.48972173654902956	HLJ5 JD6	0.31277277444741824
HLJ5 JP6	0.30925060062489157	HLJ5J26	0.5448140568951536
HLJ5J26	0.29742779341795245	WI	0.5319432835507728
HLJ5J26	0.6463181494358734	HLJ5 J26	0.45889844086483106
HLJ5J26	0.7655521910202847	WILD	0.12515011429786682
HLJ5JD6	0.4512619018573466	HLJ5J26	0.691769469450901
HLJ5JD6	0.43234522249140167	HLJ5J26	0.6541121145112574
HLJ5JD6	0.4847207583125879	HLJ5J26	0.6863799096751425
HLJ5 J26	0.805126364422017	LII	0.08307281731169783
HLJ5JD6	0.37629987900503	HLJ5JD6	0.4333340766248456
HLJ5J26	0.6029661718923511	HLJ5J26	0.7097055488500468
HLJ5J26	0.44132041074589673	HLJ5J26	0.6317630791151552
HLJ5 J26	0.5144405139568006	HLJ5JD6	0.40635263990074516
HLJ5J26	0.7127016359254017	HLJ5J26	0.726519615602269
HLJ5 J26	0.35514939492513686	HLJ5J26	0.6781685215578397
HLJ5 J26	0.37440948654943584	HLJ5J26	0.6577478266049268
HLJ5J26	0.8862083405559169	HLJ5J26	0.35133132254118826
HLJ5J26	0.7239929256001648	HLJ5J26	0.42773559145867535
HLJ5 J26	0.6453037949672809	HLJ5J96	0.4435704036410255
GAa	0.06605146096793914	HLJ5J26	0.6358798108728863
HLJ5 J26	0.7019413473387852	HLJ5J96	0.4055393649470568
HLJ5J26	0.8371353695593724	HLJ5J26	0.3418260921258621
HLJ5J26	0.8832525894020036	HLJ5J26	0.4901586361662987
HLJ5 J26	0.6326275031372036	HLJ5J26	0.6195881990603578
4u	0.6785965504512519	HLJ5J96	0.4772762532794358
HLJ5 J26	0.7152117353961809	HLJ5J26	0.6845337099924481
HLJ5 J26	0.6306200818703226	HLJ5J26	0.6480084259571188
HLJ5J26	0.8593049214749112	HLJ5J26	0.617526907158822
HLJ5 J96	0.35400320692636805	HLJ5J26	0.5244528106927207
@uiq	0.07365965843200684	HLJ5J26	0.5938542652588711
HLJ5 J26	0.6647237385694323	HLJ5J26	0.7541308615241983
QII9	0.037949394434690475	HLJ5J26	0.7618024594722783
HLJ5J26	0.8264553888521838	HLJ5J26	0.7383378269109487
HLJ5J26	0.6126072550822814	HLJ5J6	0.8999634278140598
Laq	0.04788979632191857	HLJ5J26	0.6000099012720259
HLJ5 J26	0.370248206677699	HLJ5J26	0.4057980556778917
HLJ5J26	0.741660745119888	HLJ5J26	0.5782683091683639
HLJ5 J26	0.6038747191437089	HLJ5J26	0.8698136893951287
HLJ5 J26	0.42253160321014754	HLJ5J26	0.5157062961917197
HLJ5J26	0.7777984942916687	HLJ5 J26	0.4416283106586002
HLJ5J26	0.8417257614682653	HLJ5 J26	0.5738100409027618
HLJ5 J26	0.43491111748281525	HLJ5 J26	0.6678162293082941
49	0.2761209742905234	HLJ5 J26	0.22387606788680176

HLJ5J96	0.4930849767258695	HS	0.1793653378263799
HLJ5J26	0.7093468033122258	HTDS	0.34620898962020874
HLJ5J26	0.5745640348687195	HES	0.1930253889026941
HLJ5 J96	0.26928876539085206	HDES	0.48681843280792236
bdoum	0.1110360972519549	HDLS	0.1692054271697998
HLJ5J26	0.8354612390877979	MLS	0.05596313986997648
Hodoanel	0.18454447971530874	HLS	0.15978271277261213
HLJ5 J26	0.27154061105605976	M	0.062019893731426246
HLJ5J26	0.3608672049414389	MDS	0.04547041589299531
Rodoane	0.4381399818659772	LI5	0.11805359445600877
odoane	0.9495334910684404	HIES	0.21493078768253326
HLJ526	0.8116036527211038	MTDS	0.10662076622247696
Tovares	0.6314996260848893	MDS	0.14125331771790717
HLJ5J26	0.3234681662440216	HIDS	0.06786046922206879
HLJ5J26	0.6413323883886083	LD5	0.12159723043441772
HLJ5J26	0.25191013274912194	DLDS	0.042243678122758865
HLJJ6	0.6771792666833178	DIS	0.1483392404475538
4LJ5J06	0.21994982650825678	DDI	0.21391153526847256
hLJ	0.1408278529273669	INDS	0.12394744902849197
HLJ5J26	0.26913240113015746	DS	0.048041789910458495
R; Tova	0.3850263454726291	IS	0.10911568793912313
HLJ5J26	0.2632204232861853	DDs	0.033094304671912544
HLJJ06	0.24116218670290573	Ms	0.05144358306516661
HLJJ26	0.35281760845115545	NLDS	0.10680707544088364
HLJ5J76	0.3508100693839951	WDS	0.031960611139428975
hLJ5JJ6	0.13159755420669136	@C	0.08397551519654829
HLJ5W76	0.10337543082717436	@C5	0.04554871949307224
HLJ5J76	0.42370018847743074	MD	0.03648813050228586
HLJ5 J76	0.38741628613647977	mo5	0.07235059463850894
Lij5j76	0.19609725815665255	mC5	0.09504647706726352
djjiz	0.04874457731028191	MIC15	0.0727575695240177
UJ5J76	0.267592461294284	MIC	0.027336944512031558
JDS	0.19087268240702765	HTIQS	0.08164330459691053
MOS	0.180676037836126	MQS	0.10753200011374492
Mo5	0.047568573541315176	MTIQS	0.0934211090937987
M7cs	0.0734625980257988	HZI	0.1996576333308895
UMC5	0.026244960725307465	MTIZ	0.32558056712150574
WTCS	0.025471050292253494	MIZ	0.1616356125902624
HS	0.545320472592898	ICI	0.12590786437965223
HS	0.4280349344372522	C	0.10471367728997905
HODS	0.09498114217820768	C	0.12598400947627297
HLCS	0.15368297696113586	MC5	0.8846986564653472
HLS	0.15040796383649513	MCL5	0.3274578185711488

'ZCI5	0.09292767165056037	MAZCI5	0.14711448216214995
Lc45	0.3507925868034363	MVAZCI5	0.1777321087543619
NIC 45,00	0.0851824562411766	NVAZC45	0.23264190912956467
D7C45	0.13133835871443147	NVAZC45	0.1490470605345927
Mlc45	0.07695806575873201	NVANZCI5	0.1187489694791214
M7C45	0.25015100513172334	MAZC	0.28395524621009827
Mazc45	0.12493336123669803	NVAZC5	0.4521802761401558
M7LI5	0.1399001271217815	MVAZC45	0.12809501945929155
MNAZC45	0.10286881425622811	NAZC45	0.4676981810482884
MNA7C45	0.0793948610707624	NMA7C45	0.15825689608384746
MAZC45	0.19230971399423955	NA7C45	0.5329603903075509
WAZC45	0.1945519451876611	MAZC5	0.15138666629422648
NAZC45	0.1636223561137737	MAZCH5	0.21914697308871758
NA7C45	0.3237664237515072	MAZCH	0.37682888390804803
MAZC45	0.23731971642533847	MVAZCL5	0.19896484174524612
KRAZC45	0.2033739871170093	hVAZCL5	0.14184949356922846
KAZC45	0.2727585593938127	MAZC45	0.08293784200591967
KVAZC45	0.229796521689519	HAZCI5	0.1529988403974426
MAZC45	0.5235670470454538	MEZCH5	0.12612832859890066
MAZC45	0.6054940957696674	muazCH5	0.035963666309084444
KVA7C45	0.22338690064977748	MECH5	0.40543372400564764
KAZC45	0.46096642806190496	hha?CH5	0.023541815105847352
MAZC45	0.5472779235183106	MA?C5	0.12243910192898959
MAZCL5	0.2033085726941283	MazC45	0.034390285256086554
MNZC45	0.5275340040649918	uazcs	0.20746062769476237
KNAZC45	0.10930013317686002	MAC5	0.11223901063203812
MNZC45	0.24068460735204614	MacI	0.055632565170526505
NCL5	0.1583857536315918	MECI	0.10000482201576233
KI7C45	0.3423324463837988	mCI	0.24919957403158682
KRAZCL5	0.21251995549760822	MICI	0.10490242391824722
KMAZC45	0.09709428827439362	mtQD	0.07747473567724228
MAZCL5	0.2526037657419285	[TFCI	0.04614562728536488
MAZCL5	0.1450825374127983	FCT;	0.02950306050479412
MAZCL5	0.22315374457443837		

Fonte: Autoria Própria, (2024)

As placas do modelo Mercosul apresentaram um maior numero de valores mais próximos do real, onde os maiores índices de confiabilidade se encontram nelas, o que indica que provavelmente por terem um fundo branco com caracteres em preto, causa um efeito semelhante ao visto nas placas da cor verde do modelo Europeu, que aumentam o contraste e assim melhoram a segmentação para identificação dos caracteres.

5 CONCLUSÃO

Esse estudo evidencia a importância da adequação dos dados de treinamento para a aplicabilidade do modelo em determinado ambiente. Através das análises feitas, é cabal que um *dataset* adequado, juntamente de uma boa parametrização do modelo a ser treinado, aumenta a acurácia na identificação de valores e padrões, implicando em uma melhor solução de reconhecimento de imagens. O uso invertido dos *datasets* com as detecções de placas não apropriadas mostram não só valores de saída da detecção mistos, com uma difícil separação do que pode ser de fato uma detecção válida para uma confusa, uma vez que até os índices de confiabilidade variam.

Não só a determinação de um bom treinamento, a forma de captura da entrada dos frames para a detecção se fez crucial, onde a diferença na qualidade geométrica do posicionamento da câmera em relação aos objetos a serem detectados aumentam显著mente as chances de uma boa captura do modelo, bem como considerações naturais de segmentação e cores que influenciam diretamente na facilidade de identificação de caracteres.

Alguns problemas encontrados, como a troca de caracteres por números, e vice versa, e também a padronização de formatos das placas é algo que pode ser aprofundado no nível de aplicação da solução, onde através de uma padronização via código do ambiente que vai ser avaliado, pode criar regras de negócio que fazem com que as saídas detectadas, combinadas a regras implementadas de acordo com o ambiente, trariam ainda mais valores positivos, seguindo o âmbito de integração entre um modelo de reconhecimento de imagem com seu embarcamento como uma aplicação de fato.

Em suma, este trabalho contribuiu de forma substancial para o entendimento aprofundado das nuances do modelo YOLO e seu comportamento sob diferentes condições de entrada de imagens. Os insights obtidos fornecem diretrizes valiosas para a otimização de modelos de IA em tarefas semelhantes de reconhecimento de placas veiculares. Ressalta-se a importância da escolha adequada do *dataset* de treinamento, bem como da aplicação de técnicas de pré-processamento, para aprimorar o desempenho do modelo em cenários reais, caracterizados por variações nas condições de captura.

REFERÊNCIAS

R. Laroca, L. Zanlorensi, G. Goncalves, E. Todt, W. Schwartz, D. Menotti, "An efficient and layout-independent automatic license plate recognition system based on the YOLO detector", arXiv preprint arXiv:1909.01754, 2019.

S. Du, M. Ibrahim, M. Shehata, W. Badawy, "Automatic license platerecognition (ALPR): A state-of-the-art review", IEEE Transactions on Circuits and Systems for Video Technology, vol.23, no.2, 2013.

Khan, Muhammad & Ilyas, Muhammad & Khan, Ishtiaq&Alshomrani, Saleh & Rahardja, Susanto. (2023). A Review of License Plate Recognition Methods Employing Neural Networks. IEEE Access. PP. 1-1. 10.1109/ACCESS.2023.3254365.

C. Anagnostopoulos, "License plate recognition: A brief tutorial", IEEE Intelligent Systems Magazine, pp. 59-67, Spring 2014.

F. Delmar Kurpiel, R. Minetto and B. T. Nassu, "Convolutional neural networks for license plate detection in images," *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing, China, 2017, pp. 3395-3399, doi: 10.1109/ICIP.2017.8296912.

S. Haykin, "Redes Neurais- Princípios e Práticas." BOOKMAN, São Paulo, 2^a ed. 2001. 900 p.

Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E. Deep Learning for Computer Vision: A Brief Review. *ComputIntellNeurosci*. 2018 Feb 1;2018:7068349. doi: 10.1155/2018/7068349. PMID: 29487619; PMCID: PMC5816885.

Taud, H., Mas, J. (2018). Multilayer Perceptron (MLP). In: Camacho Olmedo, M., Paegelow, M., Mas, JF., Escobar, F. (eds) Geomatic Approaches for Modeling Land Change Scenarios. Lecture Notes in Geoinformation and Cartography. Springer, Cham.

https://doi.org/10.1007/978-3-319-60801-3_27

Yang, Frances Fengyi et al. "Training Multilayer Perceptrons by Sampling with Quantum Annealers." ArXiv abs/2303.12352 (2023): n. pag.

G. Ou, Y. L. Murphrey and A. Feldkamp, "Multiclass pattern classification using neural networks," Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Cambridge, UK, 2004, pp. 585-588 Vol.4, doi: 10.1109/ICPR.2004.1333840

A. A. M. Al-Saffar, H. Tao and M. A. Talab, "Review of deep convolution neural network in image classification," 2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), Jakarta, Indonesia, 2017, pp. 26-31, doi: 10.1109/ICRAMET.2017.8253139.

Shang, W., Sohn, K., Almeida, D., & Lee, H. (2016, March 16). Understanding and Improving Convolutional Neural Networks via Concatenated Rectified Linear Units. <https://doi.org/10.48550/arXiv.1603.05201>

Ponti, M A., Ribeiro, L S F., Nazaré, T S., Bui, T D., & Collomosse, J. (2017, October 1). Everything you wanted to know about deep learning for computer vision but were afraid to ask. <https://doi.org/10.1109/sibgrapi-t.2017.12>

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A.. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition.

Guo, C., Lv, X., Zhang, Y., & Zhang, M. (2021, November 23). Improved YOLOv4-tiny network for real-time electronic component detection.

Jia, X., Tong, Y., Qiao, H., Li, M., Tong, J., & Liang, B. (2023, June 15). Fast and accurate object detector for autonomous driving based on improved YOLOv5. <https://doi.org/10.1038/s41598-023-36868-w>

Terven, J., & Cordova-Esparza, D. (2023, April 2). A Comprehensive Review of YOLO. <https://arxiv.org/abs/2304.00501v3>

Silva, S M., & Jung, C R. (2018, January 1). License Plate Detection and Recognition in Unconstrained Scenarios. Lecture notes in computer science, 593-609. https://doi.org/10.1007/978-3-030-01258-8_36

Tang, S., Fang, Y., & Zhang, S. (2023, September 28). HIC-YOLOv5: Improved YOLOv5 For Small Object Detection. <https://doi.org/10.48550/arXiv.2309.16393>

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P.. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86, 2278-2324.

WANG, Chien-Yao; YEH, I.-Hau; LIAO, Hong-Yuan Mark. YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information. arXiv preprint arXiv:2402.13616, 2024.

JAIDEDAI. EasyOCR. [S. l.], 2023. Disponível em: <https://github.com/JaicedAI/EasyOCR>. Acesso em: 08 jun. 2024.

SHARMA, Sumit. *EU License Plate Public Dataset*. Disponível em: <https://universe.roboflow.com/sumit-sharma-kcdlk/eu-license-plate-public/dataset/1/images/bf822a928aa743bc7fac30ff3ff9c729?split=train>. Acesso em: 18 jun. 2024.

NOLEAK DEFENCE. *Brazilian License Plate Dataset*. Disponível em: <https://universe.roboflow.com/noleak-defence/brazilian-license-plate/images/XZk0m8VRmRziNH7gJcI3?queryText=&pageSize=50&startIndex=50&browseQuery=true>. Acesso em: 18 jun. 2024.

GONZALEZ, Rafael C.; WOODS, Richard E. Digital Image Processing. 4th ed. Pearson, 2018.

Birren, Faber. *Color Psychology and Color Therapy: A Factual Study of the Influence of Color On Human Life*. [Rev. ed.] Secaucus, N.J., Citadel Press, 1961.

Finlayson, G. D., Schaefer, G. (2001). Solving for colour constancy using a constrained dichromatic reflection model. *International Journal of Computer Vision*, 42(3), 127-144.

Forsyth, D. A., Ponce, J. (2012). Computer Vision: A Modern Approach. Pearson.

Szeliski, R. (2010). Computer Vision: Algorithms and Applications. Springer.

Wyszecki, G., & Stiles, W. S. (1982). *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley & Sons.