



Carlos Ivan Santos do Amaral

Avaliação de plataformas para o reconhecimento de placas veiculares brasileiras

Recife

Dezembro de 2021

Carlos Ivan Santos do Amaral

Avaliação de plataformas para o reconhecimento de placas veiculares brasileiras

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Cícero Garrozi

Recife

Dezembro de 2021

CARLOS IVAN SANTOS DO AMARAL

Avaliação de plataformas para o reconhecimento de placas veiculares brasileiras

Artigo apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovado em: 14 de Dezembro de 2021.

BANCA EXAMINADORA

Cícero Garrozi (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Filipe Cordeiro Rolim
Departamento de Computação
Universidade Federal Rural de Pernambuco

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- A485a Amaral, Carlos Ivan Santos do
Avaliação de plataformas para o reconhecimento de placas veiculares brasileiras / Carlos Ivan Santos do Amaral. -
2021.
22 f. : il.
- Orientador: Cicero Garrozi.
Inclui referências.
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em
Sistemas da Informação, Recife, 2021.
1. Machine Learning. 2. Visão Computacional. 3. Yolo. 4. Placas Veiculares. I. Garrozi, Cicero, orient. II. Título

CDD 004

Avaliação de plataformas para o reconhecimento de placas veiculares brasileiras

Carlos I. S. Amaral¹, Cícero Garrozi (Orientador)²

¹Bacharelado em Sistemas de Informação - Universidade Federal Rural de Pernambuco

²Departamento de Estatística e Informática - Universidade Federal Rural de Pernambuco

Rua Dom Manuel de Medeiros, s/n, - CEP: 52171-900 – Recife – PE – Brasil

carlosivan880@gmail.com, cicero.garrozi@ufrpe.br

Resumo. Com a crescente quantidade de veículos particulares no Brasil se faz cada vez mais necessário melhores métodos de gestão e fiscalização da frota veicular. As placas veiculares (PV) são objetos únicos e obrigatórios com o objetivo de identificar o veículo assim como seu proprietário. É recomendável que a coleta eficiente das informações presentes nas placas veiculares sejam realizadas por meio de sistemas automatizados para a detecção e reconhecimento de PV. Estes sistemas são fundamentais para se realizar a fiscalização e a gestão de diferentes atividades relacionadas ao tráfego de veículos. Neste sentido, este trabalho apresenta um estudo que identifica métodos de detecção e reconhecimento de PV com algoritmos baseados em aprendizagem de máquina. Para a produção desse experimento, sucedeu a coleta de um banco de imagens de veículos em praças de pedágio que estão localizadas no município de Cabo de Santo Agostinho - PE e fazem acesso ao Complexo Industrial Portuário Governador Eraldo Gueiros - SUAPE. O objetivo desse trabalho foi prover uma comparação entre o serviço de visão computacional da Microsoft Azure para detecção de objetos PV em conjunto com os serviços de Reconhecimento Óptico de Caracteres (Optical Character Recognition - OCR) da Google Vision com o algoritmo de Aprendizagem Profunda YOLO v4. O resultado do experimento expôs que em condições semelhantes de configurações em ambos dos modelos estudados, o YOLO v4 apresentou melhor desempenho, obtendo uma taxa de 92% de precisão na detecção e reconhecimento de placas veiculares.

Abstract. With the growing number of private vehicles in Brazil, better methods for managing and inspecting the vehicle fleet is becoming increasingly necessary. License plates (LP) are unique and mandatory objects with the purpose of identifying the vehicle as well as its owner. It is recommended that the efficient collection of information on license plates be performed by automated systems for LP detection and recognition. These systems are fundamental for the supervision and management of different activities related to vehicle traffic. In this regard, this paper presents a study that identifies methods for LP detection and recognition with algorithms based on machine learning and deep learning. To produce this experiment, we succeeded in collecting an image bank of vehicles in toll plazas that are located in the municipality of Cabo de Santo Agostinho - PE and provide access to the Governador Eraldo Gueiros Port Industrial Complex - SUAPE. The objective of this work was to provide a comparison between Microsoft Azure's computer vision service for LP object detection in conjunction with Google Vision's Optical Character Recognition (OCR) services with the YOLO v4 Deep Learning algorithm. The result of the experiment showed that under similar configuration conditions in both models studied, YOLO v4 performed better, achieving a 92% precision rate in license plate detection and recognition.

1. Introdução

Segundo relatório do mapa de motorização individual no Brasil [3], entre os anos de 2008 e 2018 foram acrescentados à frota brasileira um total de 28,6 milhões de automóveis, elevando a quantidade de veículos no país para um montante de 65,7 milhões até o ano de 2018. Dados levantados pela Associação Nacional dos Detrans apontam que o país possui um automóvel para cada 4,4 habitantes [2]. Com esse aumento substancial à frota brasileira, as principais capitais registram altos índices de violação às leis de trânsito [4], o que torna cada vez mais necessário o aprimoramento nas fiscalizações das vias e uma gestão mais eficiente nas zonas que possuem acessos restritos. Um levantamento realizado em pedágios do Rio de Janeiro, mostram que a cada uma hora, um motorista evade do pedágio sem realizar o devido pagamento das taxas [7]. Sendo assim, um monitoramento automatizado de placas veiculares pode ser um método eficiente na detecção dessas placas, dentre outras aplicações [26]. Para circular livremente no Brasil, os veículos precisam ter a placa veicular (PV), que é única e obrigatória, regulamentada e licenciada por um departamento estadual de trânsito. As PV representam dados importantes que descrevem o veículo, seu proprietário, marca, cor, ano de lançamento, modelo entre outras informações. Até a presente data da produção desse trabalho o Brasil possui 2 modelos vigentes de placas veiculares, que é o modelo definido na década de 1990 e é constituído por um padrão de 3 letras seguidos de 4 números que estão com seu tamanho destacado, além do município e estado na parte superior da placa conforme a Figura 1. O segundo modelo são as placas veiculares com padrão Mercosul que estão em fase de transição para substituir o modelo citado anteriormente, onde sua criação ocorreu em 2014, mas sendo adotado em todo território nacional no ano de 2020. O modelo de placas do Mercosul (Figura 2) permite que países que o adotem como Brasil, Argentina, Paraguai e Uruguai tenham acesso a um banco de dados integrado nas quais esses modelos de placas podem ser acessados por órgãos de trânsito. Com esse crescente aumento na frota veicular brasileira, se faz cada vez mais necessário o uso de sistemas que realizem a fiscalização e a coleta de informações das placas veiculares e dentro desse contexto os sistemas de reconhecimento automático de placas veiculares (*Automatic License Plate Recognition - ALPR*) são os mais recomendados.

Na literatura é possível encontrar diversos métodos de processamento de imagem que realizam os ALPR, tais como técnicas baseadas nas bordas das placas veiculares [13], métodos baseados na coloração das PV [25] e métodos usando aprendizado profundo [19], dentre outros. Os sistemas baseados em ALPR são considerados desafiadores devido às mudanças ambientais e climáticas que influenciam diretamente os resultados das aplicações de tais sistemas, como ruídos na imagem, pouca iluminação, veículos em movimento, que são alguns exemplos de problemas que podem prejudicar a localização e identificação da placa veicular. Logo, propor um método geral que se aplique a todos os cenários é relativamente difícil.



Figura 1. Placas veiculares em uso desde 1990 [5].



Figura 2. Placas veiculares no padrão Mercosul [14].

1.1. Contexto

O problema de pesquisa deste trabalho baseia-se em como localizar e identificar placas veiculares nos modelos brasileiro, através de imagens de veículos em ambientes simples. No cenário de ALPR, os ambientes são ditos como simples quando a base de imagens coletadas seguem um padrão de cenários repetidos, conforme a Figura 3. No caso deste estudo, 1325 imagens foram coletadas em praças de pedágio, localizadas na cidade de Cabo de Santo Agostinho - PE, para compor o dataset de treinamento e testes neste estudo. Já um ambiente é dito como complexo, quando existe uma grande variação de cenários que compõem a base de imagens, conforme a Figura 4.



Figura 3. Ambientes Simples [8].



Figura 4. Ambientes complexos (Fonte: Imagens da internet).

Os estudos mais recentes abordam as técnicas que utilizam algoritmos baseados em aprendizagem profunda para realizar a detecção e o reconhecimento de placas veiculares como [19] e [28] que apresentam métodos de localização de PV usando rede neural convolucional (CNN). Nos últimos anos, as arquiteturas baseadas em *You only look once* (YOLO) inspiram muitos estudos no processo de detecção e reconhecimento de placas como [11] que utilizam a versão 2 do YOLO para construir um sistema de detecção e reconhecimento de placas de veículos robusto e eficiente.

A detecção e reconhecimento de placas veiculares através de imagens torna-se uma tarefa desafiadora visto que os registros fotográficos que compõem o banco de

imagens foram manualmente capturadas através de smartphones e isso influencia no ângulo de perspectiva da cena, além disso com a dinâmica acelerada das praças de pedágio, algumas imagens foram capturadas com os veículos em movimento. Neste trabalho foi utilizado o serviço de visão computacional personalizada da *Microsoft Azure - Custom Vision AI*, que utiliza um algoritmo de aprendizado de máquina para analisar imagens, com o objetivo de detectar os objetos placas veiculares após seu devido treinamento. Para dar continuidade ao processo de identificação, foi utilizado na etapa seguinte o serviço de reconhecimento óptico de caracteres (OCR), fornecido pela *Google Cloud - Vision API*, nas placas veiculares identificadas na etapa anterior. A nível de comparação, este estudo também explora em paralelo outro algoritmo baseado em aprendizagem profunda contido no YOLO v4, que foi avaliado em trabalhos recentes da literatura [10] e para a fase do reconhecimento dos caracteres foi utilizado um serviço de reconhecimento alfanumérico em placas veiculares da plataforma Plate Recognizer. Por fim, para o reconhecimento e detecção das placas veiculares foi avaliado o desempenho de ambos os modelos estudados, através de um mesmo banco de imagens para treinamento e amostragem de imagens para testes.

1.2. Métricas de desempenho

Algoritmos que realizam a detecção de objetos através de imagens tem aplicabilidade em vários contextos como rastreamento de veículos, identificação de patologias em rodovias, detecção de espécies de insetos através de imagens ou reconhecimento facial, por exemplo. Esses algoritmos precisam de métricas que validem seu desempenho frente aos resultados encontrados ao problema proposto para detecção. As métricas mais utilizadas para validar esses algoritmos são a *Intersection over Union (IoU)* e a *Mean Average Precision (mAP)*.

Rosembrock explica que a IoU é uma métrica que verifica se a caixa delimitadora prevista pelo algoritmo para o objeto de busca na imagem, coincide com a caixa delimitadora real do objeto [18]. A Figura 5 apresenta a equação para se obter o valor do IoU, onde é necessário dividir a interseção computacional, que é a área de sobreposição entre a caixa delimitadora prevista e a caixa delimitadora real, pela área de domínio da união das caixas delimitadoras. Quanto maior for o resultado da equação, mais precisa será a seleção de um objeto, porém isso pode causar restrições ao algoritmo. Por outro lado, um valor baixo causa uma alta aceitação no algoritmo, o que reduz sua precisão. A Figura 6, apresenta uma relação entre os valores obtidos através da métrica IoU.

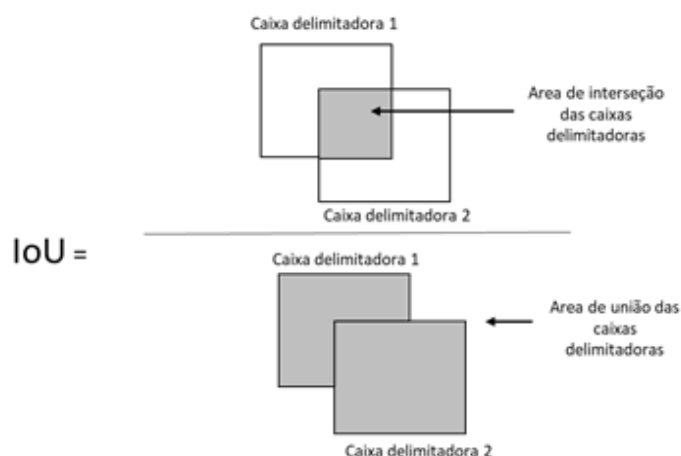


Figura 5. Representação esquemática da fórmula para calcular o IoU. (O autor. Adaptado de [21]).

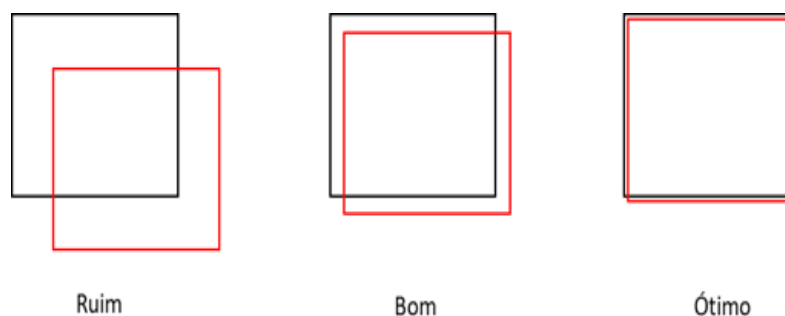


Figura 6. Em preto, a caixa delimitadora representa a localização real do objeto e em vermelho a localização estimada de acordo com o IoU. (O autor. Adaptado de [21]).

A *Mean Average Precision* (mAP) é uma métrica regularmente usada na etapa de treinamento do algoritmo. Para entender essa métrica é preciso compreender a equação da *Average Precision* (AP). Na etapa de treinamento do modelo, são coletadas informações sobre a precisão, que indica quantas classificações da classe Positivo que o modelo realizou estão corretas, e informações referente ao *recall*, que indica quantas classificações estão corretas dentre todas as situações da classe Positivo como valor esperado [17]. As Equações 1 e 2 apresentam as fórmulas para se obter a precisão e o *recall*, respectivamente. A mAP é calculada ao final do treinamento através da média aritmética do conjunto de APs que é obtido ao término do treinamento. Na escala de porcentagem, quanto maior for resultado do mAP melhor será o resultado do modelo treinado.

$$\text{Precisão} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Positivo}} \quad (1)$$

$$Recall = \frac{Verdadeiro\ Positivo}{Verdadeiro\ Positivo + Falso\ Negativo} \quad (2)$$

1.3. Objetivos

1.3.1. Objetivo Geral

Avaliar modelos baseados em aprendizado de máquina e aprendizagem profunda para detectar e reconhecer placas veiculares.

1.3.1. Objetivos Específicos

- Estudar e aplicar em conjunto os serviços de visão computacional da *Microsoft Azure - Custom Vision AI* e *Google Cloud - Vision AI*;
- Estudar e aplicar o algoritmo baseado em aprendizagem profunda para a detecção de objetos do *framework* YOLO v4, combinando os resultados com a plataforma *Plate Recognize* que possui serviços baseado em aprendizado profundo para o reconhecimento de caracteres em PVs;
- Obter métricas e avaliá-las em termos de precisão no reconhecimento das placas veiculares.

1.4. Organização do Trabalho

Este trabalho foi estruturado como segue: a Seção 2 apresenta o referencial teórico e as algumas técnicas disponíveis na literatura que tratam do problema de identificação de placas veiculares. A Seção 3 apresenta trabalhos que inspiraram esse experimento além de algumas técnicas abordadas pelos seus respectivos autores.

Na Seção 4, são discutidos os métodos aplicados, além de detalhar as etapas que foram seguidas neste experimento, bem como os recursos utilizados. Esta Seção também abordará o levantamento do conjunto de imagens para o treinamento dos algoritmos de reconhecimento de placas veiculares.

A Seção 5 contém os resultados obtidos através da comparação das diferentes abordagens exploradas para o mesmo conjunto de imagens. A Seção 6 apresenta as conclusões desse experimento e, por fim, a Seção 7 apresenta as referências bibliográficas utilizadas neste trabalho.

2. Referencial Teórico

Nos últimos anos, avanços significativos ocorreram nas áreas relacionadas à Inteligência Artificial e na Visão Computacional que permitiram o uso de soluções computacionais no trânsito, possibilitando a identificação de veículos para diversas finalidades como na aplicação de leis de trânsito e segurança veicular. Os últimos estudos destacam as Redes Neurais Convolucionais (CNN) [12], que é uma técnica de aprendizagem profunda com resoluções promissoras na atividade de classificação de imagens, segmentação e detecção de objetos. Outros estudos recentes relacionados à classificação de imagens referentes ao trânsito indicam uma tendência pela utilização de CNNs, como utilizado nos estudos de [19] e [28].

De acordo com o estudo publicado por [22], as Redes Neurais Convolucionais

treinadas para a detecção de objetos em imagens podem ser categorizadas em dois macro grupos que são os classificadores de um único estágio sem *Region Proposal Network* (RPN) e os de dois estágios com RPN. Os classificadores de dois estágios (Fast R-CNN) utilizam a RPN, como foi empregado nos estudos de [9] e [15], onde na primeira etapa sucede a produção de objetos candidatos através de caixas delimitadoras (do inglês, *bounding box*). A RPN é uma rede convolucional capaz de identificar as coordenadas do objeto na imagem, além de indicar seu nível de detecção. A segunda é responsável por extrair as características do objeto usando técnicas de *RoI-Pool* e realiza a classificação e regressão dos *bounding box* podendo ocorrer algumas variações dependendo do tipo da implementação. Os classificadores de um único estágio mais conhecidos e abordados na literatura são o *Single Shot Multibox Detector* (SSD), *You Only Look Once* (YOLO) e *RetinaNet*. Esses classificadores não possuem a RPN, logo é realizada uma proposição das regiões e das classes, que são os rótulos dos objetos que pretendem ser identificados, em apenas um estágio, sendo mais indicados para sistemas que demandam a detecção de objetos em tempo real.

O YOLO é uma arquitetura baseada em rede neural profunda *DARKNET*, elaborado em linguagem de programação C, e possui diversas versões publicadas com suporte ao uso de CPU e GPU. Das versões que foram disponibilizadas podemos encontrar o YOLO v2, YOLO v3, YOLO v4, YOLO v5 E PP - YOLO sendo a versão 4 considerada a mais promissora para a detecção de objetos em tempo real [6].

A detecção de objetos no YOLO v4 ocorre como um simples problema de regressão, onde uma determinada imagem de entrada será dividida em uma matriz $N \times N$ e cada célula da grade é incumbida de realizar a predição de um objeto que possa estar em seu centro. Desse modo, são geradas *bounding box* e uma pontuação é atribuída para representar o nível de confiança do detector a fim de identificar se existe um objeto nessa área da imagem. Por fim, é executada a predição das classes com base no valor de confiança *threshold*. Desse modo, o YOLO v4 (Figura 7) consegue em uma única etapa reconhecer diversos objetos analisando toda a imagem de entrada. É importante ressaltar que até esse presente estudo, as versões YOLO v5 e PP - YOLO não estão consolidadas em termos de documentação, sendo assim foram desconsideradas como embasamento teórico, optando-se por utilizar a versão mais promissora do YOLO que é a versão 4 como um dos métodos de estudo.

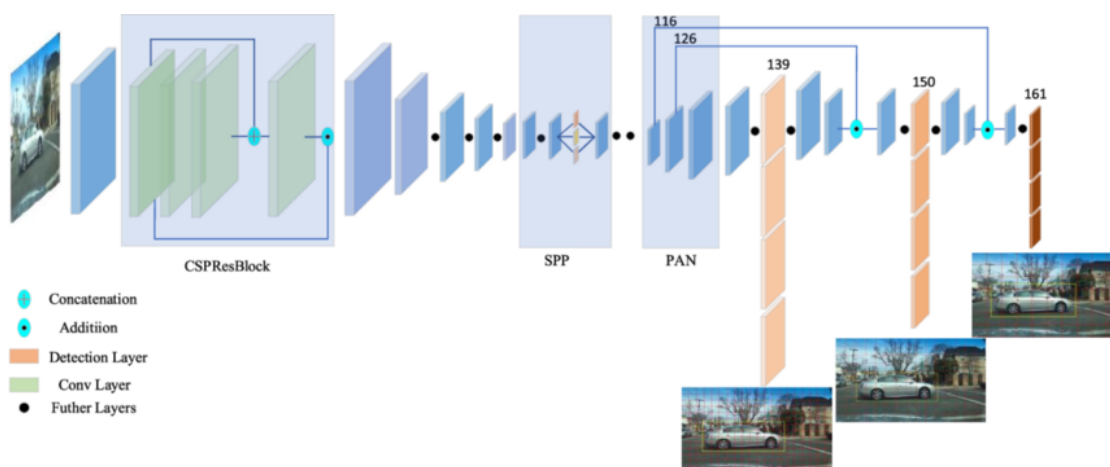


Figura 7. Arquitetura do YoloV4 [27].

A plataforma Azure da Microsoft fornece alguns serviços baseados em computação em nuvem, dentre eles está o de visão computacional *Custom Vision AI*. Neste serviço, a plataforma permite que os usuários realizem a criação e treinamento de modelos que realizam a detecção e classificação personalizada de imagens através de um sistema de redes neurais pré-treinadas. De maneira semelhante, a plataforma da *Google Cloud* fornece um serviço de detecção de objetos em imagens. O reconhecimento óptico de caracteres, é um dos módulos de visão computacional da Google, onde poderosos modelos de aprendizado de máquina realizam a detecção e o reconhecimento de textos através de imagens.

A *Plate Recognizer* é uma plataforma em nuvem dedicada à detecção e reconhecimento de placas veiculares com modelos de diversos países. No que diz respeito ao reconhecimento dos caracteres em PVs, a plataforma fornece um serviço de reconhecimento automático de números em placas veiculares através de modelos treinados de aprendizagem profunda.

3. Trabalhos Relacionados

No estudo de [24], os autores propuseram um método robusto para o reconhecimento de placas veiculares aplicado ao sistema de transporte inteligente com base técnica na remoção de sombras, inclinação da imagem, iluminação, ângulo de visão, cor da placa e um conjunto de algoritmos para o reconhecimento de caracteres. As principais estratégias que foram adotadas pelos autores é um novo método de remoção de sombra baseado no algoritmo de Bernsen aperfeiçoado e combinado com o filtro Gaussiano. Além disso, para o reconhecimento de caracteres, foi utilizado um algoritmo conhecido como a integração de máquina de vetores de suporte (SVM), no qual as características dos caracteres são extraídas da malha e toda a cadeia de caracteres é apontado como objeto de estudo em oposição a um único caracter. O algoritmo foi testado em um

conjunto de 9026 imagens capturadas em ambientes cuja a cena de captura varia, o que qualifica o experimento para ambientes considerados complexos. Unindo os resultados da detecção de placas veiculares e o reconhecimento de caracteres, os autores obtiveram o resultado de 93,54% de assertividade em seu experimento. O conjunto de imagens coletadas para a realização deste trabalho abrange diferentes países como Japão, Estados Unidos, China, Reino Unido e Alemanha, mas não considerou placas veiculares do padrão brasileiro.

Em [16], os autores apresentam uma proposta com foco na eficiência, para localizar placas veiculares em ambientes complexos, através da composição de um conjunto de imagens para treinamento. Um dos desafios dos sistemas de reconhecimento de placas veiculares são as condições climáticas adversas como chuva, neve ou neblina que influenciam na eficácia dos detectores de placas. Nesta proposta, o objetivo principal era comparar a precisão de um classificador, que foi treinado com um conjunto de imagens capturadas em ambientes ideais, com o mesmo classificador sendo treinado com imagens afetadas por climas adversos ou baixa iluminação para um mesmo conjunto de imagens de testes correspondentes. Os autores utilizaram técnicas de Histograma de Gradientes Orientados (HOG), Padrão Binário Local (LBP) além de algoritmos de integração de máquina de vetores de suporte (SVM). A proposta apresentou informações importantes sobre a influência de imagens em condições climáticas adversas e de baixa iluminação nos modelos de treinamento para a classificação de placas veiculares em ambientes complexos.

No ano de 2019, [20] apresentaram uma visão crítica e construtiva sobre as técnicas e métodos que foram utilizados na literatura recente relacionados ao desafio da detecção e reconhecimento de placas veiculares através do estudo de diversos pesquisadores desse campo, além de fornecer orientações para novos pesquisadores.

4. Experimentos

Este trabalho foi executado em 4 estágios separados, que possibilitaram a discussão dos resultados parciais, das dificuldades encontradas, além da possibilidade de melhorias para trabalhos futuros de maneira independente (Figura 8). Os quatro estágios citados são, respectivamente: coleta do banco de imagens, rotulação manual das imagens, treinamentos nas plataformas *Microsoft Azure - Custom Vision AI* e na arquitetura de aprendizado profundo YOLO v4 e scripts desenvolvidos em Python pelo autor, para o corte da região da placa veicular indicado pelos vetores da caixa delimitadora (Figura 9), bem como realizar a conexão com APIs que possibilitam a execução de técnicas de OCR.

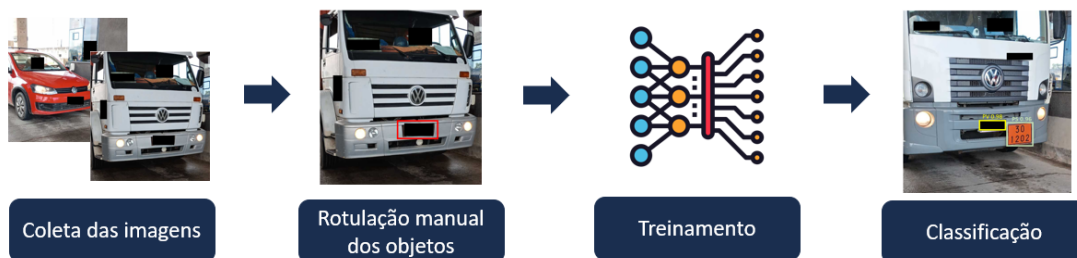


Figura 8. Fluxo com etapas seguidas no experimento.

```
def extrair_placa_da_caixa_delimitadora(imagem, vetores_da_caixa_delimitadora):
    nome = imagem.split("\\")[ -1]
    with open(imagem, 'rb') as imagem:
        data = imagem.read()

    img = cv.imdecode(np.array(bytearray(data), dtype='uint8'), cv.IMREAD_COLOR)
    l,t,w,h = vetores_da_caixa_delimitadora[0],vetores_da_caixa_delimitadora[1],vetores_da_caixa_delimitadora[2],vetores_da_caixa_delimitadora[3]
    multiplas_linhas = np.multiply([[l,t],[l+w,t],[l+w,t+h],[l,t+h]], [img.shape[1],img.shape[0]])
    img2 = cv.polylines(img, np.int32([multiplas_linhas]), 1, (255,255,0), 1, lineType=cv.LINE_AA)

    corta_eixo_x = multiplas_linhas[:,0].astype('uint16')
    cortar_eixo_y = multiplas_linhas[:,1].astype('uint16')

    recorte_da_placa = img2[np.min(cortar_eixo_y):np.max(cortar_eixo_y), np.min(corta_eixo_x):np.max(corta_eixo_x)]
    cv.imwrite("./Placas/"+'placa-' + nome, recorte_da_placa)
```

Figura 9. Script em Python desenvolvido pelo autor para o corte de caixas delimitadoras.

No primeiro estágio deste experimento foram realizadas coletas de imagens de veículos na entrada de praças de pedágio, localizadas na cidade de Cabo de Santo Agostinho, região metropolitana do Recife - PE. Praças de pedágio são considerados ambientes simples, devido a repetição da cena nas quais os veículos são capturados nas imagens. Contudo, pelo fato da coleta ser realizada manualmente através de dispositivos como smartphones, podemos identificar variações no ângulo de captura da cena. Além disso, ocorreram fatores climáticos como chuva, que podem influenciar na iluminação do ambiente, o que impacta diretamente no resultado dos registros realizados nesses dias.

Após alguns dias de coleta, foi montado um dataset com 1172 imagens que serão usadas para treinamento e validação nos estágios de treinamento da aprendizagem de máquina. Além dessas imagens de treinamento, foram coletadas imagens exclusivamente para compor um pequeno dataset contendo 153 imagens para realização de testes. As imagens coletadas apresentam diversos modelos e categorias de veículos, como caminhão de um ou múltiplos eixos, veículos de passeio, motocicletas dentre outros. Além disso, foram coletadas placas que possuem o modelo atual utilizado no Mercosul bem como placas do modelo anterior.

O segundo estágio do experimento é a realização da rotulação manual dos objetos de placas veiculares nas imagens. O serviço de visão computacional *Custom Vision AI* permite que o usuário cadastrado realize o upload do *dataset* de imagens, para que na etapa posterior sejam validadas e rotulados os objetos de busca. No cenário de visão computacional para a detecção de objetos, os rótulos são conhecidos como classes. Foram selecionados dois conjuntos de classes rotulados como “placas-veiculares” e “painel-de-segurança”. O uso de dois conjuntos de classes, ao

invés de um único conjunto, ocorreu nesta etapa pois os primeiros resultados de testes realizados com um treinamento ainda prematuro no ambiente de visão computacional da Azure, detectaram incorretamente a placa veicular, que foi confundida com os painéis de segurança de caminhões que indicam o tipo da carga. Desse modo, foi definido que esses dois objetos seriam devidamente classificados e treinados para evitar esse tipo de ocorrência. De modo paralelo a esta fase do experimento, a rotulação das imagens presentes no dataset para o treinamento na arquitetura YOLO, foi realizada com o auxílio da ferramenta gráfica *LabelImg* [23], que gera um conjunto de arquivos TXT nas quais são registradas as regiões da imagem nas quais os objetos foram classificados.

O treinamento de ambos os modelos caracteriza o terceiro estágio desse experimento. Referente ao modelo da *Custom Vision AI*, a plataforma possui métricas passíveis de configuração que são os limites de probabilidade e os limites de sobreposição. Essas métricas podem ser facilmente ajustadas na plataforma a fim de melhorar o resultado da precisão do treinamento, contudo foram mantidos os valores padrões para o limite de probabilidade em 50% e o limite de sobreposição em 30%. Após o treinamento, a plataforma executa um processamento como *k-fold cross validation* que permite determinar a acurácia do algoritmo que recomenda ao usuário o aprimoramento do modelo treinado com mais imagens caso haja necessidade. Com o dataset construído nesse estudo, o treinamento do modelo que permite a detecção de objetos no Custom Vision AI demorou em torno de 4 à 6 horas e foi utilizada a versão que atende as cotas gratuitas da plataforma.

O segundo modelo treinado aplicando o dataset de imagens coletadas nesse estudo, utilizou-se do *Darknet* que é uma biblioteca de aprendizagem de máquina desenvolvida em linguagem C que opera redes neurais e que nos permite usar a arquitetura do YOLO v4 para a detecção de objetos em imagens.

O treinamento foi executado através da plataforma *Google Collaboratory*, que é um serviço em nuvem gratuito mantido pela Google conhecido como *Google Colab* que tem como objetivo o incentivo à pesquisa. Desde outubro de 2018, o *Google Colab* viabiliza o uso de uma GPU NVIDIA Tesla K80 de 12GB que pode ser utilizada na plataforma por até 12 horas continuamente. Após o período de uso da GPU ser excedido, as atividades só podem ser retomadas após um período de 24 horas. Nesse estudo, foi constatado que o período de uso da GPU de 12 horas na plataforma para realizar o treinamento, nem sempre era atendido, tendo o uso da GPU suspenso em menos tempo. Devido às limitações de capacidade da GPU, algumas configurações no YOLO v4 precisaram ser ajustadas para não exceder o limite de memória da GPU, como o número de subdivisões por *batch*, que são parâmetros que informam quantas imagens por *minibatch* devem ser enviados para processamento na GPU; nesse experimento foi configurado o valor de *batch* em 64 e o total de subdivisões ajustado para 64, o que resulta em uma imagem por *minibatch* para serem enviados para processamento. Na fase inicial do treinamento com o YOLO v4, foi executada uma técnica conhecida como transferência de aprendizagem, na qual se utiliza de um modelo com pesos pré-treinados nas camadas convolucionais, desenvolvido por [1], que contribuirão na evolução do algoritmo para a detecção de alguns padrões que são comuns às imagens, tais como bordas, linhas e cores. Devido às limitações de processamento, o treinamento da aprendizagem profunda levou em média 7 dias para

que o modelo treinado alcançasse a geração de número 2600. No que se refere às métricas utilizadas para a validação do modelo, foram consideradas a Precision, que considera a assertividade do modelo, o *Recall*, a *Mean Average Precision* (mAP) e *Intersection Over Union*, que serão abordadas na Seção 5 com os resultados obtidos neste trabalho.

Após o treinamento, os modelos utilizados nesse experimento receberam um mesmo conjunto de imagens para que fosse possível comparar a assertividade em detectar os objetos classificados como placas veiculares nas imagens do conjunto. Nesta última fase do experimento, o serviço da *Custom Vision AI* fornece acesso a uma API na qual é possível utilizar o modelo treinado para realizar predições em imagens. Para isso, foi utilizado um script na linguagem de programação Python, que é responsável por acessar a API, enviar cada imagem para a plataforma e recuperar as informações da predição, como a região da caixa delimitadora que indica onde está o objeto na imagem com a maior probabilidade de ser uma placa veicular. O script realiza um processamento na qual a caixa delimitadora é recortada da imagem original para gerar uma imagem focada apenas na região da placa veicular. Essa etapa é importante, pois a imagem da placa veicular extraída é enviada através do script para os serviços de visão computacional da *Google AI*, que realiza o reconhecimento óptico de caracteres (OCR). A plataforma do *Google Cloud* disponibiliza um acesso gratuito pelo período de um ano e possui uma API que nos permite utilizar os serviços de detecção e reconhecimento de textos através de imagens.

Sobre o modelo Yolo, para realizar o reconhecimento óptico dos caracteres, optou-se por utilizar a plataforma *Plate Recognizer* que possui os serviços de reconhecimento alfanumérico de placas veiculares através de um modelo de aprendizado profundo dedicado a esse serviço. Para isso foi criada uma conta de usuário na versão de testes e o script em Python é responsável por realizar a integração do nosso modelo treinado para reconhecer placas veiculares com a plataforma de reconhecimento de caracteres em placas veiculares. Assim como realizado no método anterior, o script Python realiza a extração das caixas delimitadoras contendo os objetos detectados pelo modelo treinado e as enviam para a plataforma de modo que os caracteres presentes na imagem sejam extraídos.

4.1. Conjunto de Dados

Na formação do conjunto de imagens para o treinamento dos modelos inteligentes de ambas as metodologias abordadas neste trabalho, foram reunidas 1172 imagens em formatos JPG nas quais possuíam aproximadamente 975 pixels de largura por 1300 pixels de altura ou 1300 pixels de largura por 975 pixels de altura, além de um conjunto menor de 150 imagens para realizar o experimento a nível de comparação dos resultados obtidos. A coleta do conjunto de imagens ocorreu em um ambiente de praças de pedágio localizadas na cidade de Cabo de Santo Agostinho, que fica localizada no litoral sul do estado de Pernambuco, nas quais realizam o acesso ao complexo portuário de SUAPE e destinos turísticos do litoral sul do estado.

As imagens foram coletadas através de um aplicativo para *smartphones* com o sistema operacional Android, no período diurno e em dias ensolarados ou com pancadas de chuva. Os registros fotográficos foram manualmente coletados de tal maneira que a

placa veicular estivesse centralizada no ângulo de captura da cena, logo encontramos no dataset imagens próximas e distantes, além de imagens frontais ou da traseira dos veículos. O aplicativo realiza a compactação das imagens para que sejam sincronizadas com um sistema WEB, o que causa uma redução na resolução das imagens coletadas.

5. Resultados

Foram executados testes utilizando o mesmo conjunto de 153 imagens para a detecção e o reconhecimento de caracteres nas placas veiculares nos dois métodos pesquisados neste trabalho. A primeira etapa dos testes consiste em detectar e extrair o objeto da placa veicular de cada imagem. A segunda etapa trata do reconhecimento óptico dos caracteres presentes na imagem da placa que foi extraída na primeira etapa.

Para o modelo treinado na plataforma da Microsoft Azure, a detecção das placas veiculares para o conjunto de teste durou em média duas horas de execução. O modelo obteve uma taxa de assertividade na localização da placa do veículo maior que 98% conforme a Tabela 1. A Tabela 2 apresenta os resultados gerais das métricas que definem a performance da plataforma para as classes treinadas no modelo.

Tabela 1. Resultado da localização das placas (Custom Vision)

Acertos	150
Erros	3
Total de imagens	153
Localização das Placas	98,3%

Tabela 2. Resultado de performance (Custom Vision)

Classe	Precisão	Recall	AP
Placas Veiculares	97,7%	98,7%	100%
Painel de Segurança	94,1%	94,1%	94,1%

Para a segunda etapa dos testes levando em consideração os resultados da localização das placas veiculares realizadas no modelo treinado da *Custom Vision AI*, foi desenvolvido pelo autor um script em Python para enviar as placas veiculares localizadas nas imagens do conjunto de teste para a plataforma de reconhecimento óptico de caracteres da *Google Cloud - Vision AI*. A Tabela 3 apresenta o resultado do OCR aplicado às placas veiculares.

O segundo método explorado nesta pesquisa para a detecção de placas veiculares, foi através de um modelo personalizado de treinamento de aprendizagem profunda utilizando o YOLO v4. O modelo foi treinado para identificar dois tipos de classes rotulados de 'placa-veicular' e 'painel-de-segurança'. O modelo treinado obteve os resultados para as métricas mAP e IoU de 98% e 81,64% respectivamente. Considerando as classes treinadas neste modelo, a Tabela 4 apresenta os resultados obtidos após o treinamento.

Tabela 3. Resultado do reconhecimento óptico de caracteres (Google Vision)

Total de Placas	150
Acertos	109
Erros	41
Reconhecimento de Caracteres	71,2%

Tabela 4. Resultado de performance (DARKNET - YOLO v4)

Classe	Precisão	Recall	AP
Placas Veiculares	99%	100%	99,9%
Painel de Segurança	79%	94,1%	96%

Na etapa de detecção de placas veiculares, o YOLO v4 detectou corretamente todas as PVs. Foi utilizado o mesmo conjunto de imagens que no modelo da treinado na Custom Vision AI para a realização dos testes. Na tabela abaixo, são listados os resultados obtidos nessa etapa de detecção.

Tabela 5. Resultado de performance (DARKNET - YOLO v4)

Acertos	153
Erros	0
Total de imagens	153
Localização das placas	100%

Na fase do reconhecimento de caracteres, o script Python enviou as imagens das placas detectadas para a plataforma *Plate Recognize* que possui modelos treinados baseados em aprendizagem profunda para a detecção e reconhecimento de caracteres. Nesse sentido foi utilizado apenas o modelo treinado que realiza o reconhecimento óptico de caracteres em placas veiculares. A Tabela 6, apresenta os resultados obtidos para esse modelo especializado no reconhecimento de caracteres das placas.

Tabela 6. Resultado do reconhecimento óptico de caracteres (*Plate Recognize*)

Total de Placas	153
Acertos	142
Erros	11
Reconhecimento de Caracteres	92,8%

6. Conclusões

Este trabalho realizou um estudo sobre técnicas de detecção e reconhecimento de placas veiculares (PV) em imagens através da aplicação de aprendizagem de máquina. Adotou-se dois modelos de algoritmos para a detecção das placas veiculares, onde o primeiro modelo é disponibilizado pela Microsoft Azure - *Custom Vision AI* e o segundo é a arquitetura do YOLO v4. Esta última, permitindo ajustes de parâmetros na etapa de treinamento. Para a etapa de reconhecimento óptico de caracteres foram utilizados serviços em nuvem da *Google Cloud - Vision AI* e *Plate Recognizer*.

Para que os algoritmos responsáveis pela detecção das placas veiculares fossem avaliados de natureza semelhante, foi utilizada a mesma base com 1172 imagens na etapa de treinamento, nas quais foram rotuladas as placas veiculares e painéis de segurança que são placas que identificam o tipo de carga que caminhões carregam. Painéis de segurança foram rotulados devido aos testes preliminares com o *Custom Vision* confundirem a PV com esse objeto.

Os dois algoritmos apresentaram resultados satisfatórios na etapa de detecção de placas veiculares e valores de precisão muito semelhantes. Contudo, a versão do YOLO v4 conseguiu identificar todas as placas no conjunto de teste executado em ambos os modelos, enquanto o experimento na plataforma da Microsoft Azure deixou de reconhecer algumas poucas placas no conjunto de testes. Para a etapa de reconhecimento óptico de caracteres, as duas plataformas utilizadas apresentaram resultados diferentes. A *Google Cloud - Vision AI* recebeu as imagens recortadas das placas veiculares que foram detectadas pelo serviço da *Custom Vision AI*. O OCR da Google reconheceu pouco mais de 71% das placas veiculares corretamente frente aos quase 93% de reconhecimento realizado pela *Plate Recognize* que recebeu as placas detectadas pelo modelo treinado do YOLO v4. Em linhas gerais o reconhecimento da plataforma *Plate Recognize* teve melhor desempenho. Contudo, é importante ressaltar que o OCR da Google, atende a detecção e o reconhecimento de textos oriundos de diversas fontes, enquanto o *Plate Recognize* possui modelos baseados em aprendizagem profunda treinados para o reconhecimento de caracteres em placas veiculares incluindo PV dos modelos brasileiros em seu treinamento.

Como apresentado ao longo deste artigo, o experimento foi realizado através de ferramentas acessíveis, que realizam tanto a detecção de objetos em imagens quanto o reconhecimento de textos através de imagens, por meio da aprendizagem de máquinas. Assim, é importante ressaltar que apesar do conhecimento limitado do autor sobre o tema, é possível utilizar da facilidade e baixa curva de aprendizado das ferramentas utilizadas neste trabalho, para o desenvolvimento de diversas soluções.

7. Referências Bibliográficas

1. ALEXEY. AlexeyAB. Github. Disponível em: <<https://github.com/AlexeyAB/darknet>>. Acesso em: 17 nov. 2021.
2. ASSOCIAÇÃO NACIONAL DOS DETRANS. Brasil já tem 1 carro a cada 4 habitantes. AND - Associação Nacional de Detrans, 2017. Disponível em:

<<http://www.and.org.br/brasil-ja-tem-1-carro-a-cada-4-habitantes-diz-denatran/>>. Acesso em: 16 nov. 2021.

3. AZEVEDO, S. D.; RIBEIRO, L. C. D. Q. Mapa da motorização individual no Brasil. Instituto de Pesquisa e Planejamento Urbano e Regional. Rio de Janeiro, p. 27. 2019.

4. BRITO, B. Infrações e multas por excesso de velocidade aumentam na pandemia. A Tarde, 2020. Disponível em:

<<https://atarde.uol.com.br/transito/noticias/2139854-infracoes-e-multas-por-excesso-de-velocidade-aumentam-na-pandemia>>. Acesso em: 15 nov. 2021.

5. CARDOSO, R. autoescolaonline, 2016. Disponível em:

<<https://www.autoescolaonline.net/author/rrcard/>>. Acesso em: out. 2021.

6. DI DOMÊNICO, J. R.; LAU, D.; RIBEIRO, D. Um estudo comparativo de redes convolucionais profundas para detecção de insetos em imagens. Embrapa Trigo-Artigo em anais de congresso (ALICE), Porto Alegre, 2021.

7. DOUTOR MULTAS. Tudo o que você precisa saber sobre evasão de pedágio. g1, 2019.

Disponível em:

<<https://g1.globo.com/go/goias/especial-publicitario/dr-multas/dicas-de-transito-direcao-segura/noticia/2019/07/18/tudo-o-que-voce-precisa-saber-sobre-evasao-de-pedagio-fique-por-dentro.ghtml>>. Acesso em: 16 nov. 2021.

8. GABO, V. vitorgabo. Medium, 2020. Disponível em:

<<https://vitorgabo.medium.com/reconhecimento-de-placas-veiculares-brasileiras-de12f77917dd>>. Acesso em: nov. 2021.

9. GIRSHICK, R. Fast r-cnn. Proceedings of the IEEE international conference on computer vision, 2015. 1440-1448.

10. ISMAIL, A.; MEHRI, M.; SAHBANI, A. Performance Benchmarking of YOLO Architectures for Vehicle License. VISIGRAPP, 2021. 661-668.

11. LAROCA, R.; ZANLORENSI, L.; GONÇALVEZ, G. An Efficient and Layout-Independent Automatic License Plate Recognition System Based on the YOLO detector. ET Intelligent Transport Systems, 2021. 483-503.

12. LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, 27 Fevereiro 2015. 436–444.
13. LUO, L. et al. An Efficient Method of License Plate Location. First International Conference on Information Science and Engineering, 2009. 770-773.
14. OLIVEIRA, J. Governo do Ceará. ceara.gov, 2020. Disponível em: <ceara.gov.br/2020/03/16/detran-esclarece-suas-duvidas-sobre-a-placa-padrao-mercosul/>. Acesso em: out. 2021.
15. REN, S. et al. Faster R-CNN: Towards Real-Time Object Detection. Advances in neural information processing systems, 2015. 91-99.
16. RIO-ALVAREZ, A. et al. Effects of Challenging Weather and Illumination on Learning-Based License Plate Detection in Noncontrolled Environments. Scientific Programming, 2019.
17. RODRIGUES, V. Métricas de Avaliação: acurácia, precisão, recall... quais as diferenças? Medium, 2019. Disponível em: <https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c>. Acesso em: 17 nov. 2021.
18. ROSEMBROCK, A. Intersection over Union (IoU) for object detection. Pyimagesearch, 2016. Disponível em: <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>. Acesso em: 17 nov. 2021.
19. SELMI, Z.; BEN HALIMA, M.; M. ALIMI, A. Deep Learning System for Automatic License Plate Detection and Recognition. 14th IAPR International Conference on Document Analysis and Recognition, 2017. 1132-1138.
20. SHASHIRANGANA, J. et al. Automated License Plate Recognition: A Survey on Methods and Techniques. IEEE. 11203-11225.

21. SUBRAMANYAM, V. S. IOU (Intersection over Union). Medium, 2021. Disponível em: <<https://medium.com/analytics-vidhya/iou-intersection-over-union-705a39e7acef>>. Acesso em: out. 2021.
22. SUN, Y. et al. Automatic in-trap pest detection using deep learning for pheromone-based *Dendroctonus valens* monitoring. *Biosystems engineering*, 2018. 140-150.
23. TZUTALIN, D. LabelImg. GitHub, 2018. Disponível em: <<https://github.com/tzutalin/labelImg>>. Acesso em: out. 2021.
24. WEN, Y. et al. An Algorithm for License Plate Recognition Applied to Intelligent Transportation System. *IEEE Transactions on intelligent transportation systems*, 2011. 830-845.
25. WENJING , J. et al. Gaussian Weighted Histogram Intersection for License Plate Classification. *18th International Conference on Pattern Recognition*, 2006. 574-577.
26. XIE, L. et al. A New CNN-Based Method for Multi-Directional Car License Plate Detection. *IEEE Transactions on Intelligent Transportation Systems*, fev. 2018. 507-517.
27. Y. Cai et al., "YOLOv4-5D: An Effective and Efficient Object Detector for Autonomous Driving," in *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-13, 2021
28. ZOU, L. et al. License plate detection with shallow and deep CNNs in complex environments. *Complexity*, 2018.