Avaliação de Técnicas de Detecção de Pedestres para Veículos Autônomos

Gabriel Reis[1], **Wellington Lobato**[2], **Denis Rosário**[1],
Eduardo Cerqueira[1], **Leandro A. Villas**[2]

[1] Universidade Federal do Pará (UFPA)

[2]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)

gabriel.reis.ribeiro@itec.ufpa.br, wellington@lrc.ic.unicamp.br,
{denis, cerqueira}@ufpa.br, leandro@ic.unicamp.br

Abstract. *Object detection is one of the main applications within the context of Autonomous Vehicles (AVs). Pedestrian detection applications make up the vehicular perception layer, using sensors and cameras to detect the presence of objects in the area close to the AV. However, pedestrian detection techniques have limitations and restrictions according to the behavior of the vehicular scenario, mainly due to the variation in lighting conditions and the size of pedestrians. This article presents a comparative study of the main pedestrian detection techniques for AVs. The evaluation considers four types of detection techniques, namely: Faster Region-based Convolutional Neural Network (Faster R-CNN), Single Shot MultiBox Detector (SSD), You Only Look Once (YOLO), and RetinaNet. The evaluation results indicate that the YOLO and Faster R-CNN approaches achieved superior processing time and pedestrian detection performance, with an average detection time of 0.58 seconds per image.*

Resumo. *A detecção de objetos é uma das principais aplicações dentro do contexto dos Veículos Autônomos (AVs). As aplicações de detecção de pedestres compõem a camada de percepção veicular, utilizando sensores e câmeras para detectar a presença de objetos na área próxima ao AV. No entanto, as técnicas de detecção de pedestre apresentam limitações e restrições de acordo com o comportamento do cenário veicular, principalmente devido à variação das condições de iluminação e o tamanho dos pedestres. Nesse contexto, este artigo apresenta um estudo comparativo das principais técnicas de detecção de pedestre para AVs. A avaliação considera quatro tipos de técnicas de detecção, sendo eles: Faster R-CNN, SSD, YOLO e RetinaNet. Os resultados da avaliação indicam que as abordagens YOLO e Faster R-CNN obtiveram desempenho superior em termos de tempo de processamento e detecção de pedestres, apresentando uma média de detecção de 0,58 segundos por imagem.*

1. Introdução

A nova geração de veículos guiados (Autonomous Vehicles (AVs)) sem a intervenção humana/condutor, será baseada em dados de GPS, sensores/atuadores, mapas digitais e visão computacional, a qual irá surgir em cidades inteligentes, e serão incluídos no portfólio da Internet das Coisas [Saleh et al. 2020]. Nos AVs, existem 5 níveis de direção autônoma, desde o auxílio ao condutor com algumas atividades simples, como

é o caso da manutenção de aceleração por meio do uso de funções como *Cruise Control* não-adaptativo (Nível 1), até a completa direção autônoma dos veículos (nível 5), passando por níveis de direção supervisionada pelo condutor (Níveis 2 até 4) [Hussain and Zeadally 2018].

O funcionamento dos AVs consideram três camadas: percepção, planejamento e controle [Wang et al. 2018, Lobato et al. 2023]. Os sensores, como ultrasonic Radar (*RAdio Detection And Ranging*), LIDAR (*LIght Detection And Ranging*) e câmeras, servirão de entrada para a camada de percepção, o que pode ser considerado como pré-requisito para a realização de condução autônoma. A camada de planejamento utiliza as informações obtidas pela camada de percepção, além da resposta da camada de controle. Por fim, a camada de controle, implementa o controle sobre o veículo de acordo com as instruções emitidas pela camada de percepção para condução autônoma [Pendleton et al. 2017, Liu et al. 2017].

A percepção completa do ambiente e sua correta interpretação são cruciais para as aplicações de percepção dos AVs, como exemplo, as aplicações do sistemas assistência ao motorista (*Advanced Driver Assistance System* - ADAS), sistema de prevenção de colisão, *blind crossing* e detecção pedestres [Xu et al. 2017]. Nesse contexto, as técnicas de detecção de pedestres permitem que os AVs obtenham uma percepção mais precisa e abrangente de seus arredores [Yang et al. 2021]. As técnicas de detecção são capazes de localizar e categorizar objetos e pedestres. Quando um pedestre ou obstáculo é detectado, a camada de planejamento recebe as informações de posição, velocidade e aceleração dos objetos obtidos pela camada de percepção. Dessa forma, a camada de controle, altera a direção, velocidade e aceleração sobre o veículo baseado na percepção dos sensores.

A detecção de objetos é uma tarefa fundamental na área de visão computacional e um componente necessário para o funcionamento das aplicações de AVs [Krijestorac et al. 2020]. As técnicas de detecção de objetos envolvem um modelo classificador a partir de imagens que contêm o objeto a ser detectado e imagens que não o contêm. Esses modelos são treinados para identificar características específicas do objeto, como sua forma, cor e textura. Além disso, as técnicas de detecção de objetos, geralmente utilizam janelas deslizantes para percorrer toda a imagem em busca de características. Essas janelas deslizantes são usadas para determinar se o objeto está presente em uma determinada região da imagem [Yang et al. 2021].

As técnicas de detecção de pedestres, aplicadas para os AVs, devem ser capazes de funcionar em diferentes condições de iluminação, clima e responder de forma eficiente e acurada [Garlichs et al. 2018, Rasouli and Tsotsos 2020]. Nesse contexto, este artigo apresenta uma estudo comparativo das principais técnicas de segmentação de imagem para aplicações de percepção dos AVs. A avaliação considera as quatro técnicas de detecção, que são: Faster Region-based Convolutional Neural Network (Faster R-CNN), Single Shot MultiBox Detector (SSD), You Only Look Once (YOLO) e RetinaNet. Foi realizada uma avaliação exploratória com o *dataset Caltech Pedestrian* com o objetivo de quantificar o desempenho das técnicas de detecção de pedestres, em termos de tempo de detecção, número de pedestres detectados e erros de localização. Os resultados mostram que as técnicas YOLO e Faster R-CNN obtiveram resultados superiores em relação ao tempo de detecção, com reduções de até 95% e 84% de tempo, em comparação com outros trabalhos da literatura, *i.e.*, SSD e RetinaNet.

O restante deste artigo está organizado conforme descrito a seguir. A Seção 2 explica o funcionamento de cada tecnologia e apresenta os trabalhos relacionados. A Seção 3 apresenta o cenário de avaliação, explica detalhes de implementação das diferentes técnicas e discute os resultados obtidos. Por fim, a Seção 4 conclui o artigo e apresenta as principais direções para o trabalho futuro.

2. Trabalhos Relacionados

Redmon et al. desenvolveram a abordagem para detecção de objetos denominada YOLO [Redmon et al. 2016]. Uma única rede neural convolucional (Convolutional Neural network (CNN)) prevê simultaneamente várias probabilidades de classe para os objetos detectados e demarcados. Em uma CNN para reconhecimento de objetos, uma camada pode aprender a reconhecer bordas e contornos, enquanto outra camada pode aprender a combinar esses recursos para reconhecer objetos com formatos complexos [Rahmad et al. 2020]. Isso permite que as CNNs sejam muito eficientes na extração de informações relevantes a partir de dados de imagem. Ao contrário das técnicas baseadas em janela deslizante e região de interesse, o YOLO analisa a imagem uma única vez durante o treinamento, de modo que codifica implicitamente informações contextuais sobre as categorias, bem como sua aparência.

Girshick et al. apresentaram a técnica para detecção de objetos utilizando CNN denominada Faster R-CNN [Girshick et al. 2015]. A Faster R-CNN divide a imagem de entrada em regiões de interesse menores. A técnica aprende conjuntamente a classificar objetos e refinar suas localizações espaciais. As CNNs aprendem de forma hierárquica, o que possibilita extrair características de diferentes dimensões a partir dos dados de entrada [Tan and Le 2021]. Durante o treinamento, as caixas delimitadoras de detecção preditas são categorizadas como verdadeiras ou falsas com base em sua sobreposição com as caixas delimitadoras reais. A saída da camada final da CNN é então alimentada em um classificador para determinar cada região de interesse como pertencente ou não a um objeto na imagem.

Liu et al. propuseram a técnica SSD para detectar objetos em imagens usando uma única rede neural profunda [Liu et al. 2016]. A abordagem SSD é baseada em uma CNN que produz um conjunto de tamanho fixo de caixas delimitadoras e pontuações para a presença de categorias de classe de objeto detectados nessas caixas delimitadoras. O processo de detecção é semelhante ao apresentado no YOLO e otimizado para processar imagens com menor resolução.

Lin et al. introduziram uma rede unificada composta por uma rede *backbone* e duas sub-redes específicas para tarefas, denominada RetinaNet [Lin et al. 2020]. O *backbone* é responsável por computar um mapa de recursos convolucionais sobre uma imagem de entrada inteira e é uma CNN independente. O RetinaNet é uma técnica de detecção de objetos que utiliza uma função de perda focal para lidar com o problema de desequilíbrio de categorias na detecção de objetos e analisa a imagem uma única vez, o que o torna eficiente para detectar objetos menos frequentes.

Os trabalhos de Liang et al. e Vora et al. propõem uma abordagem de fusão de dados para combinar os dados coletados pelas câmeras e varreduras LIDAR, com o objetivo de obter informações de detecção mais acuradas [Liang et al. 2018, Vora et al. 2020]. Os trabalhos propõem que a câmera forneça informações visuais, como posição, cor e for-

mato dos objetos, enquanto o LIDAR fornece informações de profundidade. A fusão desses dois tipos de dados permite que a aplicação de percepção tenha uma compreensão mais completa do ambiente, o que é útil para tarefas como detecção e classificação de objetos e pedestres, dentro do contexto de AVs.

Kim et al. analisaram três técnicas de aprendizado profundo (Faster R-CNN, YOLO e SSD) para reconhecimento de tipo de veículo em tempo real [Kim et al. 2020]. O objetivo foi determinar qual técnica é a mais adequada para essa tarefa com base em sua precisão e velocidade. De acordo com os autores, a técnica YOLO é a mais adequado para reconhecimento de tipo de veículo em tempo real. A técnica Faster R-CNN também teve um desempenho muito bom, com uma taxa de precisão média de 93,2%, mas foi mais lenta do que o YOLO. A técnica SSD obteve o menor tempo de processamento, mas a taxa de precisão média foi a mais baixa, em torno de 85%.

Outra área em que a detecção de objetos tem sido aplicada com sucesso é na segurança de trânsito. O trabalho de Rasouli et al. propõe uma abordagem de detecção de pedestres em vídeos de trânsito utilizando uma combinação de técnicas para prever comportamento dos pedestres [Rasouli and Tsotsos 2020]. Os resultados demonstraram que a abordagem é capaz de detectar pedestres com uma baixa taxa de falsos positivos, enquanto que mantém uma alta taxa de detecção. Isso é fundamental para aplicações de trânsito, pois permite que os sistemas identifiquem pedestres e tomem medidas de precaução para evitar acidentes.

3. Avaliação Comparativa

Esta seção detalha a metodologia e as métricas usadas para avaliar o nível de detecção de pedestres por diferentes técnicas de segmentação de imagem, tais como Faster R-CNN, SSD, YOLO e RetinaNet. Posteriormente, é avaliado o impacto dos erros de detecção de pedestres aplicados no *dataset Caltech Pedestrian*, tempo de detecção, bem como o número total de pedestres detectados.

3.1. Descrição do Cenário de Avaliação

Para avaliar as técnicas de detecção de pedestres, foi utilizado o *framework Open Source TorchVision*[1], versão 0.15.1, do PyTorch[2], que é a versão mais recente estável do TorchVision. Essa versão estável foi escolhida, pois ela possui recursos que serão mantidos a longo prazo e não devem ter grandes limitações de desempenho ou lacunas na documentação. O PyTorch consiste em conjuntos de dados, arquiteturas de modelos e operações de imagens para visão computacional [maintainers and contributors 2016]. O PyTorch é uma biblioteca usada para aplicações como visão computacional e processamento de linguagem natural, originalmente desenvolvida pela *Meta AI* e agora parte da *Linux Foundation*. A biblioteca TorchVision permite reproduzir e avaliar os modelos pré-treinados pelas diferentes técnicas de detecção avaliadas neste trabalho. Foram realizadas 2.456 avaliações variando o conjunto total de 614 imagens de teste, cada imagem contendo um número variado de pedestres. Os resultados apresentaram valores com intervalo de confiança de 95%.

[1] `https://pytorch.org/vision/stable/index.html/`
[2] `https://pytorch.org/`

É importante destacar que as técnicas de detecção de pedestre exigem que o modelo esteja inicializado previamente. Portanto, é necessário fornecer um conjunto de imagens de treinamento para modelar o classificador e categorizar os objetos. Além disso, as técnicas avaliadas tendem a ser menos precisas quando o objeto detectado aparece em diferentes tamanhos e orientações na imagem. Dessa forma, foi utilizado o mesmo conjunto de imagens para avaliar os principais comportamentos de cada técnica. Para ajudar avaliar a performance das técnicas de detecção disponibilizada pelos seus respectivos pesquisadores, foi fornecido o Mean Average Precision (mAP) de cada técnica. O mAP é uma técnica muito usada para avaliar a precisão de técnicas de detecção, entregando uma avaliação abrangente do desempenho do modelo. Ele é calculado como a média das precisões médias para cada classe de objetos detectado.

Nas avaliações, foi considerado o *dataset Caltech Pedestrian* disponibilizado por Dollár et al. [Dollar et al. 2011]. Existem poucos conjuntos de dados disponíveis publicamente que são adaptados para aplicações de percepção de pedestres [Ye et al. 2021]. O *dataset Caltech Pedestrian* é considerado o maior conjunto de dados desenvolvido para avaliação de técnicas de detecção de pedestres, contendo uma grande variedade de imagens de pedestres em diferentes escalas, no contexto de mobilidade urbana. O *dataset* contém 350.000 caixas delimitadoras em torno de 2.300 pedestres. Oclusões e correspondências temporais também são categorizadas, imagens onde o pedestre está obstruído por algum obstáculo na sua frente também foram identificados e demarcadas. Foram coletados aproximadamente 10 horas de vídeo obtidas de um AV em trânsito, com resolução de 640×480. Todos os principais parâmetros de avaliação estão resumidos na Tabela 1.

Técnicas	**BBbox (mAP)**	**Parâmetros**	**Tamanho**
Faster R-CNN	46.7	41.8M	159.7 MB
SSD300 7	25.1	35.6M	136.0 MB
RetinaNet	36.4	34.0M	130.3 MB
YOLO	37.2	7.2M	14 MB

Tabela 1: Parâmetros de Avaliação para a Detecção de Pedestres

O cenário de avaliação descrito acima utilizou um computador equipado com um processador Intel(R) Pentium(R) CPU G4560 com clock de 3,35 Ghz, 8 GB de memória RAM e uma placa de vídeo MSI GTX 1050 com 2 GB de VRAM. Os testes foram executados em uma duração aproximada de 1 a 2 horas, aproveitando a capacidade de processamento da placa de vídeo para otimizar os resultados.

Com o objetivo de avaliar os impactos do erro de localização dos pedestres detectados, as métricas de precisão de medida podem ser usadas para a analise, tais como Root Mean Squared Error (RMSE) e Mean Absolute Error (MAE). Tanto os resultados de RMSE quanto o MAE pertencem ao intervalo $[0, \infty]$. O RMSE e o MAE são métricas utilizadas para avaliar a precisão da detecção de objetos e são baseadas na comparação do valor predito ou adquirido pela técnica de detecção, como a posição, tamanho ou orientação de objetos, em relação ao valor real. O Mean-Square Error (MSE) é a média dos erros quadráticos, dada pela Equação 1. O RMSE é obtido através da raiz quadrada da média dos erros quadráticos, como observado pela Equação 2. O MSE e o RMSE por si só não são um bons indicadores de erro de localização de detecção dos objetos [Chai and Draxler 2014] e, portanto, o MAE seria uma métrica complementar para esse

propósito, dada pela Equação 3. Um menor resultado de RMSE e MAE significam uma melhor precisão de localização e um menor erro na detecção dos pedestres. Outra característica crítica a ser observada nessas métricas é que os valores de RMSE são iguais ou maiores do que MAE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \tag{2}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid Y_i - \hat{Y}_i \mid \tag{3}$$

As equações 1, 2 e 3, apresentadas acima, podem ser interpretadas da seguinte maneira: o símbolo n representa o número de dados, Y_i representa o valor real e $\hat{Y}_i$ representa o valor predito. No entanto, é importante saber que o modelo com o menor valor de RMSE e MAE não apresenta, necessariamente, o desempenho mais acurado em relação a outras métricas, como Intersection over Union (IoU). A métrica IoU, presente na Equação 4, é utilizada para avaliar a precisão de técnicas de detecção de objetos. Essa métrica mede a sobreposição entre a área detectada A_D e a área real do objeto A_{GT} nas imagens. O IoU é importante para avaliar a capacidade do detector de definir caixas delimitadoras para os objetos detectados nas imagens. IoU calcula a intersecção entre a caixa delimitadora detectada e a caixa delimitadora real do objeto, dividida pela união da soma das duas caixas delimitadoras. Dada pela Equação 4. Um limite de 50% foi definido para contabilizar imprecisões na avaliação do IoU. O limite é utilizado para delimitar objetos altamente convexos, por exemplo, uma pessoa com braços e pernas abertos [Everingham et al. 2010].

$$IoU = \frac{A_D \cap A_{GT}}{A_D \cup A_{GT}} \tag{4}$$

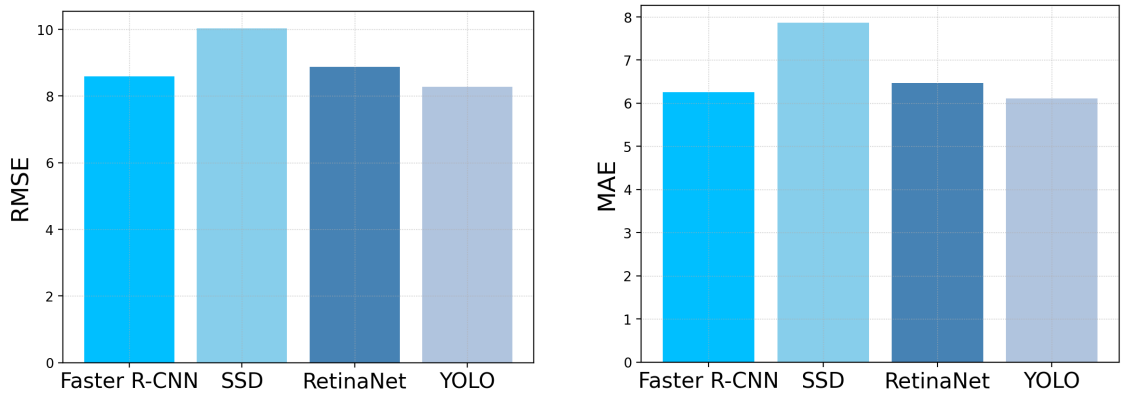
O tempo de detecção e processamento também é um fator importante a ser avaliado em cada uma das técnicas, uma vez que técnicas de detecção mais rápidas podem ser mais adequadas para aplicações de percepção dos AVs. É necessário que a detecção atenda os requisitos de latência dos AVs enquanto que mantenha a precisão e acurácia na localização e identificação dos objetos, além de evitar falsos positivos e garantir a confiabilidade dos resultados.

3.2. Análise dos Resultados

Figura 1a apresenta os resultados obtidos em relação ao valor de RMSE para técnicas YOLO, SSD, RetinaNet, Faster R-CNN. Cada dado apresentado na Figura 1a, refere-se a uma imagem do conjunto de dados Caltech analisado por cada técnica. O RMSE é baseado nos pedestres detectados em cada imagem. É possível Observar que o modelo SSD apresentou o maior valor de RMSE, indicando uma menor precisão nas detecções

em comparação com os outros modelos. O valor do RMSE obtido pelo SSD foi de 10.03 pixels. O RetinaNet obteve um RMSE de 8.88 seguido do Faster R-CNN que obteve um RMSE de 8.59, por fim, o YOLO obteve o menor valor de RMSE com 8.27 pixels, indicando que esse modelo apresenta uma maior precisão, com detecções mais próximas das posições reais, em relação aos outros modelos avaliados.

Na Figura 1b são apresentados os valores de MAE para técnicas YOLO, SSD, RetinaNet, Faster R-CNN. É possível observar que entre as técnicas avaliadas, o SSD apresentou o desempenho inferior comparado com as demais abordagens, com um erro de 7.87 pixels, em contraste com os resultados obtidos pelas demais abordagens. O YOLO obteve um MAE de 6.11 pixels, enquanto o Faster R-CNN e o RetinaNet obtiveram MAEs de 6.26 pixels e 6.47 pixels, respectivamente. Essa análise se torna complementar aos resultados apresentados na Figura 1a.



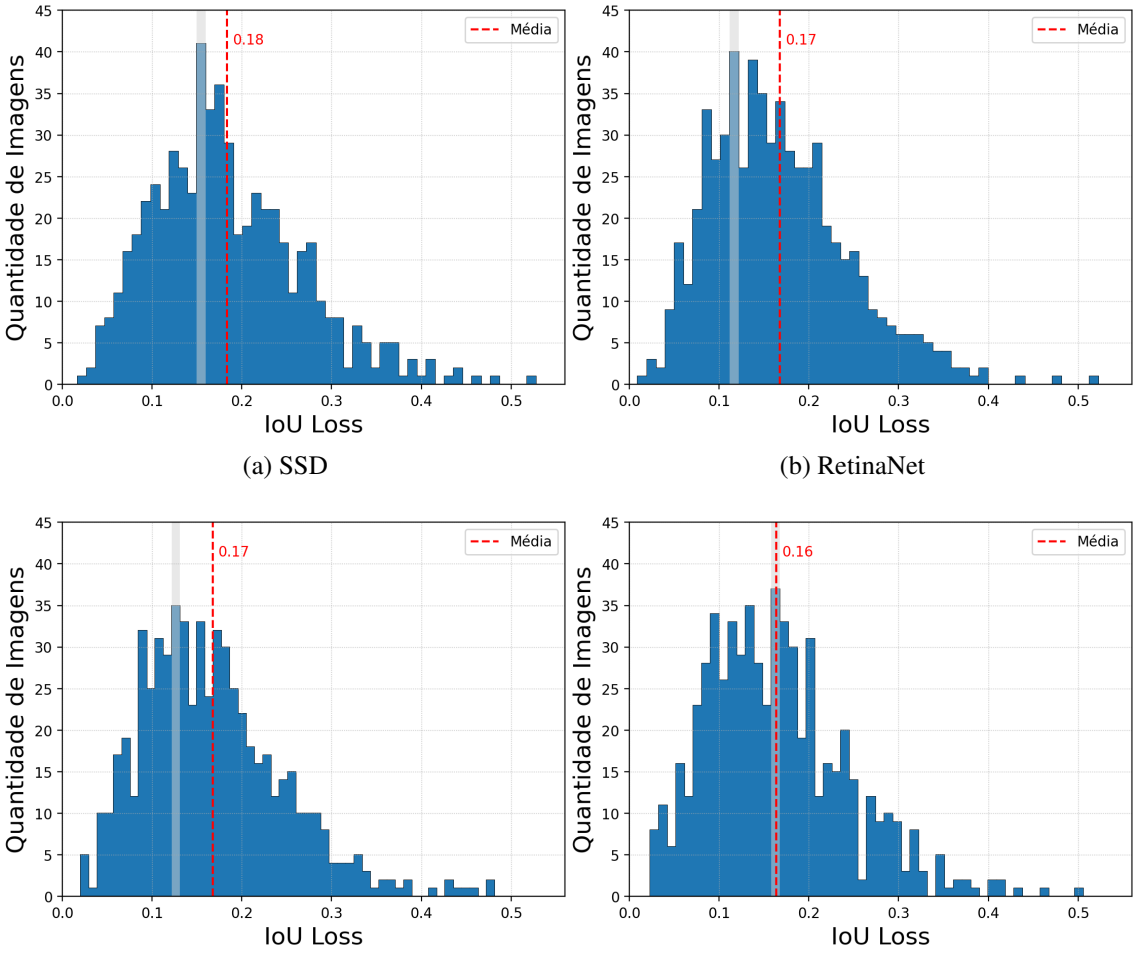
(a) Valores de RMSE médio

(b) Valores de MAE médio

Figura 1: RMSE e MAE para as técnicas de detecção de pedestres

A Figura 2 mostra o desempenho das diferentes técnicas de detecção utilizando a métrica da perda sobre o IoU. A quantidade de imagens se refere ao número de imagens do dataset Caltech utilizadas para teste, e o IoU Loss refere-se a perda sobre o IoU, quanto mais próximo de 1 menor é o desempenho. Pode-se observar entre as técnicas avaliadas que apesar de pouca diferença, o SSD apresentou o desempenho inferior, com uma perda média de 0.18, tendo a maior frequência dos dados um pouco acima de 0.15. As técnicas RetinaNet e Faster R-CNN apresentaram desempenho semelhante, com ambas apresentando uma perda média de IoU de 0.17 e com as maiores frequências de dados entre 0.12 à 0.13 de perda, enquanto o YOLO obteve a menor perda de IoU, ficando com a média de 0.16 de perda com a maior frequência também em 0.16. Essa análise é importante para avaliar a precisão das caixas delimitadoras preditas pelos modelos.

A Figura 3a apresenta o número total de pedestres detectados de acordo com as diferentes técnicas. O Faster R-CNN obteve o maior número de pedestres detectados, com uma taxa de acerto de 99,59%. O YOLO apresentou uma taxa de acerto de 99,43%, seguido do RetinaNet que obteve uma taxa de acerto de 99,23% e o SSD obteve a menor taxa de acerto, 80,27% e a maior taxa de falsos negativos. O SSD é mais impactado por imagens de pedestres que apresentam oclusões ou que possuem uma menor dimensão, visto que a técnica é otimizada para processar imagens com maior resolução, como visto na Figura 4 . O valor em porcentagem pode ser inferido tomando como base a quantidade



(a) SSD (b) RetinaNet

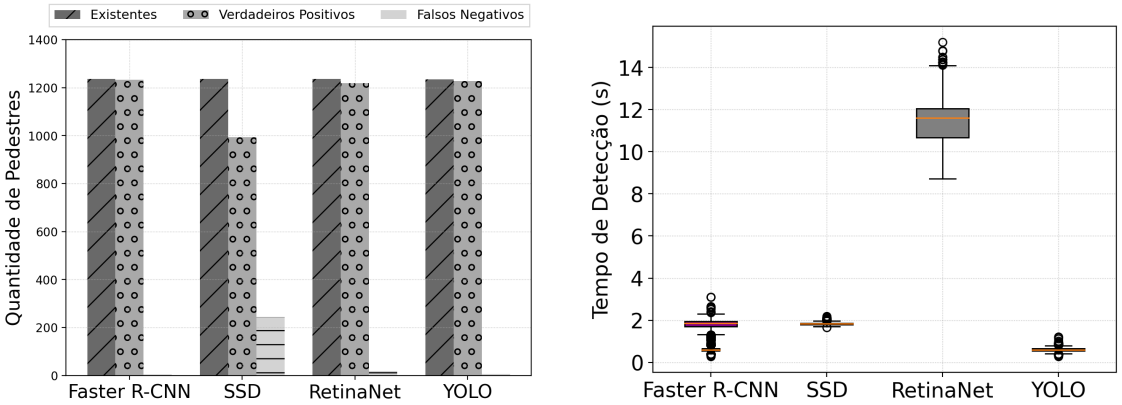
(c) Faster R-CNN (d) YOLO

Figura 2: Resultados de IoU para as técnicas de detecção de pedestre

total de pedestres existentes e a quantidade de verdadeiros positivos obtidos por cada técnica.

Na Figura 3b apresenta os resultados de tempo de detecção para as técnicas YOLO, SSD, RetinaNet, Faster R-CNN. Pode-se analisar que os valores apresentados mostram que o YOLO obteve o menor tempo de detecção de pedestres com uma média de 0,58 segundos. O maior tempo médio de detecção identificado na Figura 3b é de 11,60 segundos, da técnica RetinaNet. Além disso, é possível identificar um comportamento similar para as técnicas SSD e Faster R-CNN, com tempo de detecção médio de 1,83 e 1,85 segundos, respectivamente.

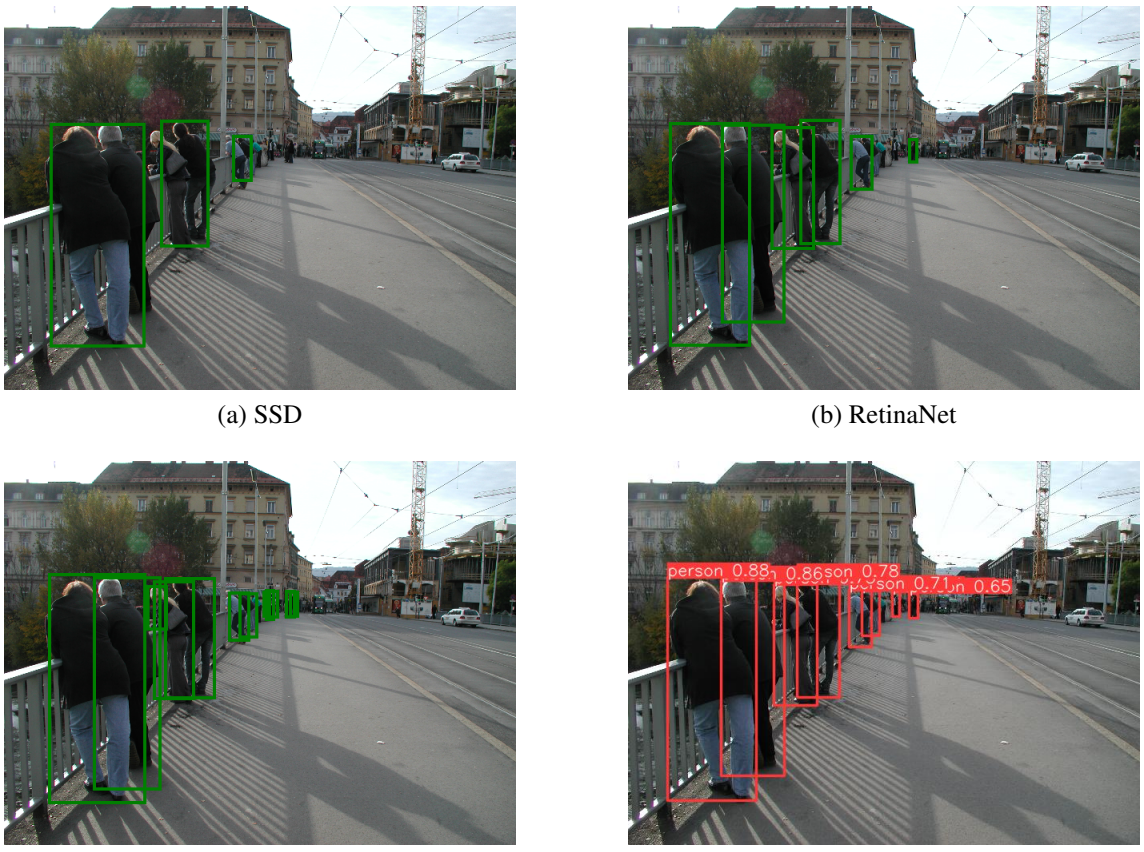
Com base na análise das avaliações realizadas, foi possível concluir que as técnicas YOLO e Faster R-CNN provêm localizações com baixos erros em termos de RMSE e MAE, isso porque o YOLO e o Faster R-CNN trabalham com várias propostas de regiões, o que pode ser muito eficaz detecção de objetos. O RetinaNet faz o uso de uma abordagem de detecção baseada em caixas delimitadoras pré definidas em diferentes escalas, porém, utiliza uma função de perda focal para lidar com o problema de desequilíbrio de categoria na detecção de objetos, o que o torna eficiente em detectar objetos menos



(a) Quantidade total de detecções (b) Tempo de detecção

Figura 3: Quantidade e tempo de detecção para cada técnica avaliada

frequentes. Já o SSD utiliza uma abordagem de múltiplas caixas delimitadoras para detectar objetos, o que pode levar a uma menor precisão nas detecções, especialmente em objetos menores.



(a) SSD (b) RetinaNet

(c) Faster R-CNN (d) YOLO

Figura 4: Predição feita por cada uma das técnicas

Além do RMSE e MAE, foi analisada a perda sobre o IoU, em que o YOLO obteve um desempenho superior às demais técnicas, devido ao fato de o YOLO utilizar

caixas delimitadoras dinâmicas e o conceito de Spatial Pyramid Pooling (SPP), que é uma técnica de processamento de imagem que reduz a resolução espacial. Isso é feito através da aplicação de filtros de diferentes tamanhos para melhorar a precisão e o desempenho em objetos pequenos. O desempenho do SSD foi inferior devido à utilização de caixas delimitadoras pré-definidas. Isso pode resultar em perdas sobre o IoU maiores do que o esperado, uma vez que as caixas delimitadoras não são adaptadas de forma adequada para cada objeto específico na imagem. O tempo de processamento também é outro fator fundamental em termos de detecção de pedestres, o YOLO obteve o menor valor em relação as outras abordagens avaliadas, com uma média de 0,58 segundos, isso porque o YOLO realiza a detecção em uma única passagem, o que o torna rápido. Em contra partida o RetinaNet obteve um resultado inferior com uma média de 11 segundos, isso ocorre porque a técnica RetinaNet usa uma abordagem de detecção densa de objetos, onde é avaliado múltiplas regiões da imagem para cada objeto, em vez de avaliar regiões predefinidas. Enquanto o YOLO usa uma abordagem baseada em grade que divide a imagem em várias células e cada célula é avaliada. As técnicas Faster R-CNN e SSD obtiveram resultados próximos em termos de tempo de detecção, apesar da diferença de tempo ser próxima à do YOLO. Tratando-se de AVs, o menor tempo de detecção possível é essencial para a tomada de decisão. Os resultados indicam que a técnica YOLO é a alternativa viável para realizar a detecção acurada dos pedestres dentro do contexto de AVs. Todos os resultados das avaliações feitas estão resumidos na Tabela 2

Técnica	**RMSE**	**MAE**	**IoU Loss**	**Eficiência**	**Tempo**
SSD	16.02	7.87	0.18	80,27%	1,12 segundos
RetinaNet	15.78	6.46	0.17	99,23%	1,04 segundos
Faster R-CNN	15.62	6.25	0.17	99,59%	0,86 segundos
YOLO	15.20	6.11	0.16	99,43%	0,58 segundos

Tabela 2: Resultados de Avaliação para a Detecção de Pedestres

4. Conclusões e Trabalhos Futuros

Este artigo apresenta um estudo comparativo com diferentes técnicas de detecção de pedestres para aplicações de percepção dos AVs. Os resultados da avaliação destacam a importância de escolher a melhor abordagem para detectar objetos móveis ou pedestres, a fim de maximizar a percepção dos AVs no cenário urbano. Devido ao cálculo do IoU, usado para medir o desempenho das técnicas de detecção de objetos, foi possível observar que apesar das técnicas avaliadas não terem apresentado grandes diferenças, uma vez que as perdas médias de IoU foram próximas entre elas, o YOLO se destaca e surge como alternativa adequada para resolver problemas como este, embora tenha obtido uma perda média de IoU ligeiramente menor em comparação com RetinaNet e Faster R-CNN, a diferença pode ser considerada relevante. Além disso, o fato de o YOLO ter a maior frequência de dados na mesma faixa de perda média (0.16) indica que ele apresentou uma consistência maior em predições com alta sobreposição entre as caixas delimitadoras. Os resultados da avaliação mostram a eficiência do YOLO em comparação com Faster R-CNN, SSD e RetinaNet, tanto na redução do erro de detecção em termos de RMSE e MAE, quanto no tempo de detecção. No entanto, é fundamental considerar outras técnicas

de detecção de objetos e considerar o tempo de processamento na tomada de decisão dos AVs, além de desenvolver mecanismos para utilizar o melhor de cada técnica de detecção dentro do contexto veicular.

Como trabalho futuro, pretende-se propor um mecanismo que detecte o melhor momento para utilizar cada técnica no cenário urbano. Também pretendemos propor um protocolo de percepção cooperativa para escolher o melhor Connected and Autonomous Vehicles (CAVs) para fazer a transmissão dos pedestres detectados e avaliar o desempenho em ambientes realistas.

Agradecimentos

Os autores gostariam de agradecer à Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), pelas bolsas #2015/24494-8 e #2019/19105-3. Também à PPI-Softex, com o apoio do MCTI [01245.013778/2020-21].

Referências

Chai, T. and Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250.

Dollar, P., Wojek, C., Schiele, B., and Perona, P. (2011). Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743–761.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*.

Garlichs, K., Wegner, M., and Wolf, L. C. (2018). Realizing collective perception in the artery simulation framework. In *IEEE Vehicular Networking Conference (VNC)*, pages 1–4. IEEE.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2015). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448. arXiv.

Hussain, R. and Zeadally, S. (2018). Autonomous cars: Research results, issues and future challenges. *IEEE Communications Surveys & Tutorials*.

Kim, J.-a., Sung, J.-Y., and Park, S.-h. (2020). Comparison of faster-rcnn, yolo, and ssd for real-time vehicle type recognition. In *2020 IEEE international conference on consumer electronics-Asia (ICCE-Asia)*, pages 1–4.

Krijestorac, E., Memedi, A., Higuchi, T., Ucar, S., Altintas, O., and Cabric, D. (2020). Hybrid vehicular and cloud distributed computing: A case for cooperative perception. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE.

Liang, M., Yang, B., Wang, S., and Urtasun, R. (2018). Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 641–656.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2020). Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327.

Liu, S., Tang, J., Zhang, Z., and Gaudiot, J.-L. (2017). Computer architectures for autonomous driving. *Computer*, 50(8):18–25.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD: Single shot MultiBox detector. In *Computer Vision – ECCV 2016*, pages 21–37. Springer International Publishing.

Lobato, W., Mendes, P., Rosário, D., Cerqueira, E., and Villas, L. A. (2023). Redundancy mitigation mechanism for collective perception in connected and autonomous vehicles. *Future Internet*, 15(2):41.

maintainers, T. and contributors (2016). Torchvision: Pytorch's computer vision library. *GitHub repository*.

Pendleton, S., Andersen, H., Du, X., Shen, X., Meghjani, M., Eng, Y., Rus, D., and Ang, M. (2017). Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1):6.

Rahmad, C., Asmara, R. A., Putra, D. R. H., Dharma, I., Darmono, H., and Muhiqqin, I. (2020). Comparison of viola-jones haar cascade classifier and histogram of oriented gradients (hog) for face detection. *IOP Conference Series: Materials Science and Engineering*, 732(1):012038.

Rasouli, A. and Tsotsos, J. K. (2020). Autonomous vehicles that interact with pedestrians: A survey of theory and practice. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):900–918.

Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE Computer Society.

Saleh, K., Hossny, M., and Nahavandi, S. (2020). Spatio-temporal densenet for real-time intent prediction of pedestrians in urban traffic environments. *Neurocomputing*, 386:317–324.

Tan, M. and Le, Q. V. (2021). Efficientnetv2: Smaller models and faster training.

Vora, S., Lang, A. H., Helou, B., and Beijbom, O. (2020). Pointpainting: Sequential fusion for 3d object detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4603–4611, Los Alamitos, CA, USA. IEEE Computer Society.

Wang, J., Liu, J., and Kato, N. (2018). Networking and communications in autonomous driving: a survey. *IEEE Communications Surveys & Tutorials*.

Xu, W., Zhou, H., Cheng, N., Lyu, F., Shi, W., Chen, J., and Shen, X. (2017). Internet of vehicles in big data era. *IEEE/CAA Journal of Automatica Sinica*, 5(1):19–35.

Yang, Q., Fu, S., Wang, H., and Fang, H. (2021). Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities. *IEEE Network*, 35(3):96–101.

Ye, M., Shen, J., Lin, G., Xiang, T., Shao, L., and Hoi, S. C. (2021). Deep learning for person re-identification: A survey and outlook. *IEEE transactions on pattern analysis and machine intelligence*, 44(6):2872–2893.