

Identificação de Veículos em Estacionamentos: Uma Abordagem Comparativa Entre Diferentes Versões do Modelo YOLO

Relatório de Formalização

Matheus de S. M. da Silva

RA: 140317

Universidade Federal de São Paulo

São José dos Campos-SP, Brasil

souza.matheus@unifesp.br

Marcos L. da Silva

RA: 140782

Universidade Federal de São Paulo

São José dos Campos-SP, Brasil

marcos.lucas@unifesp.br

I. RESUMO

O presente artigo tem o principal objetivo de apresentar uma abordagem comparativa de eficiência entre diferentes versões do YOLO (um modelo de detecção de objetos em imagens) em um sistema de Inteligência Artificial para a identificação de veículos em vagas do estacionamento do ICT-Unifesp. Serão descritos os principais conceitos de IA utilizados para as etapas de treino e teste do modelo, bem também o método de validação.

II. INTRODUÇÃO E MOTIVAÇÃO

A transformação digital está remodelando todas as áreas da sociedade, oferecendo inúmeras oportunidades para melhorar a eficiência de operações cotidianas, a segurança e a conveniência de processos que antes eram considerados monótonos ou complexos. No contexto de estacionamentos, uma das áreas de foco que ainda não foi completamente explorada é o uso da visão computacional para automatizar a gestão de veículos.

Este projeto nasce da motivação de aplicar os conceitos aprendidos na disciplina de Inteligência Artificial (IA) para resolver problemas práticos e relevantes. Embora o estacionamento do ICT Unifesp não seja frequentemente lotado, a abordagem proposta pode ser facilmente adaptada para outros ambientes com maior fluxo de veículos.

A visão computacional, uma subdisciplina da IA, oferece um caminho promissor para a identificação de veículos e o monitoramento de vagas de estacionamento em tempo real. Este projeto propõe a implementação de uma solução que, ao ser integrada ao estacionamento da ICT Unifesp, permitirá aos motoristas avaliar a disponibilidade de vagas antes mesmo de se dirigirem ao local. Dessa forma, contribuirá para a otimização do fluxo de veículos, aumento da segurança e prevenção de congestionamentos no estacionamento.

A relevância deste projeto reside na sua aplicabilidade prática e no seu potencial para demonstrar o quão poderosa a IA pode ser quando aplicada a problemas do mundo real.

Embora o estudo se concentre no estacionamento da ICT Unifesp, a metodologia proposta pode ser generalizada e adaptada para outros ambientes, ampliando o impacto e a relevância dos resultados obtidos.

Além disso, é importante ressaltar que a motivação para este projeto vai além da resolução de um problema prático. É também uma oportunidade para os estudantes colocarem em prática os conhecimentos adquiridos durante a disciplina de IA, explorando a visão computacional e contribuindo para o avanço da ciência e da tecnologia. Ao longo do projeto, serão adquiridas habilidades valiosas em IA e visão computacional, que poderão ser aplicadas em futuros projetos e em diversas áreas de pesquisa e desenvolvimento.

III. CONCEITOS FUNDAMENTAIS

Para a compreensão plena deste projeto, é necessário entender alguns conceitos fundamentais. Em particular, a estrutura e o funcionamento do YOLO, e o procedimento de treinamento de um modelo de Inteligência Artificial.

A. YOLO

Lançado em 2015 por Joseph Redmon e Ali Farhadi, o modelo YOLO (You Only Look Once) é uma técnica inovadora para a detecção de objetos em imagens e vídeos. O seu grande diferencial é a velocidade de detecção superior em relação a outros modelos que possuem o mesmo propósito (enquanto o tempo de processamento de uma imagem dos modelos mais precisos é de aproximadamente 0.5s, o do YOLO é de 0.05s com a mesma precisão) e também o fato dele ser totalmente open source e livre de licenças de uso.

Diferente de outros tipos de algoritmos, o YOLO utiliza o método de *single pass* para detecção dos objetos, isto é, analisa toda a imagem apenas uma única vez ao invés de procurar em diferentes localizações e escalas, permitindo uma maior rapidez no processo e uma precisão similar a de outros modelos.

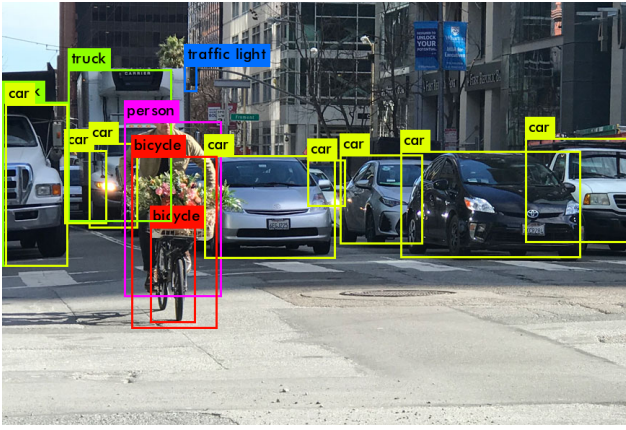


Figura 1. Representação de detecção de imagens com YOLO (Fonte: Zhang Haduo's Site)

Na detecção de objetos, uma imagem é analisada e a presença de objetos específicos é identificada, juntamente com suas localizações. No caso deste projeto, os "objetos" serão os espaços de estacionamento (podendo este ser vazio ou não) e os automóveis presentes no estacionamento.

B. Treinamento e Teste de Modelo

Em machine learning, *holdout* refere-se a uma técnica de validação de modelo. Também conhecida como "conjunto de validação", a abordagem *holdout* envolve a divisão do conjunto de dados disponíveis em dois subconjuntos distintos: um conjunto de treinamento e um conjunto de teste.

O treinamento é o processo de aprendizado em que o modelo aprende a partir de uma série de dados. No contexto do YOLO, o treinamento envolve a apresentação de uma série de imagens ao modelo, tendo cada uma delas informações associadas que indicam a localização dos objetos de interesse (neste caso, espaços de estacionamento vazios). Então o modelo aprende a identificar e localizar esses objetos a partir dessas imagens.

Primeiramente, o modelo é inicializado com pesos aleatórios para então, a cada imagem apresentada, fazer uma previsão de onde acredita que os objetos estão. O erro dessa previsão é calculado comparando a previsão do modelo com a localização real do objeto. Em seguida, o modelo ajusta seus pesos para minimizar o erro. Repetindo esse processo muitas vezes com diferentes imagens, o modelo aprende a detectar e localizar corretamente os objetos.

Para a avaliação de cada um dos modelos YOLO que serão utilizados na etapa de validação, será utilizada a técnica *Cross-validation*, que envolve a divisão dos dados em vários subconjuntos menores e a execução repetida do treinamento e teste em diferentes combinações desses subconjuntos. Ela é útil quando o conjunto de dados é limitado e você deseja obter estimativas mais robustas do desempenho do modelo. O protocolo usado neste projeto será o *2-Fold Crossvalidation*, onde este dividirá o modelo em 2 partes, reservando cerca de 50% dos dados para treinamento e o restante para teste de validação.

C. Aplicação do YOLO

Uma vez treinado, o modelo YOLO pode ser usado para analisar imagens do estacionamento em tempo real, identificando quais espaços de estacionamento estão ocupados e quais estão vazios. As vagas foram divididas em dois tipos: vaga normal e vaga de acessibilidade. Isso possibilita um monitoramento eficiente e automático de cada tipo de vaga disponível no estacionamento.

Serão utilizadas diferentes versões do YOLO na análise dos resultados para fins comparativos. Tais versões incluem: YOLOv5n, YOLOv5s, YOLOv8n e YOLOv8s. Para fins de comparação, a medida de avaliação do desempenho dos modelos será a *mAP* (mean Average Precision).

IV. TRABALHOS RELACIONADOS

A. Sistema de Visão Inteligente de Baixo Custo Para Parque de Estacionamento

Esta dissertação de mestrado propõe um sistema de visão computacional de baixo custo baseado no Raspberry Pi 2 e numa webcam HD para a detecção de vagas livres em estacionamentos. O algoritmo de visão é baseado em um detector de arestas e um limite dinâmico atribuído a cada vaga para inferir sobre o estado da mesma. O sistema foi testado em ambiente real com diferentes condições atmosféricas e oclusões parciais entre os veículos. O algoritmo funciona em tempo real e tem uma taxa de sucesso de 99,6%.

Este trabalho é relevante para o nosso projeto, pois apresenta uma abordagem de baixo custo para a detecção de vagas de estacionamento utilizando visão computacional.

B. Aplicação de Algoritmos vSLAM de Reconhecimento de Objetos Para Classificação de Veículos e Vagas de Estacionamento em Ambientes Externos Utilizando um Drone

O projeto feito por Pedro Henrique de Oliveira Garcia e Roberto Scalco propõe a utilização de técnicas de visão computacional e classificação de imagens obtidas pela câmera monocular de um drone para localização e classificação de veículos e vagas em estacionamentos em campo aberto. A técnica de mapeamento e localização simultâneas, vSLAM (visual Simultaneous Localization and Mapping), possibilita que o drone obtenha imagens dos estacionamentos contendo sua informação de localização espacial. A rede neural profunda YOLO foi aplicada para realizar a localização dos veículos e das vagas, permitindo determinar se as vagas estavam ocupadas ou não.

Este é relevante para nosso projeto, pois apresenta uma abordagem similar de utilização de visão computacional e algoritmos de aprendizado de máquina para a identificação e classificação de veículos em estacionamentos.

C. A Smart Parking System: An IoT Based Computer Vision Approach for Free Parking Spot Detection Using Faster R-CNN with YOLOv3 Method

Publicado na revista *Wireless Personal Communications*, o artigo aborda a problemática do estacionamento em grandes cidades e propõe um sistema inteligente de estacionamento

para designar um local de estacionamento livre para as pessoas que necessitam de uma vaga. O sistema é capaz de processar imagens da área de estacionamento e seus espaços livres em tempo real, notificando o usuário sobre todas as vagas disponíveis para estacionamento. A detecção do local de estacionamento é implementada usando Redes Neurais Convolucionais Recorrentes Mais Rápidas (Faster R-CNN) com a técnica YOLOv3. O sistema é robusto, eficiente em termos de energia e tem a competência para futuras melhorias.

Este artigo é relevante para o nosso projeto, pois apresenta uma abordagem de utilização de visão computacional e algoritmos de aprendizado de máquina para a detecção de vagas de estacionamento.

D. Parking Space Detection Using Deep Learning

Esta dissertação aborda o desafio da detecção de vagas de estacionamento nas cidades, onde os motoristas perdem tempo procurando por uma vaga livre. O estudo compara o desempenho dos algoritmos de aprendizado profundo M-RCNN e YOLO, utilizados para classificar imagens. O YOLO foi considerado o melhor para este estudo, pois foi capaz de funcionar com recursos computacionais de baixo custo e fornecer previsões precisas. Portanto, foi utilizado para desenvolver um protótipo usado para detectar o status das vagas de estacionamento como vazias ou ocupadas. O modelo alcançou uma precisão de 92,6% na detecção do status da vaga de estacionamento. Os experimentos mostraram que o modelo proposto pode ser usado para alcançar a detecção automatizada do status da vaga em qualquer área de estacionamento demarcada.

Esta dissertação é relevante para o nosso projeto, pois apresenta uma abordagem de utilização de algoritmos de aprendizado profundo para a detecção de vagas de estacionamento.

E. Smart Parking with Computer Vision and IoT Technology

Este artigo aborda os problemas enfrentados pelos estacionamentos em grandes cidades, incluindo a dificuldade em localizar uma vaga de estacionamento livre, a segurança do veículo estacionado e as pessoas estacionando em uma vaga reservada. O estudo propõe o uso combinado de aplicativo móvel, visão computacional e tecnologias IoT para enfrentar esses problemas. O algoritmo YOLO foi usado para desenvolver um protótipo que foi usado para detectar o status das vagas de estacionamento como vazias ou ocupadas.

Este artigo é relevante para o nosso projeto, pois apresenta uma abordagem de utilização de visão computacional e tecnologias IoT para a detecção de vagas de estacionamento.

F. T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks

Aqui é proposto um modelo de detecção de objetos profundo, leve e modificado, sendo ele baseado na arquitetura YOLO-v5. O modelo pode detectar objetos grandes, pequenos e minúsculos. Especificamente, propõe-se o uso de um mecanismo multi-escala para aprender representações de recursos discriminativos profundos em diferentes escalas e determinar

automaticamente as escalas mais adequadas para detectar objetos em uma cena (ou seja, no nosso caso, veículos). O módulo multi-escala proposto reduz o número de parâmetros treináveis em comparação com a arquitetura original YOLO-v5. Os resultados experimentais também demonstram que o módulo obteve a precisão melhorada por uma grande margem (96,34% contra 63,87%).

Tal proposto é relevante para o nosso projeto, pois apresenta uma abordagem de utilização de visão computacional e algoritmos de aprendizado profundo para a detecção de veículos.

G. Implementation of Smart Parking System Using Image Processing

O sistema proposto por Amarasooriya (2022) é capaz de identificar o tipo de veículo e os espaços de estacionamento livres. Além disso, o sistema contém um algoritmo de caminho mais curto para auxiliar os motoristas a encontrar o espaço de estacionamento vago mais próximo. O sistema utiliza uma câmera como sensor e o algoritmo YOLO para detectar a presença e o tipo de veículos na área de estacionamento. A precisão do sistema foi de aproximadamente 95% em diferentes condições climáticas e vários ângulos de imagens.

Este trabalho é especialmente relevante para o nosso projeto devido ao seu uso do algoritmo YOLO para detecção de veículos.

H. Melhoramento do estacionamento público

A dissertação de Luís Miguel Amorim Albuquerque, orientada por A. J. Soares e Luís Jorge Lima Ferrás, propõe uma solução completa e genérica para a gestão de estacionamentos de veículos. A solução pode ser implementada em qualquer cidade, pois é capaz de se adaptar a diferentes necessidades e especificidades, através do uso de IA ou de leitura remota de dados. A solução apresentada consiste na elaboração de um sistema de detecção de lugares livres, um servidor, uma aplicação para celular e um back office. O sistema de detecção é opcional, sendo somente instalado quando se pretende ter acesso ao número de lugares vagos em tempo real. A aplicação para celular permite que os utilizadores consigam ter acesso às informações dos estacionamentos disponíveis, indiquem as suas preferências e o seu destino e, posteriormente, que recebam sugestões de estacionamento por parte do servidor.

Esta dissertação é relevante para o nosso projeto, pois apresenta uma abordagem de utilização de visão computacional e algoritmos de aprendizado profundo para a detecção de vagas de estacionamento.

I. Deep Learning-Based Parking Occupancy Detection Framework Using ResNet and VGG-16

O artigo de Thakur, Bhattacharjee, Jain, Acharya e Hu (2023) destaca a necessidade de crescente pesquisa e desenvolvimento em sistemas de gestão de estacionamento para fornecer indicações em tempo real da ocupação de espaços de estacionamento internos e externos. O desafio principal tem sido desenvolver métodos de detecção acessíveis baseados em imagens para substituir as técnicas mais caras baseadas em

sensores implantadas em ambientes internos. Com o avanço da visão computacional e do aprendizado profundo, os autores visam aproveitar o desempenho notável das redes neurais convolucionais para realizar tarefas de reconhecimento de categoria de imagem para desenvolver um framework robusto de detecção de ocupação de estacionamento. A precisão relatada com o modelo ResNet e a máquina de vetores de suporte foi de 98,9%. O modelo usando VGG16 relatou uma precisão de 93,4%.

Este artigo é relevante para o nosso projeto porque apresenta um quadro para a detecção de ocupação de estacionamento com base em imagens, que é o problema principal que estamos tentando resolver. A abordagem dos autores combina visão computacional e aprendizado profundo, duas técnicas que também planejamos usar.

J. Optimized Real-Time Parking Management Framework Using Deep Learning

Este artigo descreve a utilização do YOLOv5, uma arquitetura de rede neural convolucional, para a detecção de veículos em vez da classificação de vagas de estacionamento. O modelo foi avaliado usando o conjunto de dados PKLot, obtendo uma precisão de 99,5%. Além disso, um modelo pré-treinado do YOLO v5 no conjunto de dados MS COCO foi utilizado para detectar e atribuir vagas de estacionamento livres, gerando estatísticas de veículos. O desempenho do sistema proposto foi avaliado em um conjunto de dados personalizado, alcançando uma precisão de 96,8% e um desempenho quase em tempo real de 45 FPS, o que torna o sistema mais eficiente, escalável e generalizado.

Este artigo é importante para o nosso projeto porque apresenta a utilização do YOLOv5, uma arquitetura de rede neural convolucional, para a detecção de veículos, o que é um componente chave da nossa proposta de identificação de vagas de estacionamento. Os resultados mostram uma alta precisão na detecção de veículos e na atribuição de vagas de estacionamento livres. Além disso, o desempenho em tempo real do sistema proposto é promissor, o que sugere a viabilidade de aplicar a solução em um ambiente real de estacionamento.

V. OBJETIVO

O principal objetivo deste projeto é o desenvolvimento e a implementação de um sistema de identificação e contabilização de vagas de estacionamento, utilizando técnicas avançadas de IA (como o treinamento do modelo YOLO, por exemplo). O sistema será projetado para monitorar o estacionamento do ICT-Unifesp, identificando os veículos presentes e as vagas disponíveis. Serão utilizadas diferentes versões do modelo YOLO na aplicação desse sistema e, posteriormente, os resultados de cada uma das versões aplicadas serão apresentados para fins de comparação entre os mesmos.

Espera-se que esse sistema proporcione uma série de benefícios, incluindo a otimização da utilização do estacionamento, a melhoria do fluxo de tráfego e a ampliação da segurança. Embora o foco inicial esteja no estacionamento

do ICT Unifesp, o sistema poderá ser adaptado para outros contextos, como shoppings ou estacionamentos públicos de grandes cidades.

VI. METODOLOGIA EXPERIMENTAL

A metodologia para este projeto envolverá várias etapas distintas, incluindo a preparação dos dados, o treinamento do modelo, validação e a avaliação do sistema (observe a Figura 2).

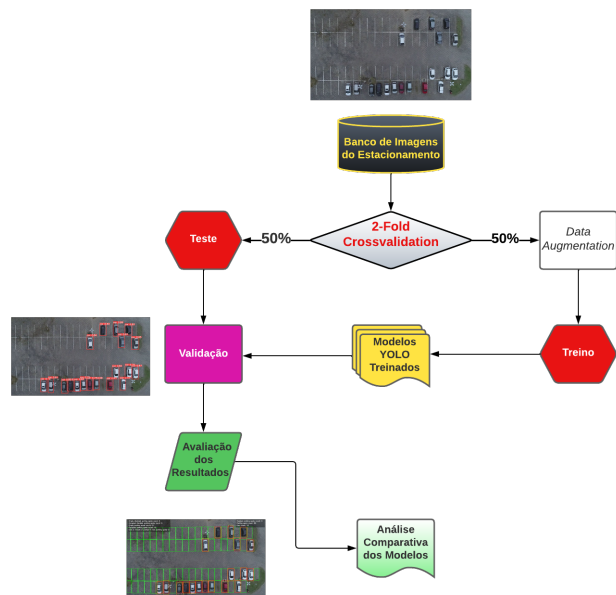


Figura 2. Pipeline Experimental do Projeto

A. Base de Dados e Tratamento das Imagens

Inicialmente, na etapa de preparação dos dados, foram utilizadas 23 imagens diferentes do estacionamento do ICT-Unifesp para treinar o modelo. Estas imagens foram tratadas e formatadas adequadamente para que possam ser usadas com o modelo YOLO. Elas foram capturadas através de um drone e possuem resolução Full HD, isto é, são de 1920x1080 pixels.

Na rotulização dos carros, foram feitas as bounding-boxes dos veículos na ferramenta TrainYOLO, disponível online. Para a contabilização das vagas do estacionamento, é feita a delimitação dos bounding-boxes de cada vaga através do software Label Studio.

Para a predição de um objeto do tipo carro, é utilizado o conceito de *Intersection over Union(IoU)*. Considerando a áreas e as posições das bounding-boxes entre os carros preditos e os verdadeiros, o cálculo do *IoU* é feito da seguinte forma:

$$IoU = \frac{\text{Área de Sobreposição}}{\text{Área de União}}$$

As vagas serão de duas categorias (normal e de acessibilidade) e podem ter dois estados (disponível ou ocupada). A classificação de estado das vagas é feita com base em

Intersection over Union (IoU) Threshold = 0.5

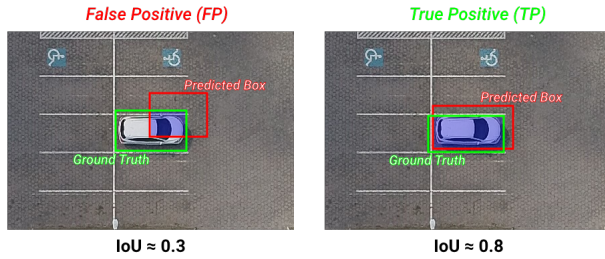


Figura 3. IoU das Boudling-Boxes

um limiar mínimo *threshold* de 40% da intersecção entre a *bounding-box* do veículo e da vaga:

- Se a intersecção for maior ou igual ao *threshold*, a vaga estará como ocupada (contorno vermelho);
- Caso contrário, a vaga estará como disponível (contorno verde).

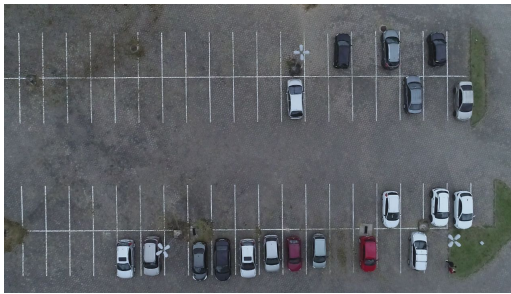


Figura 4. Exemplo de imagem do estacionamento sem rótulos

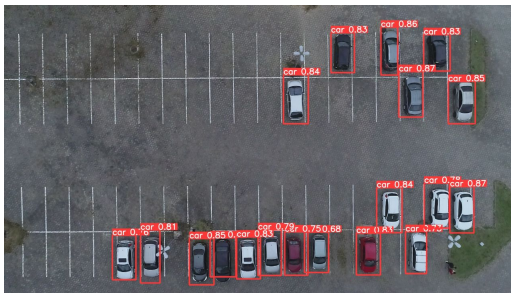


Figura 5. Exemplo de imagem do estacionamento com veículos reconhecidos pelo modelo YOLO

B. Treinamento e Protocolo de Validação

Para a etapas de treinamento e teste de cada um dos modelos *YOLO*, será utilizado o protocolo *2-Fold Crossvalidation*, onde o conjunto da base de dados será dividido em 2 partes: cerca de 50% para treino e o restante para teste. Durante o treinamento, o modelo aprenderá a identificar os carros presentes na imagem.

Nas primeiras tentativas de treino, os modelos *YOLO* apresentaram bons resultados de *mAP* e *recall*, mas isso considerando que a maior parte dos veículos estão posicionados

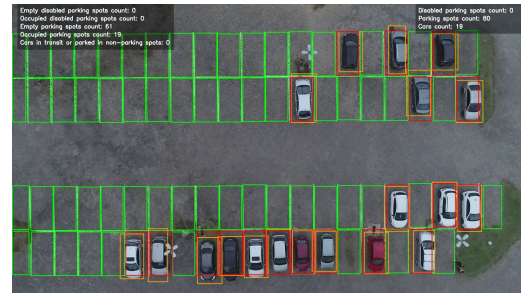


Figura 6. Exemplo de imagem do estacionamento rotularizada

horizontal ou verticalmente na imagem. Visto que as imagens iniciais da base de dados apresentavam poucos veículos posicionados na diagonal, poderia haver a grande possibilidade de falhas na identificação dos veículos nessa posição pelos modelos *YOLO*. Por isso, antes do das etapas principais de treinamento, foi aplicada a técnica *Data Augmentation* na partição dos dados de treino, que consiste em aumentar a quantidade de dados através dos dados já existentes realizando algumas modificações na imagens. Neste caso, a quantidade de dados em cada partição de treino foi triplicada, rotacionando as imagens em 30° e 60°. Essa etapa é fundamental para melhorar o aprendizado e prever com mais precisão eventuais cenários atípicos que podem ser apresentados pelos dados (um carro na diagonal, por exemplo).



Sem
Data Augmentation



Com
Data Augmentation

Figura 7. Imagem de comparação do reconhecimento dos carros antes e depois de aplicar o Data Augmentation

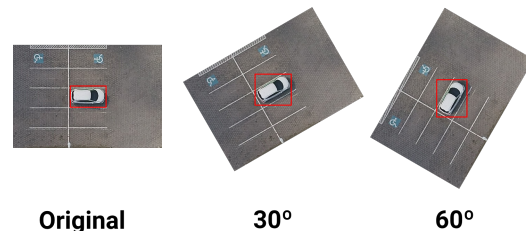


Figura 8. Imagens do base de dados rotacionadas

Todos os modelos YOLO foram treinados (em todos os *folds*) sob os seguintes parâmetros uma única vez:

- batch = 6;
- imgsz = 1920;
- epochs = 200;
- iou = 0,5.

A plataforma usada para o treinamento é a *trainYOLO*, e as versões do YOLO usadas no treinamento são *v5n*, *v5s*, *v8n* e *v8s*. Todos os modelos foram treinados através Google Colab para aproveitar o poder de processamento da *GPU NVIDIA Tesla T4*. Além disso, considerando que a rotação das imagens altera a área dos objetos vertical e horizontalmente, as *bounding-boxes* que envolvem estes objetos tiveram que ser redimensionadas, pois elas se mantêm na mesma orientação (note na Figura 8).

VII. RESULTADOS E DISCUSSÃO

Após a etapa de treinamento, foram extraídas algumas informações quantitativas de cada um dos quatro modelos YOLO usados nesse projeto com o intuito de realizar uma abordagem comparativa.

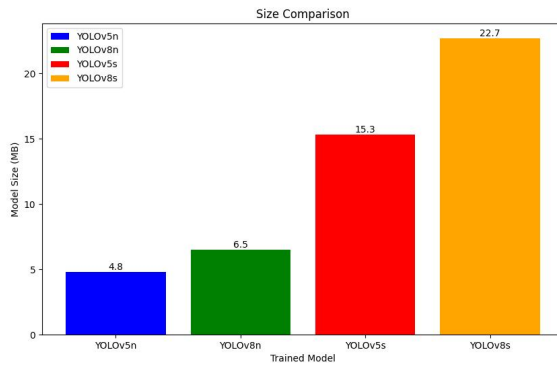


Figura 9. Comparação do tamanho dos modelos (em Megabytes)

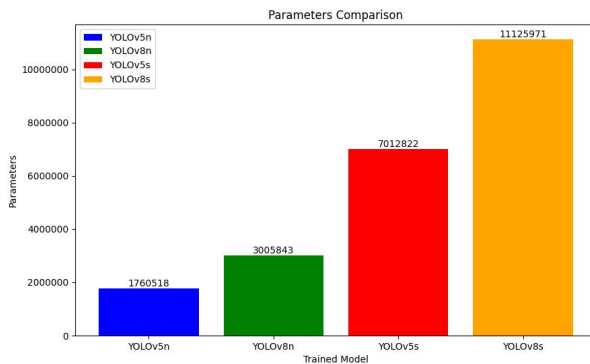


Figura 10. Quantidade de parâmetros de cada modelo

Nas Figuras 9 e 10, pode-se perceber uma grande diferença do tamanho (em Megabyte) nos modelos da versão *nano*

(*YOLOv5n* e *YOLOv8n*) em comparação com os modelos da versão *short* (*YOLOv5s* e *YOLOv8s*), onde os correspondentes ao *short* são mais que o triplo do tamanho dos correspondentes ao *nano*. Isso ocorre devido a quantidade de parâmetros considerada por cada versão, tendo uma relação diretamente proporcional com o tamanho do arquivo do modelo. Este resultado era esperado, pois a versão *short* é de complexidade maior que a *nano*.

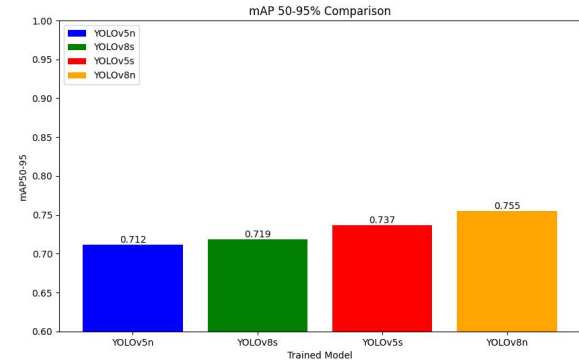


Figura 11. mAP 50-95% de cada modelo

Considerando que foi usado o protocolo de validação cruzada dividindo o modelo em cerca de 50% para treino e o restante para teste, obteve-se a como *holdout* a média da métrica *mAP* entre o fold 0 e o fold 1 para cada modelo (resultados são apresentados na Figura 11). Nela, é possível observar que os modelos obtiveram valores bem parecidos, variando entre 71% e 75%. Mesmo que o *YOLOv5n* tenha tido seu valor abaixo da média dos outros três modelos, considerando sua complexidade, ela atinge um ótimo nível de média *mAP*.

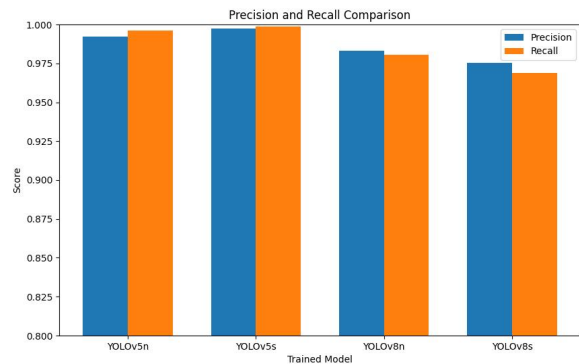


Figura 12. Precision e Recall de cada modelo

A precisão (*precision*) mede a proporção de carros detectados corretamente em relação ao total de carros detectados pelo modelo, isto é, a precisão é a taxa de verdadeiros positivos em relação ao número total de exemplos classificados como positivos (carros detectados). Na detecção de carros em

imagens, ela indica a capacidade do modelo de não fazer falsas detecções. Um valor alto de precisão significa que o modelo tem uma baixa taxa de falsos positivos, ou seja, a maioria das detecções positivas corresponde realmente a carros.

$$Precision = \frac{TP}{TP + FP}$$

O *recall*, também conhecido como taxa de verdadeiros positivos, mede a proporção de carros detectados corretamente em relação ao número total de carros presentes nas imagens, isto é, ele indica a capacidade do modelo de detectar corretamente os carros presentes nas imagens. Em outras palavras, o ele é a taxa de verdadeiros positivos em relação ao número total de exemplos positivos (carros reais) nos dados. Um valor alto de recall significa que o modelo tem uma baixa taxa de falsos negativos, isto é, a maioria dos carros reais presentes nas imagens é detectada corretamente.

$$Recall = \frac{TP}{TP + FN}$$

Observando a Figura 12, pode-se notar que todos os modelos apresentam altíssimos níveis de precisão e *recall*. Porém, note que os níveis dos modelos *YOLOv5* são superiores aos modelos *YOLOv8* em ambas as versões (*nano* e *short*).

VIII. CONCLUSÃO

Portanto, considerando as métricas abordadas para avaliar os modelos *YOLO* e tendo em vista todas as análises gráficas feitas acima, a melhor escolha de um modelo para ser aplicado em um problema com a complexidade deste projeto seria o *YOLOv5n*. Apesar dele possuir uma menor complexidade em relação ao número de parâmetros, seu tamanho de arquivo é muito menor comparado ao restante dos modelos. Seus níveis de resultados *mAP*, precisão e *recall*, apesar de menores, chegam muito perto dos maiores níveis que foram obtidos pelo *YOLOv5s*. Isso mostra que, mesmo que um modelo tenha alto poder de complexidade, como o *YOLOv8s*, nem sempre é a opção mais eficiente para ser usada em todos os tipos de problemas.

REFERÊNCIAS

- [1] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. <http://www.deeplearningbook.org>
- [2] Agarwal, A., & Sehgal, V. (2019). An IoT based smart parking system based on image processing. International Journal of Computer Applications, 181(25), 28-31. DOI: 10.5120/ijca2019918831.
- [3] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. <https://arxiv.org/abs/1804.02767>
- [4] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. <https://arxiv.org/abs/2004.10934>
- [5] Jocher, G., Stoken, A., Borovec, J., et al. (2021). YOLOv5. <https://github.com/ultralytics/yolov5>
- [6] Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., & Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications, 72, 327-334. DOI: 10.1016/j.eswa.2016.12.029.
- [7] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. <https://arxiv.org/abs/1512.02325>

- [8] Saraiva, D. V., & Rodrigues, J. M. F. (2015). Sistema de Visão Inteligente de Baixo Custo Parque de Estacionamento. Mestrado em Engenharia Elétrica e Eletrônica, Instituto Superior de Engenharia, Universidade do Algarve. Disponível em: <https://sapientia.ualg.pt/handle/10400.1/7668>
- [9] Garcia, P. H. de O. & Scalco, R. (2020). Aplicação de Algoritmos de VSLAM e Reconhecimento de Objetos para Classificação de Veículos e Vagas de Estacionamento em Ambientes Externos Utilizando um Drone. Disponível em: <https://maua.br/files/122020/aplicacao-algoritmos-vslam-reconhecimento-objetos-para-classificacao-veiculos-vagas-estacionamento-ambientes-externos-utilizando-um-drone-151457.pdf>
- [10] Nithya, R., Priya, V., Sathiy Kumar, C. et al. A Smart Parking System: An IoT Based Computer Vision Approach for Free Parking Spot Detection Using Faster R-CNN with YOLOv3 Method. Wireless Pers Commun 125, 3205–3225 (2022). Disponível em: <https://doi.org/10.1007/s11277-022-09705-y>
- [11] Murani, Samuel N. (2021). Parking Space Detection Using Deep Learning. Universidade de Nairobi. Disponível em: <http://erepository.uonbi.ac.ke/handle/11295/155870>
- [12] Dixit, M., Srimathi, C., Doss, R., Loke, S., & Saleemdurai, M.A. (2020). Smart Parking with Computer Vision and IoT Technology. IEEE Xplore. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9163467>
- [13] Padilla Carrasco, D., Rashwan, H. A., García, M. Á., & Puig, D. (2021). T-YOLO: Tiny Vehicle Detection Based on YOLO and Multi-Scale Convolutional Neural Networks. IEEE Xplore. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9658533>
- [14] Amarasooriya, S. (2022). Implementation of Smart Parking System Using Image Processing, Nāgānanda International Journal of Humanities and Social Science. 3(2), 69-82 Disponível em: <https://fhss.niibs.lk/wp-content/uploads/2021/07/Implementation-of-Smart-Parking-System-Using-Image-Processing.pdf>
- [15] Albuquerque, L. M. A. (2021). Melhoramento do estacionamento público. Mestrado em Matemática e Computação, Universidade do Minho. Disponível em: <https://repositorium.sdum.uminho.pt/handle/1822/80682>
- [16] Thakur, N., Bhattacharjee, E., Jain, R. et al. Deep learning-based parking occupancy detection framework using ResNet and VGG-16. Multimed Tools Appl (2023). Disponível em: <https://doi.org/10.1007/s11042-023-15654-w>
- [17] Sarmad Rafique, Saba Gul, Kaleemullah Jan & Gul Muhammad Khan. "Optimized real-time parking management framework using deep learning."Expert Systems with Applications, Volume 220, 2023, p. 119686. ISSN 0957-4174. Disponível em: <https://doi.org/10.1016/j.eswa.2023.119686>.
- [18] Ultralytics, 2023. Disponível em: <https://docs.ultralytics.com/>
- [19] Jacob Solawetz. "What is Mean Average Precision (mAP) in Object Detection?". Roboflow, 2020. Disponível em: <https://blog.roboflow.com/mean-average-precision/>
- [20] Cimirro J. L. S.; "Reconhecimento de Imagens: Uso do Método YOLO no Reconhecimento de Placas de Trânsito". Universidade Federal do Pampa. Trabalho de Conclusão de Curso. Bagé, 2022.
- [21] Shiviy Yohanandan, Sabina Pokhrel. "mAP (mean Average Precision) might confuse you!". Towards Data Science. Disponível em: <https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2>