

Estacionamento Inteligente

Identificador de vagas disponíveis e ocupadas por carros

1st Benjamim Rees Pereira do Nascimento

UNIFESP - Universidade Federal do Estado de São Paulo

São José dos Campos, Brasil
rees.benjamim@unifesp.br

2nd Daniel Ferreira Martins

UNIFESP - Universidade Federal do Estado de São Paulo

São José dos Campos, Brasil
df.martins@unifesp.br

I. RESUMO

Com o objetivo de organizar e facilitar a vida de estudantes e professores que fazem uso cotidiano do estacionamento de automóveis da Universidade Unifesp do Campus de São José dos Campos, iniciou-se um projeto de pesquisa para a identificação e automática de automóveis que ocupam vagas nesse estacionamento, a partir de imagens aéreas obtidas por drones equipados com câmeras e algoritmo classificador tipo rede neural artificial. Primeiramente, as imagens aéreas foram obtidas em diferentes alturas de voo, resoluções (qualidade) e ângulo, de maneira a serem processadas manualmente para a identificação dos limites das vagas de estacionamento e de carros. Desses dados separou-se um conjunto para treinamento do classificador neural e outro para os testes, observando o desempenho em diferentes alturas de voo. A rede neural utilizada foi a yolov8n e a metodologia de avaliação foi a F1-score.

II. INTRODUÇÃO

Com o aumento no fluxo de pessoas que ocorre com o passar do tempo, e também com a necessidade de um estacionamento controlado e organizado, decidimos desenvolver uma ferramenta que auxiliasse alunos e professores no trânsito dentro do estacionamento do campus da UNIFESP de São José dos Campos. Um problema extremamente comum referente aos estacionamentos tanto público quanto privado, que se relacionam a diversos problemas do dia a dia de um ser humano, é o tempo gasto por cada motorista e quando há falta de organização pode gerar um tráfego intenso e ocasionar discussões dentro desse ambiente. Para esse projeto, a ideia inicial é implementar um estacionamento inteligente de maneira que identifique quantas vagas estão sendo ocupadas, quantas vagas estão disponíveis e deixá-lo de modo a ser adaptado para novas atualizações e soluções de problemas, tanto para a faculdade quanto para empresas e negócios. Futuras atualizações seriam referentes à identificação da necessidade de ter a quantidade de vagas oferecidas, pois não necessariamente, todas as vagas estão sendo ocupadas, logo, poderia utilizar uma área para novos objetivos, sejam comerciais ou científicos. Há algumas dificuldades na realização desse projeto que já foram discutidas, a principal delas é a construção da base de dados, que necessitará de uma câmera aérea para a captação de imagens(drone), de tempo para montar uma base de dados

diversa por conta da variação de iluminação do dia, da variação das posições dos carros e as árvores do local que podem atrapalhar na visibilidade da câmera em relação a vaga. Além disso, será feito uma avaliação do comportamento do algoritmo de acordo com uma banco de imagens diferente do utilizado para o seu treinamento, essas imagens serão compartilhadas por outro grupo que está desenvolvendo o mesmo tema, porém com o seu embasamento diferente, além de capturas divergentes em ângulo e em qualidade, trazendo um comparativo entre os métodos utilizados pelas duas IAs. É importante fazer esse comparativo para entender um pouco sobre o custo da realização de um projeto como esse e a viabilização da implementação para outros locais.

III. CONCEITOS FUNDAMENTAIS

1 O que é uma Inteligência Artificial (IA)?

Inteligência Artificial (IA) é um campo da ciência da computação que se concentra no desenvolvimento de sistemas e algoritmos capazes de realizar tarefas que normalmente exigem a inteligência humana. A IA busca criar algoritmos que possam perceber, raciocinar, aprender e tomar decisões de forma autônoma, imitando ou reproduzindo certos aspectos do pensamento humano.

2 O que é bounding box?

Bounding box, também conhecida como caixa delimitadora, é uma representação retangular utilizada para delimitar a região de um objeto em uma imagem. é utilizada para envolver algum objeto de interesse

3 O que é rede neural convolucional?

É um tipo especializado de arquitetura de rede neural, projetada para processar dados com uma estrutura de grade, como imagens. são muito eficazes em tarefas como classificação de imagens, detecção de objetos e segmentação semântica.

4 O que é um drone?

Drone é um dispositivo voador controlado a distância que permite a captura de imagens aéreas

5 O que é OpenCV?

É uma biblioteca multiplataforma e livre, usado na aplicação de área de visão computacional

6 O que é Deep Learning?

Deep learning é uma área da inteligência artificial onde usamos redes neurais artificiais para ensinar um computador a aprender e tomar decisões por conta própria. A parte “deep” do deep learning indica que essa técnica utiliza redes profundas (muitas camadas de rede) para auxiliar no aprendizado de máquina.

7 Técnicas, abordagens e estratégias

Yolo v8n - o YOLO (You Only Look Once) é uma família de modelos de detecção de objetos em tempo real que se destacam por sua abordagem de divisão da imagem em bounding boxes contendo a sua classificação de acordo com que foi identificado, além de apresentar uma velocidade muito superior a outros detectores, possuir código aberto e diferentemente de outros algoritmos, ele só olha uma vez pra imagem antes de enviar para a rede neural. É importante entender que a detecção do Yolo não apenas classifica as imagens, como também identifica a localização de determinado objeto. Seu funcionamento se baseia na divisão da imagem em um grid “y” por “y” células, onde essas células irão fazer a previsão de objetos contidos nela e qual a confiança de ser tal objeto. Serão identificados muitos objetos dentro dessas células e por isso existe o parâmetro de confiança para descartar objetos não classificados corretamente. Para esse trabalho será utilizada a versão 8 do yolo e o tipo n(nano) que é uma versão mais simples e que já possui um pré-treino para o reconhecimento de alguns objetos.

Leave One Out - é uma técnica de validação, na qual o número de testes é equivalente à quantidade de elementos da base de dados, porém a cada iteração apenas uma imagem será usada para teste, enquanto as outras para treino.

IV. TRABALHOS RELACIONADOS

1 Análise e reconhecimento de padrões usando processamento de imagens e inteligência artificial[1]

Esse trabalho tem como objetivo obter análises clínicas de pacientes por meio da observação da íris de cada um deles. Para isso será desenvolvido um sistema para identificar e classificar os sinais presentes na íris através de técnicas de processamento de imagens e redes neurais artificiais, primeiramente é necessário extrair características e por fim treinar uma RNA para classificar os padrões dos desequilíbrios encontrados no corpo dos pacientes.

2 Detecção de Vagas em Estacionamento Usando Visão Computacional e Redes Neurais Convolucionais[2]

Esse trabalho visa detectar e classificar vagas utilizando um sistema que utiliza técnicas de processamento de imagens e redes neurais convolucionais. As vagas são detectadas por meio das marcações horizontais e reconhecimento dos espaços com presença ou ausência de veículos. Assim, não seria necessário conhecimento prévio e marcação manual da localização das vagas do estacionamento.

3 Vehicle Detection and Counting from VHR Satellite Images: Efforts and Open Issues[3]

Esse trabalho trata da detecção e contagem de veículos a partir de imagens de sensoriamento remoto, esse é um tópico de pesquisa ativo aplicado à vigilância de sistemas de tráfego e transporte. A ideia é fazer a classificação binária sobre a existência ou não existência de carros na imagem a partir de imagens de satélite, onde foram utilizados dois modelos baseados em deep learning para essa detecção, o Tiramisu e o YOLO.

4 Red blood cell classification using image processing and CNN[4]

A análise das células sanguíneas pode ser crucial para determinar se o paciente sofre de algum doença, portanto neste trabalho são procuradas deformidades nos glóbulos vermelhos, procurando defeitos em sua textura, formato e tamanho através do processamento de imagem, normalizando algumas imagens, convertendo RGB para cinza e segmentando as imagens. Juntamente com o uso de Redes convolucionais no qual a arquitetura do algoritmo é composta por três camadas convolucionais com max-pooling, back normalization e ReLU.

5 On-Street Parking Spot Detection for Smart Cities[5]

Com o objetivo de diminuir o gasto de combustível, e evitar a perda desnecessária de tempo e combustível, foi desenvolvido um sistema de detecção de vagas capturada por câmeras presentes em alguma região da cidade, onde as vagas livres serão mostradas em um aplicativo. Essas vagas serão detectadas por um algoritmo de CNN, esse algoritmo é usado para classificar as imagens e agrupá-las de acordo com sua similaridade.

6 OpenCV Computer Vision with Python [Livro][6]

Este livro tem o objetivo de ensinar como usar o OpenCV com Python para capturar vídeo, manipular imagens e rastrear objetos com uma webcam normal ou um sensor de profundidade especializado, como o Microsoft Kinect, além de passar configurações básicas e limpas para o projeto .

7 Objective comparison of relief visualization techniques with deep CNN for archaeology[7]

Com o intuito de classificar imagens e detectar objetos no ramo da arqueologia, esse trabalho utilizou imagens capturadas por um avião para treinar um algoritmo de deep CNN (redes convolucionais profundas). Vale ressaltar que neste trabalho foi usado o método de avaliação cross validation com $k = 5$ e uma separação de 80% / 20% train/test.

8 Facial Mask Detection using Semantic Segmentation[8]

O artigo propõe um método que detecte faces através de máscaras, usando segmentação semântica. Para isso, é usado um classificador de rostos binários e treina uma rede neural para obter os resultados a partir de fotos em RGB.

9 Remote Sensing Image Scene Classification Using CNN-CapsNet[9]

Esse artigo aborda a classificação de imagens de sensoriamento remoto usando uma combinação de redes neurais convolucionais e Capsule Networks, ele afirma que modelos de aprendizado profundo, como o CNN, são eficazes para a maioria dos problemas de classificação de imagens. Porém esse método ainda produz erros de classificação, os quais podem ser corrigidos ao adicionar um algoritmo de CapsNet (algoritmo de rede neural), obtendo assim melhores resultados.

10 Deep learning for decentralized parking lot occupancy detection[10]

O artigo propõe uma solução mais eficiente para detecção visual de ocupação de estacionamento baseada em uma Rede Neural Convolucional profunda (CNN) projetada para câmeras inteligentes

V. OBJETIVOS DO TRABALHO

O objetivo deste trabalho é montar uma base de dados com as fotos do estacionamento da Unifesp, segmentar as fotos, de maneira a separar as vagas corretamente, para que seja elaborado um algoritmo de redes neurais convolucionais que reproduza as funções de um estacionamento inteligente que identifique e faça a contagem de vagas ocupadas e vazias desse estacionamento, além disso, comparar a eficiência mediante diferentes alturas de voo, qualidade da imagem (resolução) e ângulos de visada.

VI. METODOLOGIA EXPERIMENTAL

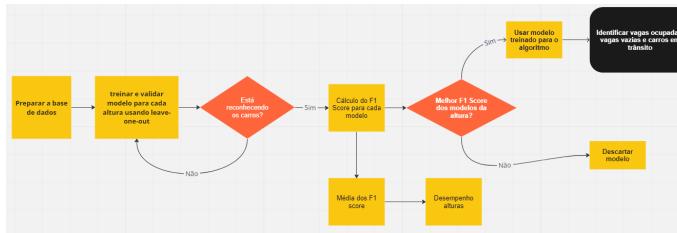


Fig. 1. Diagrama de bloco

Foram capturadas diversas imagens aéreas do estacionamento do campus da Unifesp por meio de um drone, observando variações de altura, para compor a base de dados de treinamento e de avaliação. Essas imagens foram processadas manualmente para a identificação dos "bounding boxes" as vagas do estacionamento e dos automóveis, formando as regiões de interesse para treinamento e avaliação do YOLO.

A base de dados foi dividida de acordo com as alturas de voo. Para a eficiência do identificador foi aplicado o método de validação "leave one out" para cada conjunto de imagens. Dessa maneira, o processo de treinamento dos modelos finaliza-se assim que o algoritmo esteja identificando corretamente os carros, reduzindo os falsos positivos, como

confundir uma vaga livre com um carro, ou falsos negativos, ao deixar de identificar algum carro.

Durante o treinamento, o modelo passa por várias rodadas (epochs), aprendendo e melhorando seu desempenho na tarefa de identificação de automóveis. Após cada iteração, o modelo é validado usando a imagem reservada para esse propósito, medindo o desempenho por meio da métrica F1 Score.

Depois de calcular o F1 score de cada modelo separadamente, será feito a média desses valores para comparar as diferentes alturas, para que possa ser feita uma análise de quais alturas garantem um melhor desempenho do YOLO.

Este trabalho serve de base para o desenvolvimento de soluções para outros problemas, como detecção automática de veículos em vídeos de tráfego, ou mesmo outras situações.



Fig. 2. Exemplo de imagem aérea do estacionamento obtida a 70m de altura.

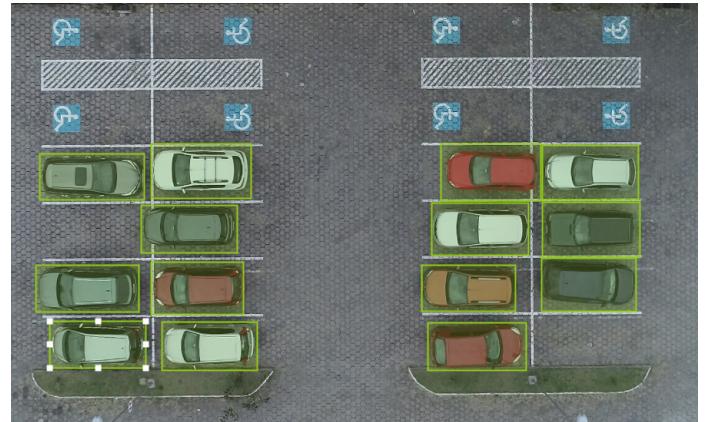


Fig. 3. Bounding box dos carros realizadas por meio do software Make Sense (<https://www.makesense.ai>)

Sobre as fotos capturadas é realizada a bounding box ao redor de cada carro, identificando-se, assim, as regiões de interesse para treinamento do classificador neural. Essas regiões serão utilizadas para treinar o YOLO a identificar carros vistos de cima, essas bounding boxes serão feitas manualmente em um site chamado makesense.ai.

Da mesma forma, é muito importante para a realização desse projeto o mapeamento da vagas no estacionamento, através de bounding box que serão feitas em todas as vagas, de modo a obter as coordenadas das outras regiões de interesse, que são as próprias vagas, para que elas possam ser comparadas com as bounding boxes dos automóveis identificadas pelo modelo treinado.

1 Base de Dados

Na constituição da base de dados de imagens, procurou-se variar o horário, a iluminação, a altura de voo e ângulo de visada, de maneira a dotar o sistema de identificação de uma certa robustez a essas variações típicas de imagens aéreas. Ao todo foram 4 dias capturando fotos para que se obtivesse uma boa variedade de atributos e essa base de dados estará disponível no github, onde temos no total 21 imagens que serão utilizadas. As fotos foram divididas em 5 alturas diferentes, sendo elas: 30m, 40m, 50m, 60m e 70m

altura	quantidade
30 m	9
40 m	4
50 m	4
60 m	2
70 m	2

As imagens de baixa altura foram obtidas em maior quantidade para cobrir todo o estacionamento, o que já não ocorre quando o drone voa mais alto, bastando poucas imagens para cobrir todas as vagas.

Para acrescentar mais imagens a base de dados, aumentando assim a diversidade das características da mesma ,também foram utilziadas imagens providas por outro grupo, as quais foram retiradas da mesma forma.

2 Protocolo de validação

O método utilizado para validar o nosso conjunto de dados será o leave one out, que será aplicado sem agrupar fotos de alturas diferentes, nesse método, é realizado o treinamento e a validação do modelo repetidamente, deixando uma única amostra de fora do conjunto de treinamento a cada rodada. Esse método se mostra mais adequado, visto que a nossa base de dados é relativamente pequena.

3 Protocolo experimental

O modelo de IA escolhido foi o yolov8n e para avaliar os resultados encontrados após o treinamento, foi utilizada a métrica de avaliação: F1-score. Essa métrica é a média harmônica entre a precisão e a revocação. A precisão mede a capacidade que o modelo tem de evitar falsos positivos, enquanto a revocação mede o quanto o modelo classifica falsos negativos.

Como a estratégia escolhida para validação, foi utilizado o leave one out, e como a base de dados conta com 21 imagens teremos, portanto, 21 modelos de treinamento diferentes. Para treinar o YOLO a reconhecer os carros, foram utilizados 100 "epochs"(iterações do treinamento), com uma "confiança" de 0,4 e com um "patience" igual a 50, ou seja, caso o

algoritmo não encontre melhora nos resultados de treinamento durante 50 epochs, o treinamento é concluído imediatamente. O treinamento e teste foi separado em relação a altura, ou seja, para testar uma imagem de 30 metros, foram usadas para treino apenas imagens de altura equivalente. Ao concluir o treinamento e realizar os testes foi avaliada a média dos F1 Score dos modelos de cada altura.

4 Análise de resultados

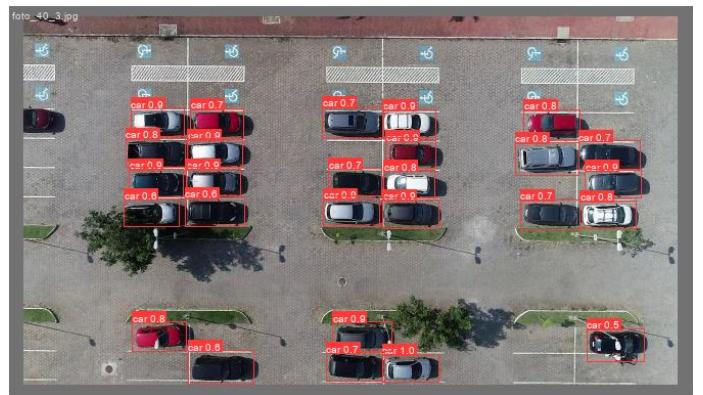


Fig. 4. Exemplo de identificação de carros pela rede neural, em imagem coletada a 40m de altura.

Antes de identificar as vagas livre ou ocupadas e fazer contagem de veículos, é necessário fazer com que o YOLO identifique os carros com um limiar de confiança adequado, com o mínimo de ocorrências de falsos positivos e falsos negativos.

Ao analisar a figura 4, pode-se inferir que o yolo está atuando de forma correta, com limiar de confiança sendo de 0.4 ele consegue identificar a maioria dos carros sem que haja a identificação de outros objetos que não sejam carros, isso demonstra que o seu treinamento foi eficaz. Porém alguns problemas tendem a ser recorrentes para todos os testes, como a não identificação carros que estão debaixo de árvores, ou a não identificação de carros que não são completamente visíveis, como os que se encontram no canto das imagens.

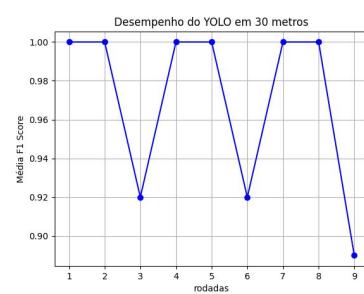


Fig. 5. Desempenho 30m

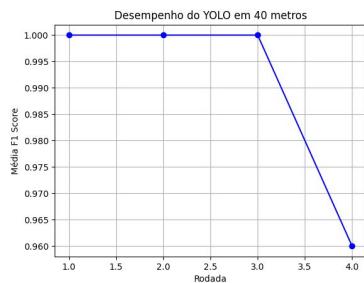


Fig. 6. Desempenho 40m

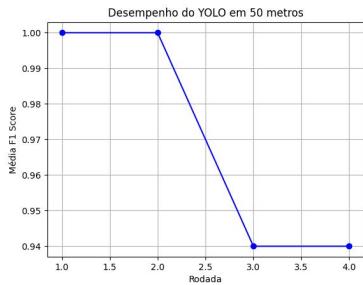


Fig. 7. Desempenho 50m

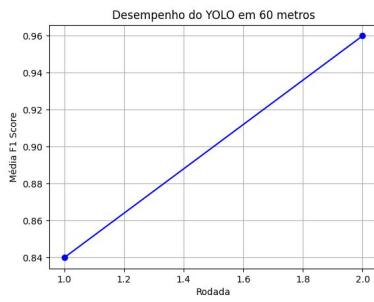


Fig. 8. Desempenho 60m

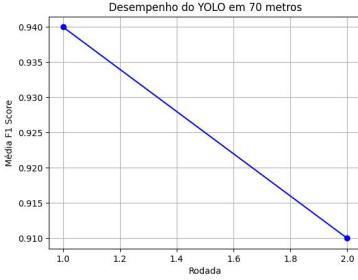


Fig. 9. Desempenho 70m

As Figuras 5 a 9 apresentam os gráficos resultantes da avaliação "F1 Score", obtido das imagens de teste. Cada gráfico corresponde a uma altura diferente, 30, 40, 50, 60, e 70. É importante destacar que o F1 score é uma métrica amplamente utilizada, que permite avaliar o desempenho de

algoritmos de detecção de objetos, pois considera tanto a precisão quanto a recall.

Ao analisar os resultados do experimento, observa-se que o F1 score foi alto (acima de 0.84) em todas as rodadas, indicando um desempenho consistente do YOLO na identificação de carros independente da altura escolhida. Esse resultado é bastante promissor, pois sugere que o modelo treinado com imagens de uma determinada altura possui uma capacidade robusta de generalização considerando que essas imagens tenham a mesma altura.

Além disso, mostra-se que o YOLO é bem robusto e eficiente, pois mesmo sendo treinado em um conjunto de dados pequeno e pouco diversificado, ele obteve ótimos resultados e foi capaz de identificar um número muito expressivo de verdadeiros positivos, evitando com sucesso os falsos positivos e negativos.

Outro aspecto relevante é a qualidade das imagens utilizadas no treinamento e no teste. Imagens de alta qualidade, com boa resolução e iluminação adequada, tendem a fornecer melhores resultados de detecção de objetos. Bons resultados de F1 score na identificação de objetos são consequência de uma boa base de dados.

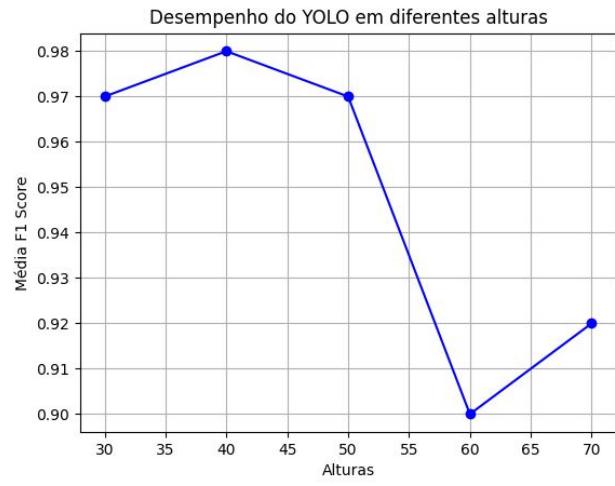


Fig. 10. Desempenho de todas as alturas

Por fim, tem-se na Figura 10 o gráfico que apresenta as alturas no eixo X, variando de 30 a 70 metros em intervalos de 10 metros, e a média do F1 Score no eixo Y. Esse gráfico trás a informação de diferença entre os F1 score para as alturas. Os resultados F1 Score permanecem relativamente altos, pois não regredem de 0.9 ao longo das diferentes alturas, isso indica uma certa consistência no desempenho do algoritmo YOLO para identificação de carros. Porém pode-se afirmar que para alturas mais baixas como 30,40 e 50 metros, o YOLO adquiriu resultados ligeiramente mais altos.

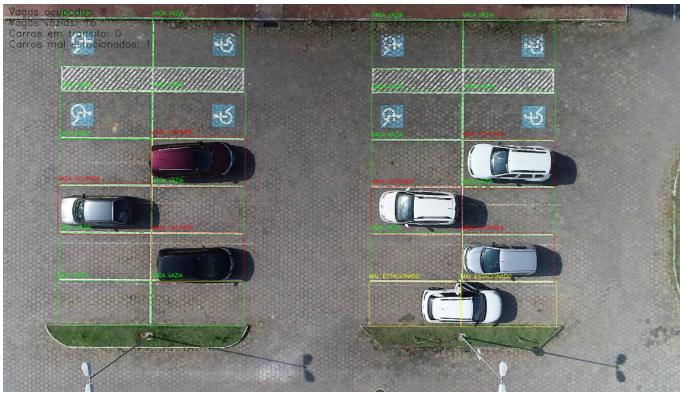


Fig. 11. Resultado 1 do algoritmo final do estacionamento inteligente

A figura 11 mostra o estacionamento da UNIFESP a 30 metros de altura, contendo vários carros e suas respectivas bounding boxes, que foram identificadas pelo algoritmo YOLO. Utilizando essas bounding boxes e as bounding boxes demarcadas a mão através do site makesense.ai e mostradas na foto através do OpenCV, podemos calcular a sobreposição entre essas bouding boxes, de forma a determinar se a vaga está vazia (bounding box verde) ou ocupada (bounding box vermelha). A sobreposição entre as bounding boxes é medida pelo coeficiente de IoU (Intersection over Union). Este coeficiente mede a proporção da área de sobreposição entre as duas bounding boxes(do carro e da vaga) em relação à área da sua união. Quanto maior o valor do coeficiente de IoU, maior é a sobreposição entre as bounding boxes.

As vagas são consideradas ocupadas quando a sobreposição entre elas atinge 60%, e são consideradas vazias quando a sobreposição está entre 30% até 0. Além dessas duas classificações, também há vagas consideradas irregulares, ou seja, com um carro mau estacionado, quando possuem o coeficiente IoU com um valor entre 60% e 30%. Para se diferenciarem das demais, essas vagas estão marcadas de amarelo.

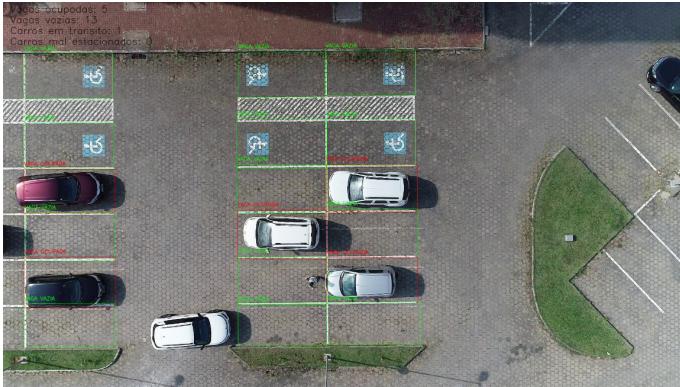


Fig. 12. Resultado 2 do algoritmo final do estacionamento inteligente

Na Figura 12, que também possui 30 metros, podemos observar que além da contagem de vagas vazias e ocupadas.

Também foram identificados carros em trânsito, que são classificados dessa maneira quando não atingem o limiar de sobreposição (IoU) de uma vaga ocupada, ou seja, quando a bounding box do carro não sobrepõe a bounding box de uma vaga.

REFERENCES

- [1] Á. B. Krug, A. Parraga, F. Lorenzi, M. Nicolao, and A. S. Morales, “Análise e reconhecimento de padrões usando processamento de imagens e inteligência artificial,” Revista de Iniciação Científica da ULBRA, vol. 7, no. 7, 2008, Accessed: Jun. 06, 2023. [Online]. Available: <http://www.periodicos.ulbra.br/index.php/ic/article/view/1649/1168>
- [2] W. Cavenagli and C. Mendes, “Detecção de Vagas em Estacionamento Usando Visão Computacional e Redes Neurais Convolucionais.” Accessed: Jun. 06, 2023. [Online]. Available: <https://periodicos.univali.br/index.php/acoth/article/view/16741/9467>
- [3] A. Froidevaux et al., “Vehicle Detection and Counting from VHR Satellite Images: Efforts and Open Issues,” IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, Sep. 2020, doi: <https://doi.org/10.1109/igarss39084.2020.9323827>.
- [4] M. A. Parab and N. D. Mehendale, “Red blood cell classification using image processing and CNN,” May 2020, doi: <https://doi.org/10.1101/2020.05.16.087239>.
- [5] S. Goren, D. Fatma Oncevarlik, and K. Doruk Yldz, “On-Street Parking Spot Detection for Smart Cities,” Sep. 09, 2019. https://www.researchgate.net/profile/Dilan-Oncevarlik/publication/340813046_On-Street_Parking_Spot_Detection_for_Smart_Cities/Street-Parking-Spot-Detection-for-Smart-Cities.pdf
- [6] J. Howse, OpenCV computer vision with Python : learn to capture videos, manipulate images, and track objects with Python using the OpenCV Library. Birmingham, Uk.: Packt Pub, 2013.
- [7] A. Guyot, M. Lennon, and L. Hubert-Moy, “Objective comparison of relief visualization techniques with deep CNN for archaeology,” Journal of Archaeological Science: Reports, vol. 38, p. 103027, Aug. 2021, doi: <https://doi.org/10.1016/j.jasrep.2021.103027>.
- [8] T. Meenpal, A. Balakrishnan, and A. Verma, “Facial Mask Detection using Semantic Segmentation,” IEEE Xplore, Oct. 01, 2019. <https://ieeexplore.ieee.org/document/8888092> (accessed Nov. 26, 2020).
- [9] W. Zhang, P. Tang, and L. Zhao, “Remote Sensing Image Scene Classification Using CNN-CapsNet,” Remote Sensing, vol. 11, no. 5, p. 494, Feb. 2019, doi: <https://doi.org/10.3390/rs11050494>.
- [10] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, “Deep learning for decentralized parking lot occupancy detection,” Expert Systems with Applications, vol. 72, pp. 327–334, Apr. 2017, doi: <https://doi.org/10.1016/j.eswa.2016.10.055>.