

Detecção de Vagas de Estacionamento Baseada em Visão Computacional com Mínima Intervenção Humana

Caio de Souza de Medeiros[✉] e Rigel P. Fernandes[✉]

Resumo— Este trabalho propõe um método de detecção automática da posição de vagas de estacionamento utilizando técnicas de visão computacional, reduzindo a necessidade de intervenção humana para tal. Baseado no modelo YOLOv8 e no agrupamento DBSCAN, avaliamos a capacidade de identificar automaticamente espaços ocupados e livres em imagens reais obtidas do dataset PKLot. O principal objetivo é proporcionar escalabilidade e praticidade na implantação de sistemas de estacionamento inteligente. Resultados preliminares confirmam a viabilidade do método proposto, demonstrando robustez e acurácia satisfatória sob diferentes condições climáticas e de iluminação. O método proposto conseguiu detectar 63,1% das vagas.

Palavras-Chave— Detecção de vagas, Visão computacional, Estacionamentos inteligentes, YOLOv8, DBSCAN.

Abstract— This work proposes a method for the automatic detection of parking spaces using computer vision techniques, reducing the need for human intervention. Based on the YOLOv8 model and DBSCAN clustering, we evaluate the ability to automatically identify occupied and free spaces in real images obtained from the PKLot dataset. The main goal is to provide scalability and to facilitate the deployment of smart parking systems. Preliminary results confirm the feasibility of the proposed method, demonstrating robustness and satisfactory accuracy under different weather and lighting conditions. The proposed method detected 63.1% of the parking spaces.

Keywords— Detection of parking spaces, Computer vision, Smart parking, YOLOv8, DBSCAN.

I. INTRODUÇÃO

O aumento da urbanização e das frotas de veículos intensificam a escassez de vagas de estacionamento e os desafios da mobilidade urbana [1]. Estacionamentos inteligentes são sistemas integrados que utilizam tecnologias avançadas como câmeras, sensores de distância, redes de comunicação sem fio, Internet das Coisas (IoT) [2], processamento de imagens, inteligência artificial e aplicativos móveis — para gerenciar, em tempo real, a disponibilidade de vagas de estacionamento.

Os sistemas de estacionamentos inteligentes podem contribuir na orientação dos motoristas para vagas livres, facilitam o pagamento eletrônico e, conseqüentemente, melhoram a mobilidade urbana. Uma das contribuições da redução das buscas por vagas é a redução das emissões de poluentes [3].

Este trabalho dá continuidade ao estudo apresentado em [4], aprofundando-se agora na automação da detecção das vagas

para o mapeamento inicial do estacionamento. Foi utilizado um dataset com o objetivo da redução da intervenção humana no mapeamento das coordenadas das vagas.

A Seção II detalha o problema e as técnicas de que podem ser usadas para a detecção. A Seção III descreve o método proposto. A Seção IV apresenta o dataset utilizado e os resultados experimentais. As conclusões são discutidas na Seção V.

II. O PROBLEMA DE REDUÇÃO DA INTERVENÇÃO HUMANA

No estudo anterior, foi abordada a detecção da ocupação de vagas de estacionamento por meio de modelos de aprendizado de máquina aplicados à identificação de veículos. As coordenadas dos carros detectados eram comparadas com as coordenadas previamente definidas das vagas, e, a partir de um limiar de distância, inferia-se a ocupação de cada vaga [4].

Contudo, as coordenadas das vagas eram inseridas manualmente — um processo trabalhoso, especialmente em estacionamentos com grande número de vagas. Era necessário identificar visualmente as demarcações no piso e, a partir delas, o operador deveria assinalar a posição de cada vaga.

Na ausência de sinalização física clara, esse processo se tornava ainda mais complexo, exigindo que o operador inferisse, com base nas posições usualmente ocupadas pelos veículos, quais áreas poderiam ser consideradas como vagas. Além disso, caso a organização espacial das vagas fosse alterada devido a modificações no ambiente físico, o sistema manual exigiria novo mapeamento, o que impacta a escalabilidade da solução. Diante dessas limitações, este trabalho propõe uma abordagem automatizada para a definição das coordenadas das vagas, com o objetivo de reduzir significativamente a intervenção humana no processo de sensoriamento.

III. O MÉTODO

A primeira opção cogitada em relação à detecção da posição das vagas foi o treinamento de um modelo de detecção de objetos YOLO [5] para identificar espaços de vagas nas imagens diretamente. Entretanto, não há variedade significativa de datasets públicos de diferentes estacionamentos. Desse modo, replicar esse método em outro estacionamento poderia ser um problema devido a potencial incapacidade de generalização do modelo. Visto a inviabilidade da criação de um dataset apenas para o treinamento desse modelo, preferiu-se a adoção de outra estratégia utilizando recursos já disponíveis.

O método proposto consiste em detectar os carros e seus centróides em todas as imagens de um dataset, acumulá-los em uma única imagem, permitindo assim a visualização das posições já ocupadas pelos carros, independentemente do momento. Desse modo, regiões em que há maior concentração de centróides indicariam maior frequência de carros detectados naquela região. Com a distribuição dos centróides é realizado a técnica de clusterização para encontrar agrupamentos que, consequentemente, indicam posições de vagas de estacionamento. Foram exploradas duas abordagens. A primeira é a clusterização única com um conjunto definido de imagens. A segunda é uma abordagem iterativa agrupando centróides detectados em conjuntos de imagens de forma iterativa em todo dataset. O algoritmo de clusterização DBSCAN [6] foi o escolhido para este método.

A seguir, uma explicação sobre a abordagem de clusterização iterativa: para uma lista L , iniciada como $L_0 = \emptyset$, agrupamos sucessivamente lotes B_i de tamanho b (onde $B_i = \{c_{ib+1}, \dots, c_{\min((i+1)b, N)}\}$), executamos a clusterização com DBSCAN(ϵ , min_samples), adicionamos todos os centróides de B_i a L , removemos de L todos os centróides com rótulo $\ell(x) = -1$ (ruído) e repetimos até usar todos os N centróides. Após $\lceil N/b \rceil$ iterações, L^* contém as coordenadas finais das vagas.

Algorithm 1 Abordagem iterativa de agrupamento de centróides com DBSCAN

Require: Conjunto de centróides detectados $\mathcal{C} = \{c_1, \dots, c_N\}$, tamanho do lote B , parâmetros do DBSCAN (ϵ , MinPts)
// Inicializar lista de centróides ativos e índice
 $L \leftarrow \emptyset$
 $k \leftarrow 1$
while $k \leq N$ **do**
// Selecionar próximo lote de centróides
 $\mathcal{B} \leftarrow \{c_k, \dots, c_{\min(k+B-1, N)}\}$
// Adicionar lote à lista de centróides ativos
 $L \leftarrow L \cup \mathcal{B}$
// Executar agrupamento com DBSCAN
labels \leftarrow DBSCAN($L, \epsilon, \text{MinPts}$)
// Remover centróides rotulados como ruído
 $L \leftarrow \{c \in L \mid \text{labels}(c) \neq \text{noise}\}$
// Avançar índice para o próximo lote
 $k \leftarrow k + B$
end while
Retornar rótulos de cluster para todos os centróides

IV. RESULTADOS EXPERIMENTAIS

Esta seção apresenta os resultados experimentais do método proposto.

A. O dataset

Neste estudo, foram utilizadas as imagens do conjunto UFPR04 do dataset PKLot [7]. O conjunto contém um total de 3.791 imagens de um estacionamento, capturadas por uma

câmera fixa, sem variação de movimento. O dataset contém imagens do estacionamento em diferentes contextos climáticos – chuva, céu aberto e com nuvens. Por outro lado, os métodos aqui implementados desconsideram variações de ângulo e posição da câmera, uma vez que não existem tais variações nesse dataset.

B. Testes e resultados

Os testes realizados nessa experimentação utilizaram a linguagem Python e bibliotecas Sci-Kit Learn [8] para o DBSCAN, OpenCV [9] para manipulação das imagens, Ultralytics para detectar os carros com YOLO e Matplotlib [10] para a exibição dos resultados no arquivo notebook Python presente no repositório do GitHub deste estudo.

Usando a abordagem de clusterização única, foi feito o teste com o dataset inteiro. Depois de alguns testes com diferentes valores dos parâmetros ‘eps’ (Distância máxima entre os pontos) e ‘min_samples’ (Mínimo de pontos vizinhos), o melhor valor obtido foi 8 e 600, respectivamente, identificando 21 de 38 vagas. Isso indica uma acurácia total de aproximadamente 55% das vagas identificadas, Figura 1(a).

Em seguida, realizado um teste de comparação da quantidade de imagens utilizadas. Comparada a clusterização única em 1000, 2000 e 3000 imagens, obtivemos uma acurácia de aproximadamente 32%, 32% e 59%. Esses resultados podem ser observados nas Figuras 1(b), 1(c), e 1(d) respectivamente. Os valores ótimos encontrados de ‘eps’ foram 10 para os 3 conjuntos e ‘min_samples’ 170, 350 e 600, respectivamente. Esses resultados indicaram que existe algum número de imagens que maximiza a detecção de vagas, especificamente usar 3.000 imagens resultou em 9 pontos percentuais a mais na acurácia da detecção do que usar todas as 3.791 imagens. A coordenada final de cada vaga é obtida a partir da média dos centróides de cada agrupamento resultante da execução do DBSCAN. A Figura 1(f) mostra a coordenada calculada das vagas em pontos verdes.

A Figura 2 ilustra os resultados do agrupamento iterativo dos centróides dos carros detectados utilizando diferentes tamanhos de lote: 500 e 1500 imagens por iteração. Observa-se que, ao utilizar 500 ou 1.500 imagens, foi obtida maior acurácia com iterações de 500 imagens, com 24 vagas detectadas corretamente (uma acurácia de 63,1%), visto na Figura 1(g). Por outro lado, a abordagem com iterações de 1500 imagens não ultrapassou as abordagens anteriores, com resultado de 22 vagas detectadas como pode ser visto na Figura 1(h). A abordagem com 500 imagens apresenta uma leve superioridade no número final de vagas detectadas após o processamento completo, indicando maior capacidade de refinamento ao longo das iterações. A razão para tal pode ser atribuída pela remoção de centróides ruidosos, ou seja, carros que estavam apenas passando pelo estacionamento e não estavam parados em uma vaga. Nestes casos, o acúmulo errôneo de centróides é evitado. Esses resultados demonstram a viabilidade do método proposto para mapear automaticamente vagas de estacionamento a partir da detecção de veículos, destacando a influência do tamanho do lote no desempenho do processo de agrupamento.

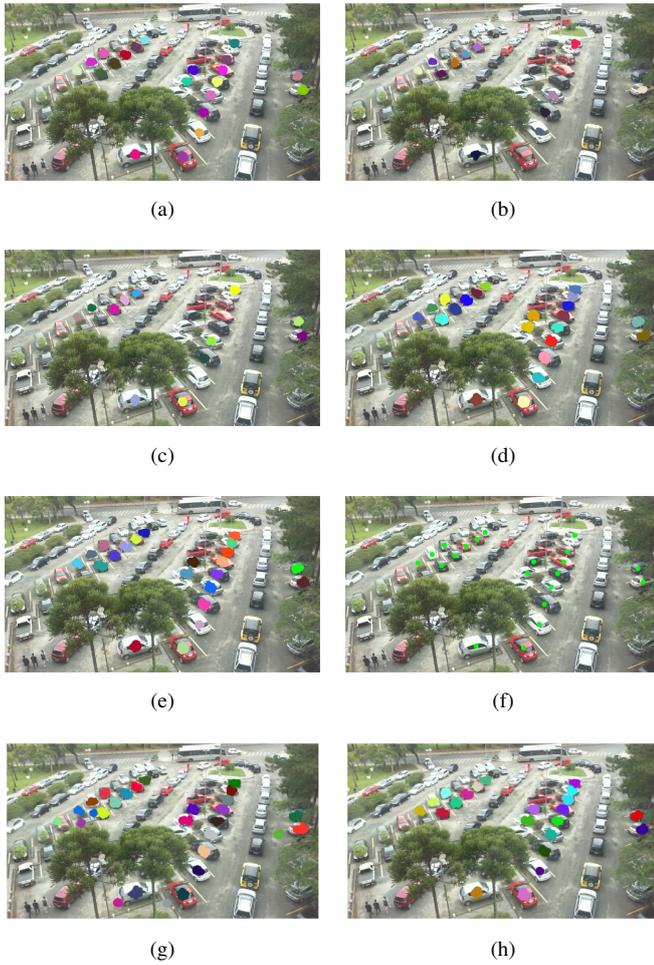


Fig. 1. Distribuição dos centróides nas diferentes técnicas para detectar vagas. (a) Usando o dataset inteiro; (b) Com 1000 imagens; (c) Com 2000 imagens; (d) Com 3000 imagens; (e) Abordagem iterativa com steps de 1000 imagens; (f) Coordenadas dos centróides da vagas; (g) Abordagem iterativa com steps 500 imagens; (h) Abordagem iterativa com steps 1500 imagens

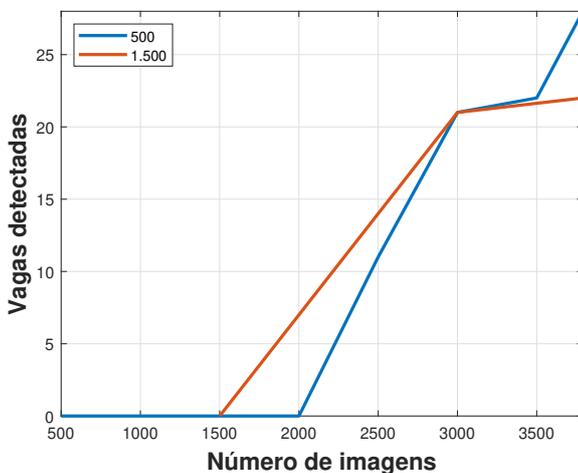


Fig. 2. Vagas detectadas usando a abordagem de agrupamento iterativo.

Tais resultados demonstram a eficácia da combinação da aplicação do YOLOv8 e clusterização com algoritmo DBSCAN para a identificação automática de vagas de estacionamento, sugerindo que essa abordagem possui potencial para definir local das vagas. Além disso, comparam diferentes variações na obtenção das coordenadas das vagas. No entanto, será necessário um ajuste fino dos parâmetros do DBSCAN para lidar adequadamente com veículos muito próximos ou parcialmente visíveis.

V. CONCLUSÕES

O método proposto demonstrou potencial para a detecção automática de vagas de estacionamento em cenários reais, com intervenção humana mínima. Os experimentos iniciais confirmaram a viabilidade da abordagem, destacando, contudo, a necessidade de ajustes para aprimorar a acurácia. Como trabalhos futuros, pretende-se investigar a padronização e o ajuste fino dos parâmetros utilizados na clusterização, explorar algoritmos alternativos como HDBSCAN, além de avaliar o desempenho da abordagem em diferentes conjuntos de dados, visando sua generalização e robustez.

REFERÊNCIAS

- [1] R. Arnott, T. Rave, and R. Schöb, "Alleviating urban traffic congestion," *MIT Press Books*, vol. 1, 2005.
- [2] P. Marwedel, "Embedded systems foundations of cyber-physical systems, and the internet of things," *Embedded System Design; Springer Nature: Cham, Switzerland*, 2021. [Online]. Available: <https://doi.org/10.1007/978-3-030-60910-8>
- [3] K. Reddy, R. Karthikeyan, M. Sudhakar, M. Jabbar, and P. K. Sarangi, "IoT enabled intelligent parking solutions for modern cities," in *2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE, 2025, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/SCEECS64059.2025.10940207>
- [4] H. G. V. Assumpção, C. d. S. de Medeiros, G. d. S. P. Duarte, H. B. D. Rolan, and R. P. Fernandes, "Estacionamento inteligente: uma comparação entre sensores ultrassônicos e visão computacional," in *XLII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Belém, PA, Brazil: Sociedade Brasileira de Telecomunicações, 01-04 October 2024. [Online]. Available: <http://dx.doi.org/10.14209/sbrt.2024.1571036315>
- [5] Y. Liu and S. Shen, "Vehicle detection and tracking based on improved YOLOv8," *IEEE Access*, 2025. [Online]. Available: <https://doi.org/10.1109/ACCESS.2025.3538556>
- [6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. Portland, Oregon: AAAI Press, 1996, pp. 226–231. [Online]. Available: <https://dl.acm.org/doi/10.5555/3001460.3001507>
- [7] P. R. De Almeida, L. S. Oliveira, A. S. Brito Jr, E. J. Silva Jr, and A. L. Koerich, "PKLot – A robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015. [Online]. Available: <https://doi.org/10.1016/j.eswa.2015.02.009>
- [8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, no. null, p. 2825–2830, Nov. 2011. [Online]. Available: <https://dl.acm.org/doi/10.5555/1953048.2078195>
- [9] I. Suneetha, C. Prathima, S. Sirisala, S. Mahalakshmi, K. B. N. K. Reddy, and B. Sambana, "A real time approach for detecting and counting vehicles using opencv and numpy," in *2025 International Conference on Pervasive Computational Technologies (ICPCT)*. IEEE, 2025, pp. 365–370. [Online]. Available: <https://doi.org/10.1109/ICPCT64145.2025.10940439>
- [10] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.