

# Detecção de Vagas em Estacionamento Usando Visão Computacional e Redes Neurais Convolucionais

Willian Marrion Cavenagli  
Universidade Positivo  
Curitiba, Paraná  
willian\_cavenagli@hotmail.com

Caroline Mazetto Mendes  
Universidade Positivo  
Curitiba, Paraná  
caroline.mendes@up.edu.br

## ABSTRACT

Parking lots are no longer practical solutions but become another topic of urban mobility problem due to the difficulty in finding available parking spaces. This work proposes a parking space detection system to assist drivers. The system detects unoccupied vacancies by image processing techniques and convolutional neural networks. Vacancies are detected through horizontal markings and by recognizing spaces with or without vehicles. Finally, a mobile application makes available to the user the occupancy status of vacancies. Initial results showed that the system detects vacancies with visible markings during the daytime. To improve detection in adverse situations, the vacancy detection algorithm is being improved.

## KEYWORDS

Detecção de Vagas, Estacionamento, Visão Computacional, Redes Neurais Convolucionais

## 1 INTRODUÇÃO

Encontrar uma vaga de estacionamento vazia tornou-se uma tarefa diária para os motoristas nas grandes cidades. Como mais um contratempo do cotidiano urbano, os estacionamentos tornaram-se mais um tópico dos problemas da mobilidade urbana, devido a dificuldade em encontrar vagas disponíveis [1].

Além de ocasionar congestionamento, a busca demorada por vagas tem grande impacto na saúde das pessoas. Entre os problemas têm-se o estresse causado pela busca, e dificuldades respiratórias devido ao aumento da emissão de gases CO<sub>2</sub> pelos automóveis [2].

Soluções surgiram para auxiliar o motorista nessa tarefa de encontrar vagas livres nos estacionamentos, tais como sistemas baseados em sensores RFID, ultrassônicos ou infravermelho. Contudo, o custo de instalação e manutenção desses sensores em cada vaga pode se tornar um parâmetro desfavorável, quando se trata de um estacionamento de grande porte [3].

Comparado ao uso de sensores por vaga, abordagens baseadas em imagens podem apresentar melhor custo benefício [4]. Com apenas uma câmera é possível monitorar diversas vagas de estacionamento simultaneamente, diminuindo assim o custo de instalação e manutenção do sistema.

Em geral, soluções baseadas em visão computacional requerem o conhecimento prévio do estacionamento [2, 5]. Nesse caso, quando o posicionamento das câmeras é alterado, é necessário atualizar a localização das vagas, geralmente marcadas manualmente, fazendo com que o sistema não trabalhe com informações em tempo real [6].

Este trabalho apresenta o desenvolvimento em fase inicial de um sistema de detecção de vagas de estacionamento para auxiliar os

motoristas. O sistema realiza a identificação de vagas não ocupadas usando técnicas de processamento de imagens e redes neurais convolucionais (CNN). As vagas são detectadas por meio das marcações horizontais e reconhecimento dos espaços com presença ou ausência de veículos. Assim, a solução proposta não requer conhecimento prévio e marcação manual da localização das vagas do estacionamento.

## 2 SOLUÇÃO PROPOSTA

A solução proposta considera estacionamentos com vagas adjacentes e linhas horizontais paralelas, com ou sem inclinação. Além disso, requer imagens de estacionamentos em perspectiva isométrica (visão superior com inclinação). Cumpridos esses requisitos, o sistema torna-se apto para detecção de vagas com marcações horizontais visíveis, realizando também a predição de vagas com marcações não visíveis devido a presença de carro.

O sistema é composto por módulos com funcionalidades específicas. O módulo de captura obtém imagens do estacionamento por meio de uma câmera, as quais são enviadas ao servidor para análise. O módulo de análise é responsável pela detecção das vagas e definição das suas ocupações. E por fim, o aplicativo móvel disponibiliza para o usuário o estado de ocupação das vagas.

A detecção de vagas ocorre em duas etapas. Na primeira, são detectadas as vagas livres com marcações visíveis por meio de técnicas de processamento de imagens. A segunda etapa realiza a predição de vagas com marcações não visíveis, usando como auxílio uma rede neural convolucional para realizar a detecção de carros estacionados.

Considera-se que uma vaga consiste na área entre duas marcações horizontais paralelas. Para detectar as marcações, são utilizadas técnicas como filtragem, detecção de bordas, operações morfológicas e detecção de linhas. Caso as linhas detectadas sejam paralelas, a região entre elas é definida como sendo uma vaga.

Na segunda etapa, foi usada a abordagem de aprendizado de máquina profundo Mask R-CNN [7]. Dada uma imagem de entrada, ela retorna uma imagem com a área da região de interesse delimitada, incluindo a classe do objeto e suas coordenadas. Para o treinamento da rede, foi utilizada a base de dados COCO (Common Object in Context) [8], que contém mais de 12 mil imagens de carros segmentados em diferentes contextos.

Para que a região contendo um carro seja considerada uma vaga, o carro deve estar presente no mesmo local por um período mínimo de tempo pré-definido. Essas regiões são consideradas como vagas enquanto a posição da câmera não for alterada. Assim, o número de vagas detectadas pode mudar com a aparição de um novo carro na cena. Por fim, é realizada a união do resultado da primeira etapa com o resultado da segunda etapa.

O estado de ocupação de uma vaga é determinado pela porcentagem de sobreposição que um carro detectado têm sobre área da vaga. Esse cálculo é feito por meio de uma operação matemática de conjuntos chamada de interseção sobre união. Por fim, o dados sobre a localização e o estado de ocupação das vagas são armazenados no banco de dados do sistema.

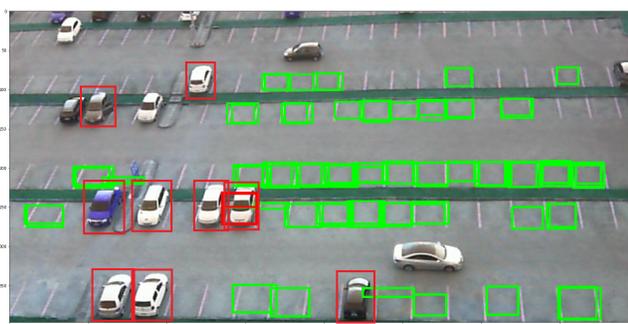
Os dados gerados são acessados pelo aplicativo móvel por meio de um Web Service. O aplicativo exibe a imagem capturada pela câmera, a qual pode ser atualizada com o toque na tela. As vagas são delimitadas por retângulos. O vermelho indica que a vaga está ocupada por um carro, enquanto o verde é usado para indicar vagas livres. Além disso, o aplicativo informa o número de vagas detectadas pelo sistema.

Para o desenvolvimento do módulo de análise foi utilizada a linguagem de programação Python e as bibliotecas OpenCV, Keras e TensorFlow. No desenvolvimento do aplicativo móvel foi usado o *framework* Google Flutter, com o qual é possível gerar versões Android e iOS a partir de um único código.

### 3 CONSIDERAÇÕES FINAIS

Uma das vantagens da solução proposta é que ela não requer o conhecimento prévio ou a marcação manual da localização das vagas. Contudo, o desempenho do sistema depende da qualidade da imagem e das marcações horizontais. A estratégia de determinação de vagas contendo carros estacionados por um período de tempo torna a solução adaptativa. O sistema pode ser iniciado mesmo que existam carros estacionados nas vagas.

Os testes iniciais mostraram que o sistema realizou a detecção de vagas com marcações visíveis, porém não todas as vagas existentes. A Fig. 1 mostra a detecção de vagas em um estacionamento. As marcações em verde são vagas detectadas na primeira etapa, enquanto as marcações em vermelho são vagas detectadas na segunda etapa. Para melhorar os resultados, o algoritmo requer ajustes. Apesar das falhas mencionadas, os estados de ocupação foram corretamente atualizados para o usuário no aplicativo móvel.



**Figura 1: Resultado final, incluindo as vagas determinadas pela presença de carros.**

Na detecção de carros, observou-se que ela nem sempre funciona corretamente em transmissões de vídeo. Mesmo com a Mask R-CNN possuindo boa precisão, eventualmente os carros podem não ser detectados em alguns *frames* do vídeo. Estratégias estão sendo estudadas para minimizar esse problema.

Para realizar a avaliação do sistema serão utilizadas imagens de um estacionamento específico. As imagens serão capturadas por uma câmera de forma contínua por um determinado período de tempo. Também, serão realizados testes para avaliar o desempenho do sistema em tempo real. Espera-se que a solução proposta possa auxiliar os motoristas a encontrar vagas livres em estacionamentos.

### REFERÊNCIAS

- [1] K. B. Dsouza, S. Mohammed, and Y. Hussain. Smart parking – an integrated solution for an urban setting. *2017 2nd International Conference for Convergence in Technology (I2CT)*, pages 174–177, April 2017. doi: 10.1109/I2CT.2017.8226115. URL <https://ieeexplore.ieee.org/document/8226115>.
- [2] L. Baroffio, L. Bondi, M. Cesana, A. E. Redondi, and M. Tagliasacchi. A visual sensor network for parking lot occupancy detection in smart cities. *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pages 745–750, Dec 2015. doi: 10.1109/WF-IoT.2015.7389147. URL <https://ieeexplore.ieee.org/document/7389147>.
- [3] X. Li, M. C. Chuah, and S. Bhattacharya. Uav assisted smart parking solution. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1006–1013, June 2017. doi: 10.1109/ICUAS.2017.7991353. URL <https://ieeexplore.ieee.org/document/7991353>.
- [4] S. Valipour, M. Siam, E. Stroulia, and M. Jagersand. Parking-stall vacancy indicator system, based on deep convolutional neural networks. *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 655–660, Dec 2016. doi: 10.1109/WF-IoT.2016.7845408. URL <https://ieeexplore.ieee.org/document/7845408>.
- [5] N. Bibi, M. N. Majid, H. Dawood, and P. Guo. Automatic parking space detection system. *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, pages 11–15, March 2017. doi: 10.1109/ICMIP.2017.4. URL <https://ieeexplore.ieee.org/document/8221062>.
- [6] Giuseppe Riccardo Leone, Moroni Davide, Pieri Gabriele, Petracca Matteo, Salvetti Ovidio, Azzarà Andrea, and Marino Francesco. An intelligent cooperative visual sensor network for urban mobility. *Sensors*, 17:2588, Nov 2017. doi: 10.3390/s17112588.
- [7] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL <http://arxiv.org/abs/1703.06870>.
- [8] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL <http://arxiv.org/abs/1405.0312>.