Efficient Urban Parking Solution through Machine Learning

Jianger Yu
Department of Computer Science
Virginia Tech
Fall's Church, VA
jianger@vt.edu

Asmi Panigrahi
Department of Computer Science
Virginia Tech
Fall's Church, VA
asmipanigrahi@vt.edu

Atharva Sardar
Department of Computer Science
Virginia Tech
Fall's Church, VA
atharvasardar02@vt.edu

ABSTRACT

Finding a parking space is always an issue for drivers in the city. In this project, we used modern computer technologies to detect the vacancy status of a parking spot from photo. Specifically, we used both Support Vector Machine (SVM) and Computer Vision (CV) approach for determining if a parking spot is empty or vacant, and count the overall vacant spots in a photo. We also used the vacancy information to predict the time of the day (morning, afternoon etc) and the date (weekend or non-weekend). The SVM approach of predicting the vacancy of a car has a decent accuracy of 0.96. The CV approach shows that it can successfully count the overall vacant spots. Finally, we used 7 classification models to predict vacancy based on the time and date of a picture and get the accuracy ranged from 0.68 to 0.92.

KEYWORDS

Parking Spot Detection, Machine Learning, Classification, Neuron Network, SVM, CV

1 INTRODUCTION

The process of urbanization, coupled with a rapid increasing in the number of vehicles on the road, has exacerbated the challenge of finding available parking spots, particularly in metropolitan areas. One strategic approach for addressing this issue is to develop an intelligent parking management system that has the following functions: 1. Detecting the real-time vacancy status of each parking spot in lots or garages, 2. Aggregation of data to provide customers with an overall count of vacant parking spots. 3. Predicting the vacancy status for parking lots in the future.

The key part of this parking management system is how to accurately determine the vacancy status of a parking spot. Fortunately, Close-circuit television (CCTV) systems were already widely deployed in parking lots and garages, offering live camera feeds for monitoring parking spaces. Leveraging this infrastructure, it is feasible to develop a machine learning-based model that is capable of detecting real-time vacancy statuses through these live camera feeds and subsequently calculating the overall number of vacant spots within a garage. In this project, our group has developed a machine learning-based model that is able to resolve the following problems: 1. Given a photo of a parking spot, determine the vacancy status (either occupied or vacant) of this parking spot; 2. Given a garage photo that is captured by the CCTV system, count the total number of parking spots and the number of vacant parking spots, then calculate the vacant percentage of the particular photo; 3. Given the information in the photo, predict the time of the day and if it is the weekend.

In this project, we developed codes for these objectives. For the vacancy detection part we developed two sets of algorithms for implementing it, one is based on the Support Vector Machine (SVM) and the other is based on Computer Vision (CV). For the classifier of prediction the time and date analysis, we used 7 different models for the predictions the details and results will be shown below in this report.

We firmly believe that our developed model serves as a cornerstone of an intelligent parking management system, poised to bring substantial benefits for both the drivers and the garage management organizations. For drivers, the system offers direct access to real-time information and future predictions regarding the vacancy status of parking lots and garages. This functionality empowers drivers to save valuable time and fuel by efficiently locating available parking spots. For garage management organizations, the system can enhance the overall management planning of parking facilities. By providing accurate and up-to-date parking spot occupancy data, the system assists in optimizing resource allocation and improving the overall efficiency of parking facilities.

Overall, there are two contributions to the work of our project:

1. The machine learning model we developed in this project sufficiently addressed the three problems outlined above, specifically, the model employs a neural network to proficiently classify the vacancy status of parking spots. It not only accurately determines the real-time status of individual parking spots but also performs an aggregate count of available spots. Additionally, the model showcases predictive capabilities, foreseeing future vacancy statuses.

2. Beyond its successful implementation, our model demonstrates broad potential applications, promising substantial benefits for both customers and parking management organizations, as well as contributing to the efficiency and effectiveness of parking facilities.

2 RELATED WORK

Much research has been done in this area in the past. In fact, there are several review paper that summarizes the previous and past work on this topic.[2][6]. According to these review papers, the related work can go back to as early as 2007 [5], where the authors used an 8-class Support Vector Machine (SVM) classifier to detect the parking spot vacancy from the input video captured by a camera. Their results showed that the accuracy of their model reached approximately 92%, and the average conflict rate dropped to 4% with over 2400 training samples. Another pioneering work on this topic was published in 2016 by Amato et al[1], they used a Convolutional Neural Network (CNN) classifier for detecting car parking occupancy by using smart networks and Deep Learning. Specifically, they used two datasets, PKLot and CNRPark dataset. They trained and tested their model and baseline models for both intra-datasets and inter-datasets, The results showed that the performance of their

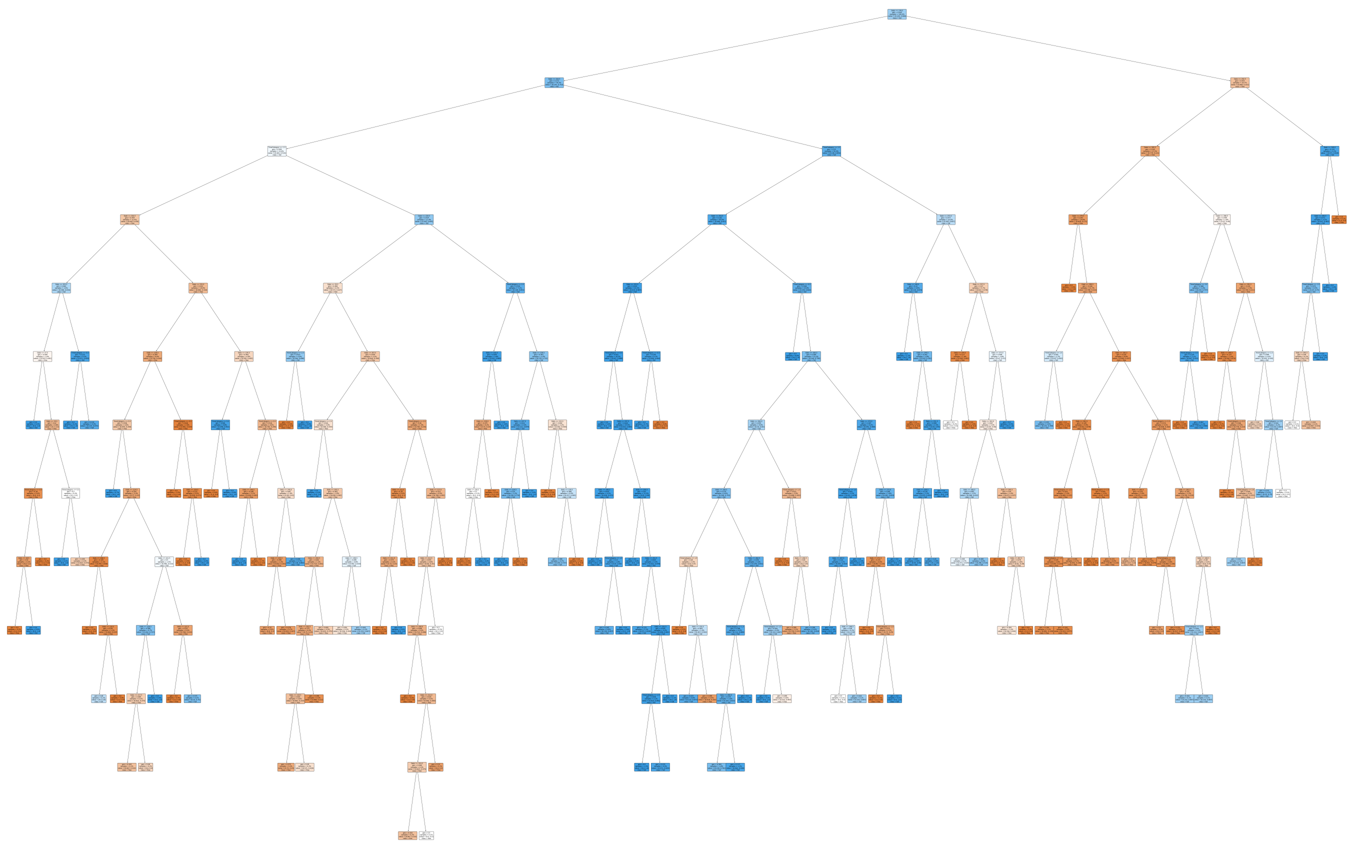

Figure 1: The Full Decision Tree of the Prediction Based on Time and Data Analysis, The Full Version of This Diagram is Attached with this Report

model was better than their baseline models. The overall accuracy were reported to be 0.901-0.981 for the intra-dataset experiments and 0.829-0.904 for the inter-dataset tests.

There have been many more related works in recent years. Ng et al[3] added a structural similarity (SSIM) decision module on top of the CNN classifier and found it could greatly reduce the reaction time in identifying the occupancy status of parking space images extracted from live parking lot camera feeds. Padmasiri et al [4] presented an end-to-end automated vehicle parking occupancy detection system that the object detection process using RetinaNet one-stage detector and region-based CNN. Zhang et al [7] proposed an improved parking space recognition algorithm based on panoramic vision. All of these approaches have improved the performance compared to the original model.

Overall, as a popular topic that is interdisciplinary in both urban planning and computer science, there are many existing related works and papers available on this topic. Due to the limitation of time of this project, it is very difficult to list and introduce all related works in this section. Nevertheless, there are two things we learned from these related works: 1. Most of the related works used the Convolutional Neural Network and deep learning techniques, which provided our group with a direction on solution and implementation of the parking lot vacancy detection problems. 2. The accuracy of these vacancy detection models ranged from 0.83 to 0.98, depending on the datasets and model used. In this case, we simply set the target accuracy of our model to be 90%.

3 ALGORITHMS AND EXPERIMENT

3.1 Dataset

We choose the dataset “Find a Car Park” dataset from the Kaggle.com. This dataset provides us the images of a parking lot which are classified into 2 parts into 2 different folders. The first folder named “Free” contains the images of the parking lot when it has some free space i.e a spot for parking. While there is another folder named “Full” which has the parking lot which is full and does not have any parking space available. This dataset provides us with the perfect opportunity to apply classification models on the dataset. The link and URL can be found at the end of this paper at the "Acknowledgement" section.

3.2 Parking spot vacancy detection from images

There are two approaches for this part: The first part is a SVM approach, the second part is a Computer Vision(CV) approach. Here are the details for each approach:

3.2.1 SVM Approach. Along with this, we applied a machine learning script for image classification using a Support Vector Machine (SVM) classifier and Histogram of Oriented Gradients (HOG) features. It begins by importing necessary libraries such as OpenCV for image processing, NumPy for numerical operations, scikit-image for HOG feature extraction, and scikit-learn for SVM model implementation and evaluation. The script defines constants, including the target image size. The loadimagesfromfolder function reads grayscale images from two folders ('full' and 'free'), resizes them to a specified size, extracts HOG features, and assigns labels (1 for 'full' and 0 for 'free'). The script then combines the data, splits it into training and testing sets, initializes an SVM classifier, trains it on the training data, and evaluates its performance on the test set using accuracy and a classification report.

3.2.2 CV Approach. We implemented a real-time parking lot monitoring system using OpenCV. The script captures image input, applies background subtraction to detect stationary cars, and outlines these objects with bounding rectangles in real-time. This provides a live image representation of parking spot activity, contributing to a simple yet effective surveillance system.

3.3 Prediction based on time and data analysis

We had a data in the format of images which were free and full. It also had the information of the date and time of the image. We extracted this data into a dataframe and based on this data, we train different classifiers to predict if at a specific day and time, will there be a free spot or will it be full. For this we created a machine learning pipeline for image classification using various classifiers, including Decision Trees, Neural Networks, Random Forest, Support Vector Machines (SVM), Logistic Regression, Naïve Bayes, and K-Nearest Neighbors (KNN). We learned the majority of this classifiers in our class. Decision Trees recursively split the data based on features to make decisions, forming a tree structure for classification. It is based on entropy and gini index to predict the classification. Neural Networks consist of interconnected layers of nodes which are based on perceptron which are interconnected in layers that learn complex patterns from data through iterative optimization, making them powerful for capturing intricate relationships in large datasets but requiring careful tuning. Random Forest combines multiple Decision Trees to enhance predictive accuracy and reduce overfitting by aggregating their outputs through voting or averaging. Basically it is a combination of many decision trees which provides us with a final classification. Support Vector Machines (SVM) constructs a hyperplane that maximally separates different classes in the feature space, making it effective for binary and multiclass classification tasks, especially in high-dimensional spaces. Logistic Regression models the probability of binary outcomes using a logistic function, providing interpretable coefficients and simplicity for binary classification tasks. Naïve Bayes relies on the Bayes theorem and assumes independence between features, making it computationally efficient and particularly suitable for text classification tasks. K-Nearest Neighbors (KNN) classifies data points based on the majority class of their k-nearest neighbors in the feature space, making it simple and effective for classification tasks. The pipeline involves importing necessary libraries, defining functions to extract metadata from images and create a DataFrame, preprocessing the data by converting date and time information, encoding categorical variables, and splitting the dataset into training and testing sets. It then initializes each classifier. As we have learned in our class, we should always performs cross-validation to evaluate their performance. Then it fits the models on the training data, predicts on the test data, and evaluates their accuracy. The Decision Tree classifier is visualized with a tree plot.

4 RESULTS

4.1 Parking Spot Vacancy Detection from Photo

As discussed, we have run two algorithms in this part, one is based on the SVM, the results are displayed in Figure 3, and the other code is based on the CV (the results is displayed in Fig 4)

For the CV approach of this part, we concluded the code by closing all OpenCV windows, ensuring a clean exit. Overall, we devised a comprehensive solution of real-time computer vision for a robust parking lot monitoring system.

4.2 Prediction based on time and data analysis

The Decision Tree model achieved an accuracy of 92%, demonstrating strong performance in classifying images. The classification report indicates high precision, recall, and F1-score for both classes (0 and 1), with weighted averages of 92%. The Neural Network, however, shows lower accuracy at 69%, with challenges in correctly classifying the minority class (0). The Random Forest model performs similarly to the Decision Tree with an accuracy of 92%, maintaining high precision, recall, and F1-score for both classes. The Support Vector Machine (SVM) and Logistic Regression models have accuracies of 70%, displaying a notable limitation in correctly identifying class 0. The Naïve Bayes model shows an accuracy of 68%, struggling with precision and recall for class 0. Finally, the K-Nearest Neighbors (KNN) model achieves a high accuracy of 92%, with well-balanced precision, recall, and F1-score for both classes.

5 DISCUSSIONS

5.1 Parking Spot Vacancy Detection from Photo

5.1.1 SVM Approach. When we use the SVM coupled with skimage, we start by converting the images to the grayscale. Then we do the image processing using the hog feature from the skimage library. When we train this model on our entire dataset and predict whether we have a free spot or not, we find that we get a accuracy of 96%. Along with this it provides with a high precision and recall as well. This method works good when we have a lot of images of the same location and gives us a very high accuracy then. The high accuracy shows that it is good at predicting the true positive and the true negative. High precision shows that it is highly good at predicting true positive and less false positive. Along with this it has high accuracy which shows it is good at limiting the false negative. This leads to a good F1 score.

```
Accuracy: 0.96
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.92      0.94       204
           1       0.96      0.98      0.97       449

    accuracy                           0.96       653
   macro avg       0.96      0.95      0.95       653
weighted avg       0.96      0.96      0.96       653
```

Figure 2: The Results of the Prediction of the Vacancy of Parking Detection Using SVM-based Model

```
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 2
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 3
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 4
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 5
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 6
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 7
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 8
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 9
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 10
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 11
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 12
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 13
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 14
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 15
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 16
[INFO] SPACE pressed! New parking area saved.
[INFO] Number of parking lots: 17
[INFO] Exit drawing parking area.
```

Figure 3: The Results of the Prediction of the Vacancy of Parking Detection Using CV-based Model

5.1.2 CV Approach. The provided output suggests that the parking lot annotation process was executed multiple times, resulting in identifying and saving distinct parking areas. Let's discuss the qualitative implications of these results:

1.The repetitive use of the SPACE key to save parking areas indicates a systematic and consistent approach to annotating the parking lot.

2. The number of annotated parking areas provides insights into the density of parking spaces within the given parking lot. This information is valuable for decision-makers using this data to optimize parking lot usage or plan for future expansions.

3. The SPACE key's continuous use to save parking areas indicates user engagement and interaction with the annotation process. This is crucial as it ensures that the user actively participates in defining parking spaces, potentially leading to more accurate and contextually relevant annotations.

4. The informational print statements, such as "SPACE pressed! New parking area saved." and " Exit drawing parking area," provide helpful feedback to the user. This helps in maintaining transparency and clarity throughout the annotation process.

5. The script does not provide visual feedback regarding the drawn parking areas during annotation. Depending on the complexity of the parking lot layout, users might need help to keep track of the areas they have already annotated. Consideration could be given to incorporating visual indicators or overlays to address this potential challenge.

6. The iterative process of annotating multiple parking areas demonstrates the script's efficiency in handling various inputs. The script's ability to manage and save each area efficiently contributes to the potential accuracy of the overall annotation process.

7. The results suggest successfully executing the parking lot annotation script with user-friendly feedback and a systematic approach to identifying and saving multiple parking areas. Further refinements, such as real-time visual feedback during the annotation process, could enhance the user experience and contribute to even more accurate annotations.

5.2 Prediction based on time and data analysis

Based on the results we can see that Decision tree, Random forest and KNN has the best accuracy of 92%. As we have learned in class, Random Forest is an ensemble machine learning algorithm that constructs multiple decision trees during training and outputs based on the prediction of the individual trees for improved accuracy and generalization. So we would like to choose Random Forest algorithm for this classification problem. Based on this we could predict whether we will get a parking spot at a specific place without having the live input of the data, based on the patterns in the past. The high accuracy shows that it is good at predicting the true positive and the true negative. High precision shows that it is highly good at predicting true positive and less false positive. Along with this it has high accuracy which shows it is good at limiting the false negative. This leads to a good F1 score.

```
Decision Tree Accuracy: 0.92
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.88      0.86       195
           1       0.95      0.93      0.94       458

    accuracy                           0.92       653
   macro avg       0.90      0.91      0.90       653
weighted avg       0.92      0.92      0.92       653
```

```
Neural Network Accuracy: 0.69
Neural Network Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       195
           1       0.70      0.99      0.82       458

    accuracy                           0.69       653
   macro avg       0.35      0.49      0.41       653
weighted avg       0.49      0.69      0.57       653
```

```
KNN Accuracy: 0.92
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86       195
           1       0.93      0.95      0.94       458

    accuracy                           0.92       653
   macro avg       0.91      0.89      0.90       653
weighted avg       0.92      0.92      0.92       653
```

```
Random Forest Accuracy: 0.92
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.87      0.87       195
           1       0.95      0.94      0.94       458

    accuracy                           0.92       653
   macro avg       0.90      0.91      0.90       653
weighted avg       0.92      0.92      0.92       653
```

```
SVM Accuracy: 0.70
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       195
           1       0.70      1.00      0.82       458

    accuracy                           0.70       653
   macro avg       0.35      0.50      0.41       653
weighted avg       0.49      0.70      0.58       653
```

```
Logistic Regression Accuracy: 0.70
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       195
           1       0.70      1.00      0.82       458

    accuracy                           0.70       653
   macro avg       0.35      0.50      0.41       653
weighted avg       0.49      0.70      0.58       653
```

```
Naïve Bayes Accuracy: 0.68
Naïve Bayes Classification Report:
              precision    recall  f1-score   support

           0       0.22      0.03      0.05       195
           1       0.70      0.96      0.81       458

    accuracy                           0.68       653
   macro avg       0.46      0.49      0.43       653
weighted avg       0.55      0.68      0.58       653
```

Figure 4: The Results of the Prediction based on Time and Data Analysis by Using 7 Different Models for Classification

5.3 Application of Our Project

As we mentioned in the introduction section, our work can be used in a smart parking management system. Both parking management and drivers can benefit from this system. The results showed that we can accurately predict the vacancy based on the SVM approaches. However, there is still a long way from what we currently have to a completed system. For the ethics issues, we think the only possible issue is the privacy of the car that appears in the photos taken as the sample.

6 CONCLUSION

This project was divided into two parts, part one being the use of image processing techniques to find whether there is a parking spot available based on the data in the form of image. We divided the data into train and test data. We trained our model and tested the model on the test data. In the part b of the project, we extracted the date, time day from the dataset. On that dataset we trained different classification algorithms. Based on our trained models, we could predict at a specific time and day, will we get a free parking spot or it will be free. After comparing different models, based on parameters like accuracy, precision, recall and f1 we choose random forest for the classification. So based on the past data, we could predict the availability of slots at a specific time and date interval.

7 AKNOWLEDGEMENT

All three authors listed contributed equally to this project. Specifically, Asmi Panigrahi (AP) and Atharva Sardar (AS) contribute to the programming and experiment parts, AP contributes to the programming and experiments of the vacancy recognition from the photo by the CV approach. AS contributes to the programming AS contributes to the programming and experiments of time and date analysis. Jianger Yu (JY) contributes to the literature review and related work search. All three authors contributed to writing the report, AP and AS contributed to the written parts related to their algorithms, experiments, results, and discussion, JY contributed to the abstract, introduction, related work and the Applications. JY contributed to editing and formatting the report together in the Latex.

The URL of our dataset:
https://www.kaggle.com/datasets/daggysheep/find-a-car-park/data
The code can be found at:
https://github.com/atharva-sardar02/uc$_{project}$

REFERENCES

[1] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, and Claudio Vairo. 2016. Car parking occupancy detection using smart camera networks and Deep Learning. In *2016 IEEE Symposium on Computers and Communication (ISCC)*. 1212–1217. https://doi.org/10.1109/ISCC.2016.7543901

[2] Abrar Fahim, Mehedi Hasan, and Muhtasim Alam Chowdhury. 2021. Smart parking systems: comprehensive review based on various aspects. *Heliyon* 7, 5 (2021), e07050. https://doi.org/10.1016/j.heliyon.2021.e07050

[3] Chin-Kit Ng, Soon-Nyean Cheong, and Yee-Loo Foo. 2020. Low Latency Deep Learning Based Parking Occupancy Detection By Exploiting Structural Similarity. In *Computational Science and Technology*, Rayner Alfred, Yuto Lim, Haviluddin Haviluddin, and Chin Kim On (Eds.). Springer Singapore, Singapore, 247–256.

[4] Heshan Padmasiri, Ranika Madurawe, Chamath Abeysinghe, and Dulani Meedeniya. 2020. Automated Vehicle Parking Occupancy Detection in Real-Time. In *2020 Moratuwa Engineering Research Conference (MERCon)*. 1–6. https://doi.org/10.1109/MERCon50084.2020.9185199

[5] Wu Q., Huang C., Wang S.-Y., Chiu W.-C., and Chen T. 2007. Robust parking space detection considering inter-space correlation. *Multimedia and Expo, IEEE International Conference* (2007).

[6] Bhattacharjee E. Jain R. et al Thakur, N. 2023. Deep learning-based parking occupancy detection framework using ResNet and VGG-16. *Multimedia Tools and Applications* (2023). https://doi.org/10.1007/s11042-023-15654-w

[7] Jindong Zhang, Tong Liu, Xuelong Yin, Xue Wang, Kunpeng Zhang, Jiabin Xu, and Donghui Wang. 2021. An Improved Parking Space Recognition Algorithm Based on Panoramic Vision. *Multimedia Tools Appl.* 80, 12 (may 2021), 18181–18209. https://doi.org/10.1007/s11042-020-10370-1