Free Categorized Space Detection in Parking Lots

Yuchen Hou
University of California, Santa Barbara
UCSB, Santa Barbara, California 93106
yuchenhou@ucsb.edu

Kimia Afshari
University of California, Santa Barbara
UCSB, Santa Barbara, California 93106
kimia_afshari@ucsb.edu

Abstract

With the increasing number of vehicles on the road, parking spaces have become scarce resources in urban areas, and it can be challenging to find available spots, especially for people with disabilities. Automatic parking space detection can not only facilitate the process of looking for an available parking spot but also help drivers save time and effectively reduce emissions as well as traffic congestion by navigating directly to the pre-suggested spot. In parking lots, there are some spaces allocated to people with disabilities to improve their accessibility to the spots. To help them automatically find those spaces, we proposed an image-based smart parking system that looks for available regular and accessible parking spots. Thus, this project highlighted the importance of developing automatic parking systems with special attention to accessible parking spots in order to create a more equitable and accessible urban environment.

1. Introduction

Finding parking space in metropolitan areas is a major problem. Automatic parking space detection can facilitate the process of looking for an available parking spot, help drivers save time, and effectively reduce emissions and traffic congestion by navigating directly to the pre-suggested spot. Traditional parking space detection techniques used sensors to record the entry and exit of each vehicle [29] or relied on low-level features (such as lines, corners, and colors of the cars' or parking spots' edges) to recognize and classify each parking place [30]. However, these methods either have high costs or low robustness towards changes in light and road surface conditions. Recently, deep learning has shown promising results in object detection, and it is less expensive and more stable under noisy environments [30]. Therefore, the deep-learning approach may provide a better solution for the parking space detection problem.

Additionally, nearly 1.3 billion people worldwide have significant disabilities [1]. Due to the need for mobility-aiding devices to perform daily activities, those people are more likely to have difficulty finding suitable parking spots [1, 2]. To help them find accessible facilities in parking lots, monitoring and detecting empty accessible parking spots is crucial. Previous methods for these tasks involved manual inspections or the installation of sensors beneath each disabled parking spot [2]. These approaches are labor-intensive, require significant maintenance funding, and rely on the pre-existing knowledge of each disabled parking spot's location. In contrast, using deep learning techniques to detect and regulate disabled parking spots is much more efficient and can adapt to changes in the location of each disabled parking spot.

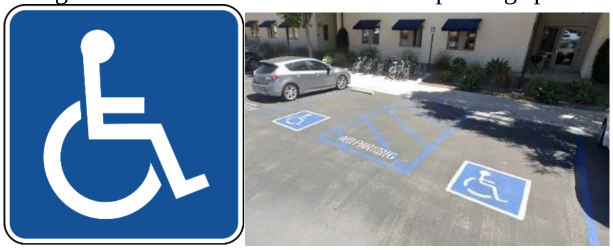

Figure 1. Left: International Symbol of Access. Right: Example of two disabled parking spots with international symbols of access road marks. The image is taken from Google Street View.

In order to monitor the parking spots and locate the suitable ones for drivers or passengers with disabilities, we designed two modules: parking space detection and international access symbol (handicap) road mark detection. The parking space detector fine-tuned a Mask-RCNN [32] model to locate and classify the parking spots, and the handicap symbols road mark detector fine-tuned a Faster R-CNN model to find the handicap road marks. By combining these two detectors, the current study aimed to identify both regular and accessible parking spots.

Commonly used parking lot datasets, such as PKLot [4] and CNRPark [5], lack images with handicap symbols, and there is no dataset designed specifically for handicap road mark detection in parking lots. The only two publicly available detection datasets containing images and labels of handicap symbols in parking spaces are the Mapillary Traffic Sign Dataset (MTSD) [7] and San Francisco Parking Sign Detection Dataset [8]. However, in both

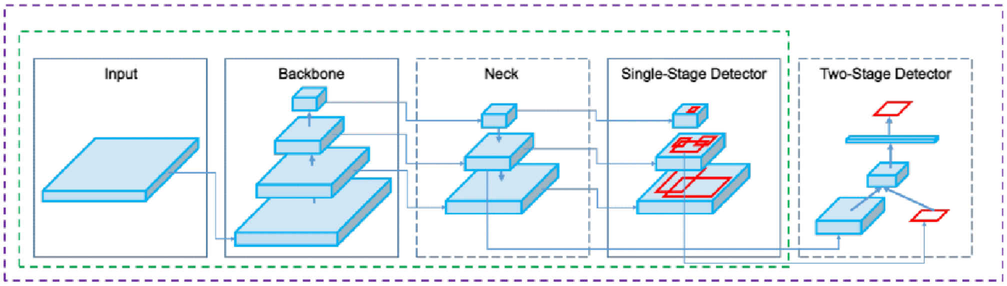

Figure 2. Object detection framework of the one-stage object detection (green-based box) and two-stage object detection (purple-based box). Adapted from [15].

datasets, the symbols are presented in the format of traffic signs instead of road marks painted on the ground. While the handicap traffic signs are helpful for drivers to find parking space from a long distance, the handicap road marks give a more precise location of the disabled parking space and are more visible from the aerial view, which is the typical angle of a CCTV camera for parking lot monitoring.

Due to the reason mentioned above, for the parking spots detection module, we used the Action-Camera Parking Dataset (ACPDS) [21], which contains 293 parking lot images taken from unique views and has several images containing handicap road marks. For the handicap road mark detection module, we designed a custom dataset by collecting and labeling parking space handicap mark images from various sources, including some images from the ACPDS dataset.

Finally, we investigated the performance of our proposed methods on the ACPDS dataset for parking spot detection and on the custom dataset for handicap mark detection. The evaluation results showed 56.3 % mAP@0.5 for space detection and 73.2% mAP@0.5 for handicap detection on the test set. We also found that the model performed better at detecting handicap marks in regions with no shadow and where the mark's size is larger than 32×32 pixels on the image.

2. Related Work

2.1 Object Detection Models

Object detection is one of the most essential techniques in computer vision. It involves localizing the object of interest within images and classifying the objects based on categories. Since AlexNet [12], Convolutional Neural Networks (CNNs) have greatly improved object detection task performances by making use of a hierarchy of convolutional kernels with learned weights sliding across the previous layer to extract complex feature representations of each image [13].

Currently, there are two major types of deep-learning-based detection frameworks (Figure 2). The first one is single-stage object detection which divides images into grids and predicts the bounding boxes along with object classification confidence scores for each grid [14]. The second method is two-stage object detection, which combines a region proposal framework for generating possible regions of objects and a CNN framework for predicting each object's location and category [16].

2.1.1. Faster R-CNN

The Faster R-CNN architecture (Figure 3), proposed by Ren et al. [19], is one of the most influential two-stage object detection frameworks. It consists of 2 modules: the Fast R-CNN [17] module and a Region Proposal Network (RPN) module. The RPN module takes the feature map produced by the feature-extraction CNN as the input. On each feature map position, a maximum *k* number of candidate boxes are predicted and passed to a convolutional layer with 3×3 filters. The resulting features are forwarded into two separate convolutional layers: the binary classification softmax layer that outputs *2k* values to represent the estimated probability of whether the corresponding anchor contains objects or not, and the regression layer that outputs *4k* values encoding the predicted bounding boxes coordinates for the objects detected in *k* anchors. The Fast R-CNN module takes the feature map produced by feature-extraction CNN and object proposals produced by RPN as inputs. For each object proposal, a Region of Interest (RoI) max pooling layer extracts information from the feature map and is then fed into fully connected layers to output the softmax probabilities for *m* categories plus 1 for background, as

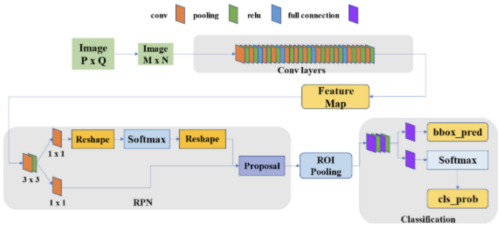

Figure 3. Architecture for Faster R-CNN. Image adapted from [13].

well as the refined four values for each of the per-class bounding box positions.

2.1.2. Mask R-CNN

Mask R-CNN [32] is another powerful two-stage model that builds on the architecture of Faster R-CNN for object detection and instance segmentation. Mask R-CNN extends Faster R-CNN by adding an additional branch for pixel-level segmentation. This branch generates a binary mask for each detected object in the image, indicating the exact pixels that belong to that object (Figure 4). This is particularly useful in scenarios where the objects in the image have complex shapes or overlap with each other.

The architecture of Mask R-CNN consists of four main components: a backbone network, a region proposal network, a detection network, and a mask prediction network. The backbone network is typically a pre-trained convolutional neural network (CNN) that extracts features from the input image. The region proposal network generates object proposals based on these features, which are then passed to the detection network.

The detection network classifies and refines the object proposals to generate the final object detections. It outputs bounding boxes for each detected object and a probability score indicating the likelihood that the object is present. The mask prediction network takes the feature maps generated by the backbone network and the region proposal network and produces a binary mask for each detected object. This mask prediction network uses a fully convolutional network (FCN) to produce pixel-level segmentation masks for the detected objects.

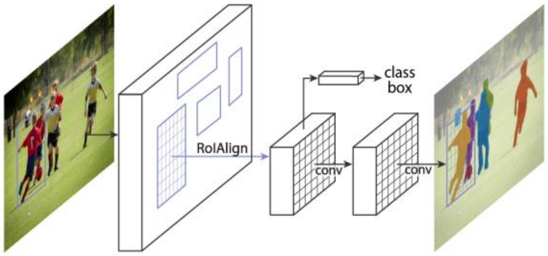

Figure 4. Architecture for Mask R-CNN [32].

2.2. Parking Space Detection

Exiting parking space detectors can be split into three main groups: Video-based, Image-based, and Sensor-based.

2.2.1. Video-based models

This method involves using cameras with high-resolution image sensors mounted on poles or buildings to monitor the parking lot. The cameras capture real-time video footage of the parking spaces, which can be analyzed using computer vision algorithms based on deep learning techniques, such as CNNs, to detect the presence of vehicles [37]. The algorithms are designed to extract low-level features to detect each parking place [30]. However, this method requires powerful computing resources, such as Graphical Processing Units (GPUs), to process the video feed in real time.

2.2.2. Image-based models

The image-based approach is another method that can be used to detect parking spots. They provide good information about the exact position and boundaries of each parking spot and are cost-effective.

An abundance of research has been done on parking space classification to determine parking space occupancy. In recently conducted research, the classification task is applied to predefined spots, meaning that the location of each spot is known to the classifier. “Image-Based Parking Space Occupancy Classification: Dataset and Baseline” [32] proposes a new dataset with a simple baseline for occupancy classification with a pre-knowledge of the spot location. “A Multi-Classifier Image Based Vacant Parking Detection System” [33] and “Real-time image-based parking occupancy detection using deep learning “ [34] are also prominent research in the classification task.

On the flip side, not much research has been conducted on image-based parking space detection using deep learning techniques. Some of them use image-processing techniques to do template matching, and find spots using their textures or shapes [30], and cannot work under different light and weather conditions. The most recent deep learning approaches show outstanding results on parking space detection while do not support spaces with special painted marks on the ground and cannot distinguish between different spots regardless of their occupancy.

2.2.3. Sensor-based models

Another method used for detecting parking spots is using various sensors, including Ultrasonic, Magnetic, and Infrared. Although each sensor has its own advantages, all of them are expensive and not cost-effective.

Ultrasonic sensors [29] use a time-of-flight method to measure the distance between the sensor and the vehicle. The sensor generates a pulse of sound waves and measures the time it takes for the waves to bounce back from the vehicle. The time delay provides information about the distance, which helps determine whether a parking space is occupied or vacant. The performance of these sensors can be easily affected by environmental factors, including air temperature, the functionality of the sensors, and/or any human destructive factors.

Magnetic sensors [35] detect changes in the magnetic field when a vehicle is in a parking spot. The sensors are based on magnetoresistive technology that measures the resistance of a magnetic material to a magnetic field. The sensor is embedded in the ground and can transmit information about the occupancy of a parking space to a central system. Although magnetic sensors are highly accurate and can detect the presence of any metal object, they can be affected by changes in the magnetic field caused by nearby buildings or power lines.

Infrared sensors [36] emit infrared beams to detect the presence of vehicles in a parking spot. These sensors measure the amount of reflected infrared light to determine whether a parking space is occupied or vacant. Infrared sensors can be integrated with other sensors, such as ultrasonic sensors, to improve accuracy. However, they can be affected by weather conditions, such as fog, rain, or snow, which can reduce their accuracy.

2.3. Disabled Facility Detection

Many deep-learning-based disabled facility detection studies are about identifying the accessibility of sidewalks or crosswalks. For example, Sun and Jacobs proposed a Siamese Network with convolutional layers that learn to focus more on the background and can localize 27% of the missing curb ramps in city streets on Google Street View [20]. In Wu et al.'s study, the accessible zebra crossings were detected by performing a sliding window detection on two CNN-based models to localize the empty crossing regions and estimate the direction of the crossings [31]. However, there is limited work on detecting disabled facilities in parking lots.

One of the most relevant studies for this report is Chau et al. 's on-street parking sign detection [8]. They collected 4,191 images of parking signs from street-level images with 27 categories, including handicap signs [8]. They used RetinaNet, YOLOv5, and Swin Transformer for real-time on-street parking sign detection. Their results showed that YOLOv5m was the best in terms of sign detection and could achieve 0.746 mean average recall and 0.804 mean average precision at 0.5 thresholds for diverse categories of signs on the test set [8]. However, the YOLOv5m only had 0.469 average recall and 0.727 average precision for detecting handicap signs, which was much lower than the overall performance, and the poor performance was attributed to the complex shapes and large image-to-image variations of the handicap signs [8]. It is also important to note that this study was about detecting traffic signs, which are often perpendicular to the ground and not easily visible from an aerial angle. Therefore, compared with handicap road mark detection, their work is less applicable for detecting disabled parking spaces from the top view.

Another relevant study is by Hanpinitsak et al. [6] on parking plot disability facility classification. They conducted transfer learning on pre-trained models, Inception-V3, Xception, and EfficientNet-B2, to classify whether the input image has handicap marks on the floor [6]. The EfficientNet-B2 yielded the best results, with a 0.956 F1-score on classifying the presence of handicap marks on the floor [6]. However, their studies also had some limitations. They never released the dataset to the public, making it difficult to assess the overall quality and nature of the images. Moreover, the handicap-presence images shown in the paper only contained one or two parking spots, with one single handicap mark located at the center of the image and having a size approximately equal to half of the image's size. Given that the handicap marks have limited variations in their dataset, and a classification task cannot provide the precise position of each handicap mark, a more comprehensive approach that can detect handicap marks with various sizes and locations is necessary to handle these limitations and provide more information about accessible parking places for people with disabilities.

3. Methodology

3.1. Space Detector Dataset

ACPDS dataset (Figure 5) was used to create training, validation, and test sets for the parking spot detector. This dataset contains 293 images with 11,236 unique views of a parking space taken by a GoPro Hero 6 action camera 12 meters above the ground in different parking lots and under different light (e.g., daylight and road lamp light) and weather conditions (e.g., raining, sunny, and foggy). Each image has a resolution of 4000×3000, and each parking spot is represented by a quadrilateral (4 coordinates) and its occupancy label (empty and occupied). We used 231, 35, and 27 images for the training, validation, and test sets. This dataset has not been fully annotated and has lots of unlabeled spots. Therefore, to further improve the model, we added additional annotations to the unlabeled parking spots following the annotation procedure mentioned in [21].

In addition to the quadrilaterals information, we provided x-, y-axis aligned bounding boxes to be able to work with both bounding boxes and quadrilaterals. These quadrilaterals were considered segmentation masks for parking spots. We further customized the dataset by converting the dataset into a COCO format [28]. This facilitated the evaluation of our model on the COCO standard benchmark and took the benefit of logging and some COCO utility functions.

3.2. Handicap Mark Detector Dataset

There is no publicly available dataset designed specifically for handicap road mark detection in parking lots. Therefore, we manually collected images of parking lots with handicap road marks from various sources.

3.2.1. Test Set

To work in parallel with the parking space detection task, the handicap mark detection test set also used images from the ACPDS Dataset. Since the goal was to evaluate the model's performance in detecting handicap road marks, only images containing handicap marks were selected as the test set images. In total, there were 37 images in the test set.

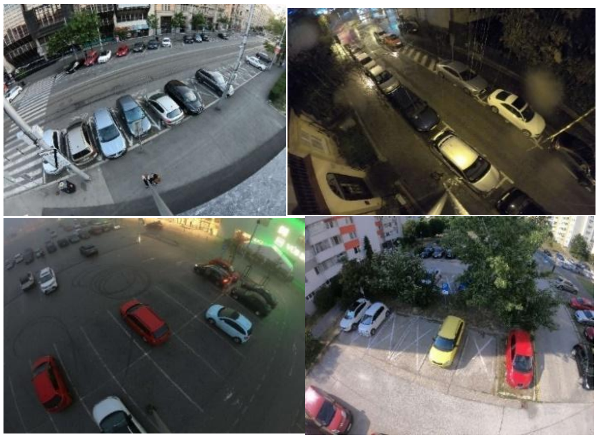

Figure 5. Images from Action-Camera Parking Dataset under different light and weather conditions.

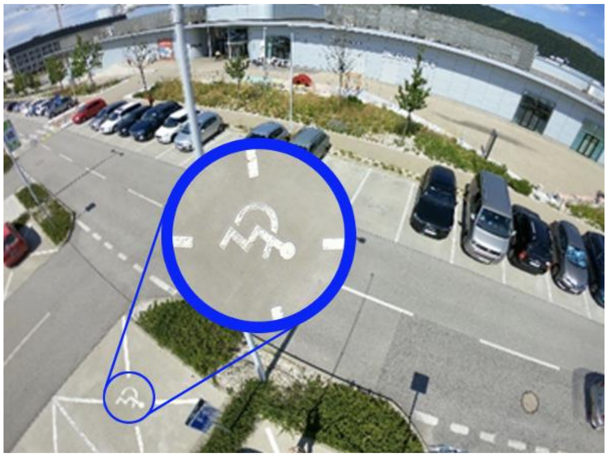

Figure 6. Sample images in the test set. Each image in the test set will contain at least one handicap mark.

3.2.2. Training and Validation Sets

Since the number of available handicap mark images in the ACPDS dataset was insufficient to train a deep-learning model, we collected a different set of images for the training and validation sets. To get a diverse representation of the parking lot images taken from different angles in different countries, 15 images were selected from the MTSD dataset [7], and 70 images were downloaded by entering relevant keywords on Google Search. These keywords included but were not limited to: "parking lots," "disabled parking lots," "CCTV camera monitor in parking lots," "parking tutorial," and "drone view in the city" in multiple languages, such as English, French, Japanese, and Slovak, and Czech. To mitigate the issue that the collected set lacked images with similar features as the test set, an additional 25 images were synthesized using Adobe Photoshop based on the images in the Action-Camera Parking Dataset. The final custom dataset contained images with varying resolutions ranging from 500×375 to 4000×3000. We then used a random seed to split these images into training and validation sets based on a ratio of 8:2, respectively.

Image Source	Count
MTSD dataset	15
Google Street View	13
YouTube video frames	17
Online websites	39
Synthesized	25
Total	109

Table 1. Training and validation image sources and count

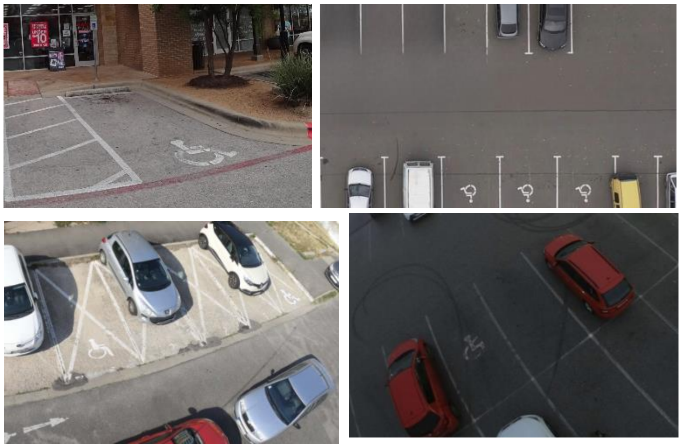

Figure 7. Top: sample images collected from the MTSD dataset and online. Bottom: synthesized images based on the Action-Camera Parking dataset. Each image contains at least one handicap mark.

3.2.3. Dataset annotation

Images in the training, validation, and test sets for handicap detection were manually labeled using LabelBox.com. Each handicap mark was labeled using one rectangular bounding box, with each side of the bounding box aligned with the edge of the handicap mark (Figure 8). Ten rounds of image and label quality checks were conducted to ensure the annotation quality met the requirement of deep learning model training. In total, there are 88 images with 225 handicap labels in the training set, 21 images with 42 handicap labels in the validation set, and 37 images with 109 labels in the test set.

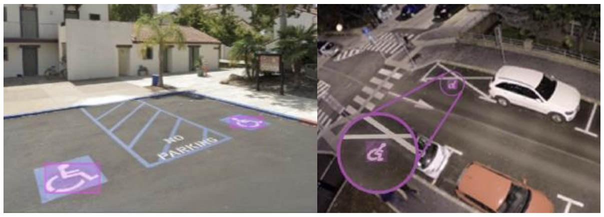

Figure 8. Samples of labeled handicap marks (pink rectangular bounding boxes) in the dataset.

3.3. Data pre-processing and feature engineering

Due to the limited computational resources, the input images in the dataset needed to be rescaled into 2400×1800 and 1280×1280 pixels for parking space and handicap mark detections, respectively (see Section 4 for details). However, downscaling images can lead to losing some useful information. To ensure that all handicap marks in the resized images were at least visible to human eyes, we eliminated the bounding boxes with areas of smaller than 300 pixels. Following this criterion, 11 bounding boxes from 3 images were removed in the handicap training/validation set, and 18 bounding boxes from 10 images were removed in the handicap test set. We did the same for the parking spaces with areas smaller than 3200 pixels.

During preliminary experimentation on handicap mark detection, it was observed that the detection model tended to mistakenly identify random blue patches as handicap marks in the validation set. It was possible that lots of handicap marks in the training dataset had blue backgrounds, and the model considered the background color as one of the important features in identifying the handicap marks and focused less on learning the structure of the handicap symbol itself. To address this issue, before training and evaluation, the blue color channel of each image was replaced by the grayscale version of the image using the following equation [22]:

$$\text{Blue} = \text{Red} * 0.299 + \text{Green} * 0.587 + \text{Blue} * 0.114 \quad (1)$$

The resulting images had a more reddish and greenish appearance (Figure 9), and the color blue would have less effect on the model's performance.

3.4. Model

3.4.1. Space Detector

In the aerial view of parking lots, parking spots were small and densely packed, resulting in the foreground and background class imbalance. In addition, due to the complexity and irregular shape of parking spaces, working with well-aligned bounding boxes resulted in a high overlap between each spot and its adjacents, making detection a challenging task. To address this issue, we used the Mask R-CNN [32] model to consider segmentation masks. This improved the detection performance, because by using only Faster R-CNN, we got the x- and y-axis aligned bounding boxes which included a larger area around each spot[1]. However, the actual spots were not well aligned and have orientations with respect to the x and y axes. So, predicting masks for parking spots yielded better performances.

We fine-tuned a Mask R-CNN to meet our needs in this particular application. ResNet50 backbone with Feature Pyramid Network (FPN) had shown to outperform others. In the segmentation task, class 0 is considered as a background. So, we added two additional classes representing empty and occupied spots as the model output, resulting in overall three output classes for the classification head.

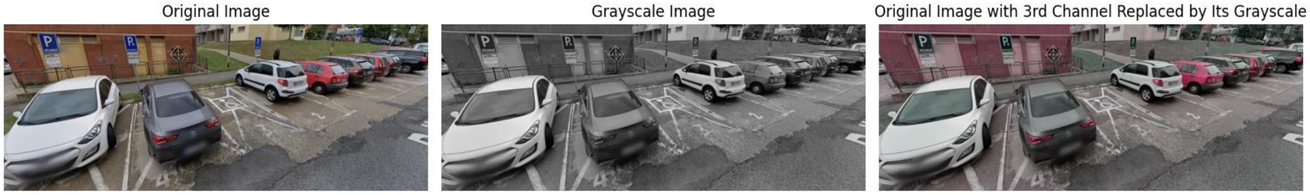

Figure 9. Example of images with the blue channel replaced by its grayscale version.

[1] In our preliminary experiments, we found that Faster R-CNN gives about 20% lower accuracy than Mask R-CNN in parking space detection tasks.

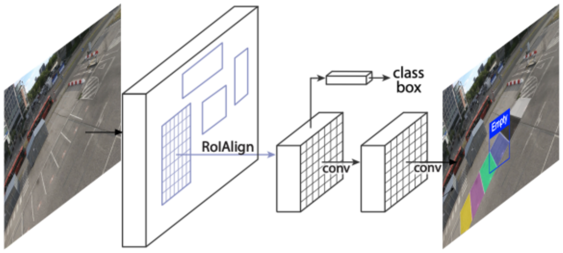

Figure 10. Mask R-CNN training pipeline, adapted from [32].

3.4.2. Handicap Mark Detector

Faster R-CNN was used for handicap mark detection. As mentioned earlier, the Faster R-CNN relies on the feature map generated by the CNN feature-extraction model for object classification and bounding box regression. As a result, selecting the feature-extraction model is a very critical step. After experimenting with various backbone and neck structures, we observed that the ResNet50 backbone with FPN neck yielded the most promising results[2].

3.4.3. Categorized Space Detector

The final module is a combination of the space detector and handicap mark detector modules. These modules performed their individual detection tasks in parallel and sent the outputs[3] to the processing module. The processing module took in the bounding boxes and/or segmentation masks along with the classified labels of the detected objects as inputs and output the final combined parking locations with the categories they belong to. There were two spot occupancy categories: empty and occupied spots, and two spot type categories: regular and disabled spots. In order to combine the detected parking spaces with the predicted marks, we computed the Intersection over Union (IoU) among the predictions to find the corresponding pairs. Finally, we filtered out the result based on the overlap of each predicted mark with the available spot.

4. Experiments

4.1. Setup

Models were trained and evaluated on 12th Gen Intel(R) Core(TM) i9-12900K CPU, 64 GB RAM, and Nvidia GeForce RTX 3060 graphics card.

4.1.1. Space Detector

We experimented with Mask R-CNN with ResNet50-FPN and ResNet50-FPN_V2 as the backbone and trained the models with different hyperparameters. Due to the small dataset, the pre-trained weights were used to boost detection performance and prevent overfitting. Regarding the learning setups, the AdamW optimizer [25] with an initial learning rate of 0.00008, a Cosine Annealing learning scheduler with a maximum number of 10 iterations, and a weight decay of 0.0005 yields the best results.

We tried with different image resolutions, and due to the resource limitations (such as memory), only images with a size of less than 2400×1800 could be worked with. To resize images, we also needed to resize the masks and bounding boxes to be consistent with the resized images. Hence, we defined a custom transform method to resize masks and boxes of each image along with the image resizing. In order to train the model with the highest resolution (2400×1800), we set the batch size to 1 to keep the information as much as possible. Experiments showed that the machine ran out of memory by increasing the batch size unless we decrease the image size more.

In addition, to further improve the model performance, we designed random augmentation methods (such as horizontal flip, rotation, photometric distort[4], etc.) that augmented images along with their annotations.

4.1.2. Handicap Mark Detector

We used the second version of the Faster R-CNN-ResNet50-FPN by [26]. The pre-trained weights were also used to mitigate the effect of the small dataset. Specifically, the ResNet50-FPN module used the pre-trained weights from the ImageNet1k dataset, and the Faster R-CNN head used the pre-trained weights from the COCO dataset. The trainable backbone layers were set to 2, meaning the first 24 layers of the ResNet-50 model were frozen, and the weights were not updated during backpropagation. The AdamW optimizer was used, with an initial learning rate of 0.0004, a Cosine Annealing Warm Restarts scheduler that reduces the learning rate every five epochs, and a weight decay of 0.0005. During preliminary experiments, we observed

[2] Other backbone models have been attempted, including but not limited to DarkNet, EfficientNet-B0, MobileNet-V3, ResNet18, and ResNet101. Unfortunately, due to limited time and computational resources, we could not report the results for these models.

[3] Outputs for the space detector are boxes and masks for each spot along with the labels and for handicap mark detectors are boxes and labels for each mark.

[4] Photometric distort, randomly changes image contrast, hue, saturation as well as the brightness.

that the model's performance on the validation set would either decrease or become stable after 25 epochs, and the machine would experience memory issues if the batch size was 11 or larger. Therefore, we set the number of epochs to 25, and the batch size to 10.

The input image size was 1280×1280, and every image in the dataset will be rescaled into this ratio. On each image, the task was to classify and localize the handicap marks from the background, so the number of classes was 2 (handicap and the background). Inspired by YOLOv4 [15], we performed mosaic augmentation on the training set by randomly combining each image with three other images in a certain ratio (see Figure 11). Other augmentation techniques to the original images, such as random shadow, random rain, flip, rotate, or scale, had been attempted. However, these methods resulted in either a similar or worse performance compared or combined with the mosaic augmentation. Therefore, these augmentations were removed during training.

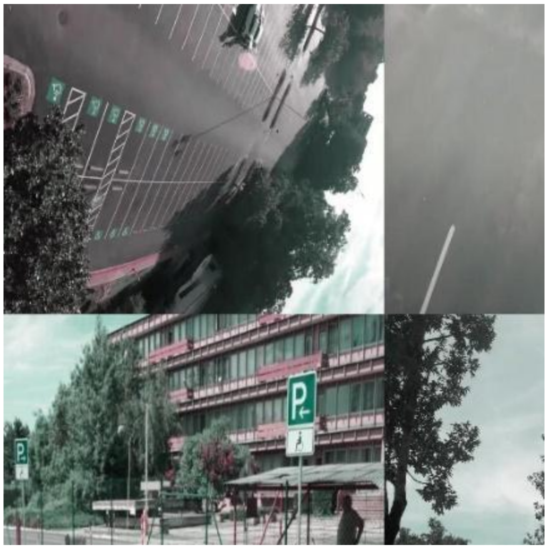

Figure 11. Training images with mosaic augmentation.

4.2. Evaluation metric

The model performance was evaluated using the COCO standard evaluation metrics. The Intersection Over Union (IoU) is calculated based on the overlapped and union areas between the predicted and the ground-truth bounding boxes:

$$IoU = \frac{Area\ of\ overlapping}{Area\ of\ union} \qquad (12)$$

The result is considered a correct detection (True Positive) if the detection with IoU is larger than or equal to a predefined threshold and considered an incorrect detection (False Positive) otherwise. The result is False Negative if the ground truth is not detected and True Negative if the bounding boxes are correctly not being detected.

The precision and recall are calculated by:

$$Precision = TP \,/\, the\ number\ of\ detections \qquad (13)$$

$$Recall = TP \,/\, the\ number\ of\ ground\ truths \qquad (14)$$

Holding everything else constant, as the detection threshold increases, the precision will increase while the recall will decrease. The precision-recall curve represents this trade-off between precision and recall under different detection thresholds, and the area under the precision-recall curve is defined as the Average Precision (AP). The mean Average Precision (mAP) is the AP averaged across all classes of objects and/or across various detection thresholds, with a higher mAP indicating better model performance. The Average Recall (AR) was also used in COCO metrics for model evaluation. It was computed by obtaining the recall values across 0.5 to 0.95 IoU thresholds and "averaging the largest recall values such that the precision is larger than zeros for each IoU threshold" [27]. Same as mAP, mAR represents the AR averaged across classes.

4.3. Model Evaluation

4.3.1. Space Detector

We evaluated the performance of the space detector on both validation and test sets. Figure 12 depicts the box and segmentation mAPs under different epochs during the training process. Tables 3 and 4 also report the model's performance on the unseen images (the test set). It can be seen from table 3 that the model achieves 0.563 and 0.473 mAP at 0.5 IoU, 0.300 and 0.229 mAP at 0.5:0.95 IoU, and 0.368 and 0.291 mAR at 0.5:0.95 IoU for boxes and masks, respectively, on the unseen data. Gaining higher accuracy in such crowds and highly packed scenes is very challenging and needs more time and complicated techniques in addition to the current ones. Therefore, this performance with this small dataset is a good result.

We also visualized predictions on some images to further inspect the model performance and see how reliable the model is. For each parking spot, we plotted both the bounding boxes and masks belonging to the spot. As shown in Figure 13 and 14, [5]masks tended to be more aligned with the real spot orientation than the bounding

[5] Due to technical issues, we cannot sure figures produced by Mask R-CNN soloy. We'll include them in the presentation.

boxes, meaning that they could provide more accurate information about the location of each parking space. Boxes and masks for empty spaces were colored green, and occupied spaces were colored red. In order to distinguish between regular and accessible spaces, we colored empty accessible spaces in blue and the occupied ones in red.

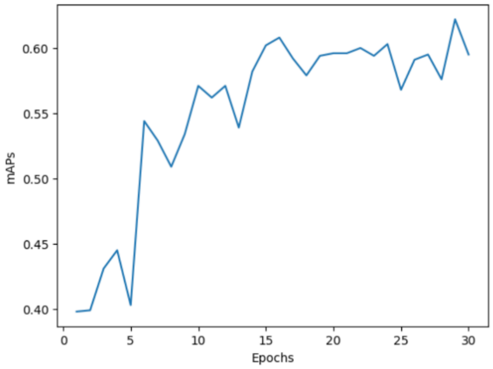

Figure 12. Box mAPs at IoU = 0.5 on the validation set as a function of epochs.

Box IoU metric	IoU	Object area	Validation Score	Test Score
mAP	0.50:0.95	all	0.334	0.300
mAP	0.50	all	0.622	0.563
mAP	0.50:0.95	small	N/A	N/A
mAP	0.50:0.95	medium	0.009	0.015
mAP	0.50:0.95	large	0.386	0.338
mAR	0.50:0.95	all	0.421	0.368
mAR	0.50:0.95	small	N/A	N/A
mAR	0.50:0.95	medium	0.046	0.061
mAR	0.50:0.95	large	0.473	0.411

Table 3. Model performance on the validation and test set (Box IoU metric in parking space detection). AP: mean average precision given at maximum 100 detections per image. AR: mean average recall given at maximum 100 detections per image. IoU: Intersection over Union of the predicted and ground-truth bounding boxes. Object area: small=the detected object's area is smaller than 32^2 pixels; medium=the object's area is between 32^2 to 96^2 pixels; large=the object's area is larger than 96^2 pixels. N/A: not applicable.

Segmentation IoU metric	IoU	Object area	Validation Score	Test Score
mAP	0.50:0.95	all	0.260	0.229
mAP	0.50	all	0.537	0.473
mAP	0.50:0.95	small	N/A	N/A
mAP	0.50:0.95	medium	0.005	0.004
mAP	0.50:0.95	large	0.318	0.267
mAR	0.50:0.95	all	0.339	0.291
mAR	0.50:0.95	small	N/A	N/A
mAR	0.50:0.95	medium	0.052	0.048
mAR	0.50:0.95	large	0.379	0.325

Table 4. Model performance on the validation and test set (Segmentation IoU metric in parking space detection).

	IoU	Object area	Validation Score	Test Score
mAP	0.50:0.95	all	0.617	0.501
mAP	0.50	all	0.83	0.732
mAP	0.50:0.95	small	0.407	0.308
mAP	0.50:0.95	medium	0.702	0.673
mAP	0.50:0.95	large	0.135	N/A
mAR	0.50:0.95	all	0.702	0.573
mAR	0.50:0.95	small	0.64	0.393
mAR	0.50:0.95	medium	0.762	0.733
mAR	0.50:0.95	large	0.133	N/A

Table 5. Model performance on the validation and test set (Handicap detection).

4.3.2. Handicap Mark Detector

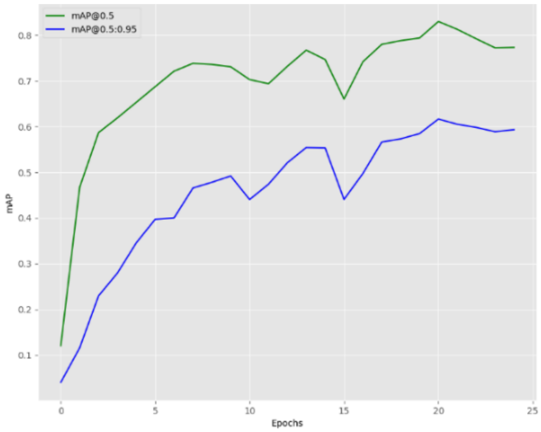

Figure 15. mAPs on the validation set as a function of epochs.

For the handicap mark detection model, Figure 15 shows the mAP at 0.5 IoU threshold and from 0.5 to 0.95 IoU thresholds (written as "0.5:0.95 IoU") in the validation set as a function of epochs. Table 4 represents the model's performance on the validation and test sets. As the table indicates, the model is able to achieve 0.83 mAP at 0.5 IoU threshold, 0.617 mAP at 0.5:0.95 IoU, and 0.702 mAR at 0.5:0.95 IoU in the validation set. Due to the differences between the training/validation and test set (e.g., the training/validation sets had images taken from various angles, while test set images were always taken from an aerial angle), the evaluation performance was lower but still acceptable for the test set. Specifically, the model achieved 0.732 mAP at 0.5 IoU, 0.501 mAP at 0.5:0.95 IoU, and 0.573 mAR at 0.5:0.95 IoU on the test set. Moreover, across both sets, the model's handicap mark detection performance was highest when the ground-truth handicap marks were between 32^2 to 96^2 pixels on the image, and the detection performance decreased significantly when the handicap mark sizes were larger than 96^2 pixels.

Furthermore, we investigated the performance of the model by inspecting the predicted images[6]. As shown in Figure 16, the model was able to detect the handicap marks in both validation and test sets in most cases. The two most common detection mistakes across images were detecting white patches with complex shapes on the floor as handicap marks, and being unable to detect handicap marks when they were partially covered by shadows (Figure 17).

[6] During the training and evaluation, each image was feature-engineered such that its blue color channel was replaced by its grayscale version. In this report, we predicted bounding boxes using feature-engineered images, then plotted these bounding boxes onto the original-colored images for better visualization.

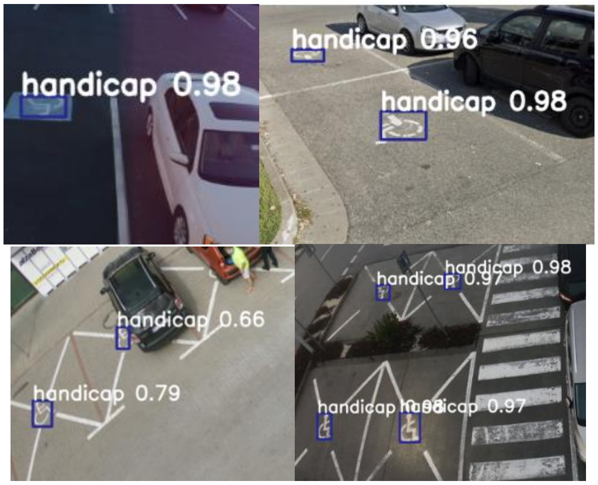

Figure 16. Handicap detection results in the validation set (top) and test set (bottom).

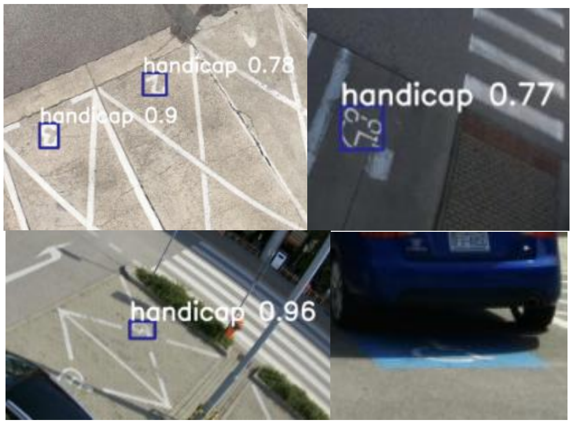

Figure 17. The model sometimes mistakenly considered white patches with complex shapes as the handicap marl (top) and was less likely to detect marks when the shadow covered some parts of the mark (bottom).

4.3.2. Categorized Space Detector

We also presented the results by combining segmentation results from the space detection task and the handicap bounding box results from the handicap detection task. Unfortunately, this is very preliminary, and quantitatively evaluating the performance of the combined results was rather challenging. Therefore, only the produced images were provided. See Figure 18.

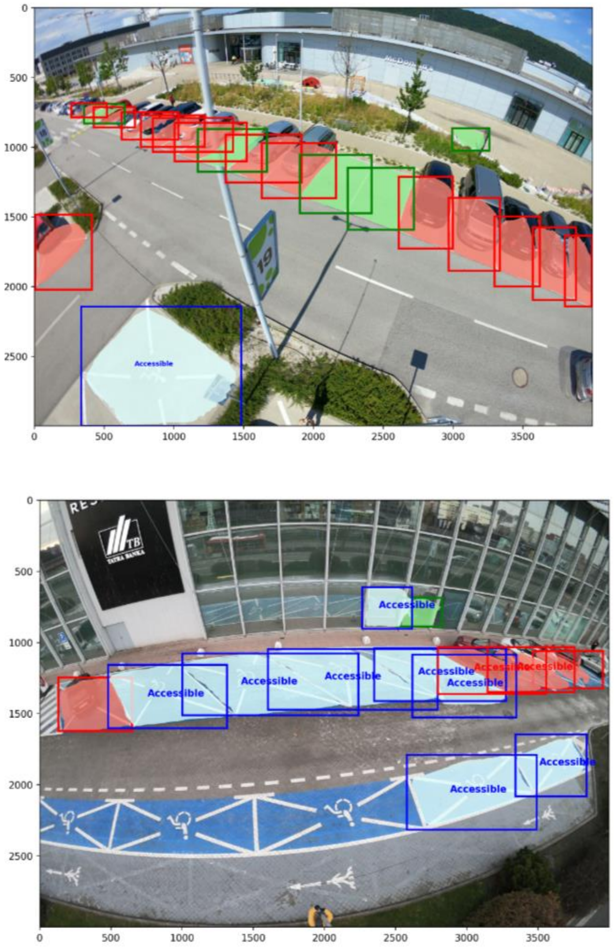

Figure 18. The combined results.

5. Conclusion

Parking space detection, along with disabled facility detection, is crucial for efficiently managing parking lots and helping drivers find suitable empty spots. However, there is a limited amount of research that considers both tasks. This study fills this gap by conducting parking space detection, introducing a new dataset explicitly designed for handicap mark detection in parking lots, and testing the feasibility of detecting parking spaces and painted handicap marks using R-CNN family models. We could achieve 0.622 and 0.83 mAP at 0.5 IoU on the validation set and 0.563 and 0.732 mAP at 0.5 IoU on the test set for the space and mark detectors, respectively.

Due to the small dataset, the density of the scene, the smallness of the parking spots, and the angles of each parking spot in the aerial images, achieving a high score is challenging. We could successfully address the problem of spot orientation by working with masks along with the bounding boxes. In the future, one can leverage oriented bounding boxes instead of well-aligned bounding boxes to improve the model. Besides, training the model using a larger dataset would increase the performance.

There are several limitations in this handicap mark detection task. Firstly, the model detected the handicap marks under the assumption that there was at least one handicap mark in each image. However, according to Americans with Disabilities Act regulations [3], the ratio between disabled parking spots and the total number of parking spots is 1:25, meaning that the chances of encountering a handicap mark in a given CCTV camera image are much smaller. Future work should take the uneven distribution between the disabled parking spots and normal parking spots into consideration by introducing images with no handicap marks into the dataset. Secondly, the model considered white and complex patches on the floor as handicap marks and was not able to detect handicap marks when the shadow covered parts of the mark. This was likely due to the small dataset, and most images in the train/validation set are taken under daylight with no shadows or occlusions. Future studies should also add images with different weather and light conditions to the parking lot handicap mark dataset.

In conclusion, the current study shows promising results in detecting regular and accessible parking spaces in parking lots using deep-learning-based approaches. With further development, future technologies could lead to more efficient parking lot management and provide more accessible parking solutions for people with disabilities.

6. Contributions

Kimia took care of the parking space detection. Tasks include conducting literature review, programming, experimenting and training space detection models, visualizing results, and writing her task’s procedure in the paper.

Yuchen took care of the handicap mark detection. Tasks include conducting literature review, creating a new dataset and annotating images, experimenting and training handicap detection models, visualizing results, and writing her task’s procedure in the paper.

Moreover, Yuchen took charge of writing a larger portion of the paper, while Kimia took charge of merging, analyzing, and presenting the combined results (parking space + handicap detection).

We’d also like to thank Dr. Beyeler and grader Xinlei for providing us with valuable suggestions and feedback along the way!

References

[1] “Disability,” *World Health Organization*, 07-Mar-2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/disability-and-health. [Accessed: 20-Mar-2023].

[2] Srdjan Tegeltija, Mladen Babić, Laslo Tarjan, Igor Baranovski, Goran Stojanović. One solution for validation of legal usage of reserved parking spaces for people with disabilities. *2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2021.

[3] “Accessible parking spaces,” *ADA.gov*, 17-Mar-2023. [Online]. Available: https://www.ada.gov/topics/parking/. [Accessed: 20-Mar-2023].

[4] Paulo Almeida, Luiz S Oliveira, Alceu De Souza Britto, Eunelson J Silva, and Alessandro L Koerich. Pklot-a robust dataset for parking lot classification. *Expert Systems With Applications*, 42(11):4937–4949, 2015

[5] Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Carlo Meghini, Claudio Vairo. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, vol. 72, pp. 327–334, 2017.

[6] Panawit Hanpinitsak, Pitiphum Posawang, Sumate Phankaweerat and Wasan Pattara-atikom. Method for image-based preliminary assessment of car park for the disabled and the elderly using convolutional neural networks and transfer learning. *Multi-disciplinary Trends in Artificial Intelligence*, pp. 99–110, Nov. 2022.

[7] Christian Ertler, Jerneja Mislej, Tobias Ollmann, Lorenzo Porzi, Gerhard Neuhold, and Yubin Kuang. The mapillary traffic sign dataset for detection and classification on a global scale. In *ECCV*, 2020.

[8] Hieu Chau, Yin Jin, Jiayu Li, Juhua Hu, and Wei Cheng. Real-Time Street Parking Sign Detection and Recognition. In *IJCAI-ECAI 2022 Workshop + Challenge*, 2022.

[9] David G Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[11] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *CVPR*, 2018.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[13] Wenze Li. Analysis of Object Detection Performance Based on faster R-CNN. In *Journal of Physics: Conference Series*, vol. 1827, no. 1, p. 012085, 2021.

[14 Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016

[15] Alexey Bochkovskiy, Chien-Yao Wang, and HongYuan Mark Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv:2004.10934 [cs, eess]*, Apr. 2020. arXiv: 2004.10934.

[16] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[17] Ross Girshick. Fast R-CNN. In *ICCV*, 2015

[18] Jwyang. Jwyang/Faster-rcnn.pytorch: A faster pytorch implementation of faster R-CNN, *GitHub*. [Online]. Available: https://github.com/jwyang/faster-rcnn.pytorch. [Accessed: 20-Mar-2023].

[19] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2015.

[20] Jin Sun, David W. Jacobs. Seeing what is not there: Learning context to determine where objects are missing. *In CVPR, 2017*.

[21] Martin Marek. Image-based parking space occupancy classification: Dataset and baseline. 2021, arXiv preprint *arXiv:2107.12207*.

[22] “Image module,” *Pillow (PIL Fork)*. [Online]. Available: https://pillow.readthedocs.io/en/stable/reference/Image.html#PIL.Image.Image.convert. [Accessed: 20-Mar-2023].

[23]] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016

[24] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[26] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. In *preparation*, 2021.

[27] Rafael Padilla, Wesley L. Passos, Thadeu L. Dias, Sergio L. Netto, and Eduardo A. da Silva. A comparative analysis of Object Detection Metrics with a companion open-source toolkit. *Electronics*, vol. 10, no. 3, p. 279, 2021.

[28] “Common objects in context,” *COCO*. [Online]. Available: https://cocodataset.org. [Accessed: 20-Mar-2023].

[29] Wan-Joo Park, Byung-Sung Kim, Dong-Eun Seo, Dong-Suk Kim, and Kwae-Hi Lee, Parking space detection using ultrasonic sensor in parking assistance system. *2008 IEEE Intelligent Vehicles Symposium*, 2008.

[30] Yong Ma, Yangguo Liu, Lin Zhang, Yuanlong Cao, Shihui Guo, and Hanxi Li. Research Review on parking space detection method. *Symmetry*, vol. 13, no. 1, p. 128, 2021.

[31] Xue-Hua Wu, Renjie Hu, Yu-Qing Bao. A regression approach to zebra crossing detection based on convolutional neural networks. *IET Cyber-Systems and Robotics*, Volume 3, Issue 1 p. 44-52, 2021

[32] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017

[33] Junzhao Liu, Mohamed Mohandes, Mohamed Deriche. A multi-classifier image-based vacant parking detection system. In *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, pages 933-936, 2013.

[34] Debaditya Acharya, Weilin Yan, and Kourosh Khoshelham. Real-time image-based parking occupancy detection using deep learning. In *proceedings*, 2018.

[35] Jörg Wolff, T. Heuer, Haibin Gao, Michael Weinmann, Stefan Voit, U. Hartmann. Parking monitor system based on magnetic field sensor. In *2006 IEEE Intelligent Transportation Systems Conference*, 2006.

[36] A. S. Bokhari. Smart parking management for electrical vehicles: solar parking lots. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2022.

[37] Bill Yang Cai, Ricardo Alvarez, Michelle Sit, Fabio Duarte, and Carlo Ratti. Deep Learning-Based Video System for Accurate and Real-Time Parking Measurement. *IEEE Internet of Things Journal*, 6(5):7693–7701, 2019.