ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W4-2024
8th International Conference on Smart Data and Smart Cities (SDSC), 4–7 June 2024, Athens, Greece

Machine Learning Based Mobile Capacity Estimation for Roadside Parking

Aruscha Kramm [1], Eric Peukert [1], André Ludwig [2], Bogdan Franczyk [2]

[1] University of Leipzig, Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig, Germany - (kramm, peukert)@informatik.uni-leipzig.de

[2] University of Leipzig, Information Systems Institute, Germany - (ludwig, franczyk)@wifa.uni-leipzig.de

KEY WORDS: street view images, aerial images, parking, geo-referencing, machine learning, digital twins

ABSTRACT:

The growing number of cars and limited street space present significant challenges for cities, applying not only to moving but extending to stationary traffic. The quest for parking spaces exacerbates traffic congestion, noise, and air pollution, particularly in residential areas. To develop effective parking solutions for these challenges, a trustful data foundation on available parking space capacities, its usage and parking type is crucial. Gathering this data is currently time-consuming, requiring manual labeling and street inspections. Moreover, it must be repeated to keep the data current. Research on parking space management has heavily focused on monitoring designated parking lots with fixed cameras to identify free or occupied parking spaces. However, due to privacy concerns fixed cameras are not applicable for the larger part of the street space in European cities. This paper introduces a novel computer vision-based method for automatically collecting parking space capacities and parking type information. Our approach combines both street view and aerial imagery, which are recorded by a moving camera source. We tackle challenges in geo-referencing images, identifying parking types, classifying moving and stationary cars and dealing with partial occlusions in images. By not permanently recording the same environment, our approach lowers the surveillance risk, making parking capacity estimation scalable. We conduct a thorough evaluation of our methods and release a novel validation data set to allow for further research. In the future, more moving camera sources will be available when attached to city cleaning vehicles or to delivery drones.

1. INTRODUCTION

The increasing numbers of private passenger cars are progressively occupying an increasing amount of space both in moving and stationary traffic which leads to an insufficiency of the available space [Lutz, 2014]. In dense urban spaces especially the search of parking space contributes to the already existing traffic, noise and air pollution. In areas where parking pressure is high, meaning many cars competing for few spaces, an unsuccessful search often leads to improper parking on sidewalks, intersections or exits where the curb is lowered [Parmar et al., 2020]. In addition to this, people like to park on the roadside because of its proximity to their destination (e.g. home) which aggravates the problem [Biswas et al., 2017]. This poses challenges for individuals with mobility aids, such as people with wheelchairs or walkers, as well as families with strollers when navigating neighborhood streets. Moreover, the increased traffic from parking searches amplifies the risk of accidents by irritated drivers [Bezirksamt Berlin-Neukölln, 2023]. To effectively address this issue in urban planning, it must be known in which suburbs parking pressure is particularly high to be able to create corresponding concepts and solutions. In order to calculate the pressure, comprehensive data on the number of roadside parking spaces must be available. As the number of parking spaces in a street depends on the type of parking, the different types of parking *longitudinal, diagonal, and perpendicular* must be distinguished. With this parking information per street, the number of potential parking spaces in residential areas can be estimated by using the street length and average car widths and lengths.

Parking surveillance has been thoroughly studied, with the main objective to recognize whether a parking space is free or occupied. In most of these cases vehicle detection in image data is utilized to monitor parking lots using fixed-location cameras [Fahim et al., 2021]. While a camera-driven approach can be applied to the recognize parking spaces on the roadside, a high density of cameras would be required for wide-area coverage introducing privacy issues such as the risk of surveillance. In addition, most studies only distinguish between free and occupied spaces, but cannot distinguish the type of parking which is required for the estimation.

We present a novel approach that assigns the type of parking to each street side of a street. We do so by training neural networks to detect cars and distinguish between the different parking types in two different images sources. By using street view imagery, we utilize a mobile data source that moves through the streets, taking images every three meters. By moving continuously, this data source covers a wide area which allows for our approach to scale to the entire city while keeping privacy concerns low. In the future moving camera sources can be created by attaching cameras to delivery drones or other city vehicles such as buses or cleaning trucks. To increase quality, we add another modality and within each street compare the street view images to the aerial image. Since street view images are taken every three meters and are compared to one aerial image of the street, one of the challenges is to find a suitable method for this one-to-many comparison. For the comparison it is necessary to assign detected cars to one side of the street while ensuring that the street side is the same in both images. When deciding for a parking type, the method has to tackle various difficulties such as moving cars, construction sites or no-parking zones which is why this paper initially focuses on the main challenges of the approach. The code to our work is publicly available. [1]

[1] `https://github.com/aruscha-k/roadside-carparking-direction-detection.git`

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W4-2024-99-2024

The main contributions of this work include:

1. We train neural networks to detect and differentiate the parking type of parking cars.
2. We consolidate two image sources based on their location (geo-referencing).
3. We compute street sides and assign a parking type per side to each street.
4. We conduct a thorough evaluation to identify strengths and weaknesses and to inspire future work.
5. We publish an excerpt of our result data to trigger further research.[2]

2. PROBLEM DEFINITION

We have tested, that using neural networks for the recognition of parking cars in images while at the same time detecting the parking type works reliably with average correct recognition rates above 90 %. In doing so, the neural network encounters challenges that many similar use cases in computer vision also face, such as shadows, various lighting conditions, or obstructions by trees (see Fig. 1a and b). As a consequence, some cars in the image remain unrecognized. More problematic is that the neural net cannot distinguish whether the car is parked or moving. It will therefore incorrectly assign a parking type to moving cars. One of the challenges therefore is to distinguish moving from stationary cars on these images (see Fig. 1a and c, red dashed rectangle). Within this initial work, we exclude multi-lane roads since these introduce further challenges. We were provided with a data set of edges of street segments of Leipzig, subdividing streets at every intersection with an average segment length of 138 meters (see Fig 3). Within one of these segments, the parking type can change (see Fig 1a and d, blue/yellow dashed line). In order to assign all correct parking types to a segment, it must be divided into smaller units. Therefore a suitable division unit must be found.

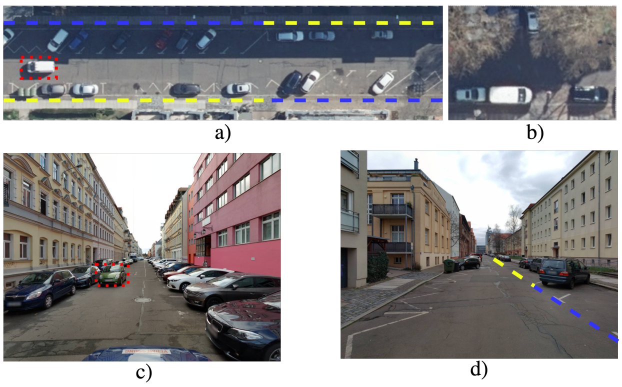

Figure 1. a) Aerial image of a segment with changing parking type (blue/yellow), a driving car (red) and different lighting conditions b) Aerial image with obstruction by trees. c) Street view recording with driving car and d) changing parking type

Our multi modal data set consists of *geo-referenced* (see chapter 4.1) street view and aerial imagery of Leipzig. First, a crucial task involves identifying all the images associated with a specific segment. Further, having a multi modal data source involves comparing two perspectives: street view images, which are taken approximately every 3 meters, and images taken from the air. To compare the results of object detection in both types of images, the detected objects must be assigned to the same sides of the street. Due to the different recording modalities of the image types, regardless of which subdivision of the segment is made, there will be $n > 1$ street view images meeting one aerial image (see Fig. 8). The challenge here is to find a suitable method to compare one to n images, while also managing scenarios where the comparison yields different results between the two image types. It should also be noted that the parking type can change over time. As both image sources are not recorded at the same time, the detection from the neural net can be correct and the result can still be different in both image types (see Fig. 2, orange dashed line). Another challenge we faced, was the verification of our results. Due to the novelty of the problem, there was no ground truth data available. We manually labelled the parking types of two larger residential areas and compiled a validation set.

Figure 2. Cut out of the same segment with aerial image (left) from 2019 and street view image (right) from 2021 showing different parking types on the right side

3. RELATED WORK

Detection of cars in aerial and street view image data in urban areas has already been explored on a large scale with first works dating back to 2003 [Zhao and Nevatia, 2003]. In these, the goal of detecting cars is the recognition of vacant or occupied parking spaces, as well as the transmission of the information to drivers looking for parking spaces. In most use cases in this area, the data source is in a fixed position (e.g. camera on a pole) observing a selected area and transmitting real-time data. To be able to estimate the capacity for roadside parking, it is not sufficient to only recognize cars in images, further their parking direction must be identified. Extending the analysis to a large area or whole city is hardly possible with a data source in a fixed position. To the best of our knowledge, there is little research on either using a moving data source or analysing the parking space regarding the parking method in particular (longitudinal, diagonal, or perpendicular parking).

The work of Fahim et al. [Fahim et al., 2021] provides an overview of different smart parking systems (SPS). The authors list various technologies used for this application such as ML methods, GPS or Bluetooth. All of the mentioned works (in total 54 studies) monitor parking lots, both inside and outside. One common goal of most of the studies is to categorize parking spaces into two classes: free or occupied.

Most research focuses on the surveillance of parking spaces through vision-based technologies like image or video data. These approaches typically involve cameras providing real-time data from fixed locations. Vision-based applications face challenges

[2] dataset available at: `https://github.com/aruscha-k/roadside-carparking-direction-dataset.git`

in handling varying light conditions, weather, shadows, and occlusions (e.g., by trees, street lamps). Primarily utilized for parking occupancy detection, these approaches also extend to additional use cases such as license plate recognition, facial recognition for billing, and generating traffic congestion reports [Fahim et al., 2021]. Examples of vision-based approaches recognizing the free or occupied state are the work of Masmoudi et al. who implement parking lot monitoring based on feature extraction in video data using the SURF and HOG algorithm [Masmoudi et al., 2014] or the work of Amato et al. [Amato et al., 2017] who use convolutional neural networks (CNNs).

While most studies focus on the surveillance of parking lots, the work of Goren et al. [Goren et al., 2019] uses a fixed camera positioned on the roadside to recognize parking cars and inform drivers about free spaces. The authors also concentrate on distinguishing free from occupied parking spaces. The work only scales with sufficient coverage of roads by cameras.

Apart from image data, LiDAR sensors are used for parking monitoring as well. LiDAR is a form of three-dimensional laser scanning that does not produce any image material. The research of Chen et al. [Chen et al., 2023] uses a stationary LiDAR sensor positioned on the side of the road supplying real-time data. The authors claim that an advantage of LiDAR sensors is that no image material is recorded making the data more privacy-friendly.

By training neural networks to recognize additional parameters (such as restricted areas or markings) in addition to cars, similar methods can be used to detect parking in unauthorized zones such as the work of Peng et al. [Peng et al., 2022].

4. METHODS

We trained two neural networks for object detection in both aerial images and street view images. For this task, we used *detectron2* [Wu et al., 2019] which is a state-of-the-art network for object detection and segmentation. Each trained model is capable of detecting cars along with their parking type *longitudinal, diagonal* or *perpendicular* parking. Due to the high similarity of diagonal and perpendicular parking in street view images, the model for street view images considers these two types as one class. The differentiation between diagonal and perpendicular is then made using the more precise detection from the aerial image, provided that this network has reached the same result. The output of both neural networks is bounding boxes around detected cars and corresponding classes. We did not evaluate the relationship between CNN parameters and recognition accuracy since the correct recognition accuracy of our trained networks already lies between 93 to 100%. The networks were trained by creating a data set of manually labeled parked cars in street view and aerial images. Due to the different orientations of streets and the two possible ways of navigating through a street, a fixed definition of which side of the street is left and which is right must exist. If it is clear which side in the image is the left/right side of the street, the bounding boxes found can be assigned to one of the sides. This makes it possible to compare the results from both image types for each side of the street. The final step is to merge the results from both image sources into one result per side.

4.1 Data set

The city administration of Leipzig provided us with *360° panoramic street view imagery* with a recording for approximately every three meters which is accessible via API. Our second image source consists of orthophotos, which are images depicting a bird's-eye view of the city that have been rectified in the ground plane [GeoSN Saxony, 2024]. The images have a resolution of ten centimeters per pixel (see Fig. 2 for examples). Both image types are *geo-referenced*: each street view image has an assigned recording point coordinate and in each air image, each pixel is mapped to a coordinate. We were further provided with street edges and nodes. In this data set, a street is subdivided into segments which means a subdivision at each intersection. Each street segment is defined by two endpoints in coordinates, as well as the median width of the street. In the process, each segment is subdivided into iterations (see Fig. 3, red dashed lines).

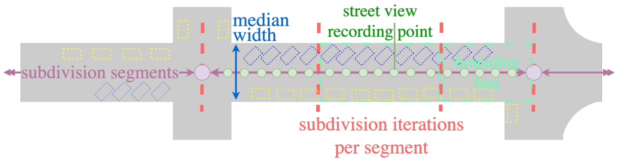

Figure 3. Data set of street segments and iterations. A segment (purple) is subdivided into iterations (red). A bounding box (turquoise) can be calculated using the endpoints and the median width (blue).

4.2 Aerial image extraction and iteration units

For every street segment, we use its two endpoints (x_{start}, y_{start}), (x_{end}, y_{end}) and its median width w to compute a bounding box $b = [(x_1, y_1), .., (x_4, y_4)]$ around the street with the following formula[3]:

$$[(x_{start}, y_{start} - \frac{w}{2}),\ (x_{start}, y_{start} + \frac{w}{2}),\\ (x_{end}, y_{start} - \frac{w}{2}),\ (x_{end}, y_{start} + \frac{w}{2})] \quad (1)$$

The resulting bounding box coordinate points can be used to cut out a segment from the geo-referenced aerial image directly using various Python libraries. Using the *WFS-API* of the street view image provider, we conduct a geospatial mapping of point-in-box to extract all street view recordings lying within this bounding box which results in a list of recording points. As mentioned in chapter 2, a segment has an average length of 138 meters. But, there are large differences within the lengths, with the smallest segment length being 7 meters and the longest 2539 meters (see Fig. 4).

It is therefore possible, that there is more than one parking type on one side within one segment. Further, if the segment length is used for cutting out the images, the varying lengths lead to a variety of image sizes. This in turn leads to poor neural network results since the size of the cars in the images also varies with the size of the images. Therefore, as a preprocessing step, each street segment is further subdivided into sections with a manually chosen length of 30 meters called *iterations* (see Fig. 3). These iterations are generated by dividing the segment every 30 meters, resulting in around five to ten evenly-sized iterations for each segment. To cut out the iteration-sized images from the aerial image, for each iteration, its bounding box is computed as described before (see formula 1).

[3] The formula is simplified for explanatory purposes, as it is only applicable to streets that run parallel to the vertical and horizontal axes.

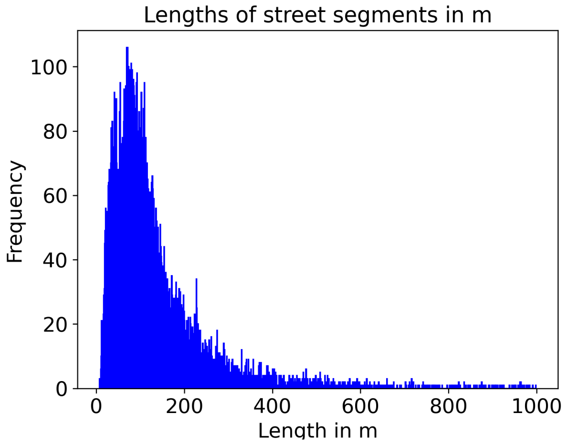

Figure 4. Distribution of lengths of street segments in meters. Outliers > 1000 m are cut off.

4.3 Street view image extraction and side assignment

To compare the results of the detections from both image types, each detected car must first be assigned to one side of the street. We want to assign the detected bounding boxes defined as $bbox = [(x_1, y_1), (x_2, y_2)]$ to the left or right side using their x-values and the image midpoint (mid_x, mid_y) (see Fig. 5):

$$\forall[(x_1, y_1), (x_2, y_2)] : \text{if } \forall i \in \{1, 2\}, x_i < mid_x \text{ left, else right} \tag{2}$$

Cars recognized on the left-hand side of the image are thus assigned to the left-hand side of the street and vice versa (see Fig 5).

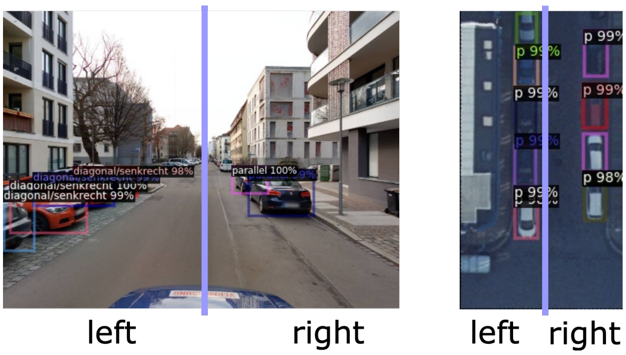

Figure 5. Street sides in street view and aerial images with violet line representing image center.

To ensure that the image is aligned so that the left side of the image is also the left side of the street, further measures must be taken. In the case of street view images, the street side assignment must therefore be known prior to image extraction for the images to have the correct alignment.

Since there is no universally valid definition of which side of a street is the left and which is the right, we define a reference point from which the sides of the street are viewed. This reference point is the city center. We define that the endpoint of a segment closer to the city center is the starting point and the endpoint further from the city center is the ending point of a segment. We consider street sides as left or right as if we had navigated through the street from its starting to its ending point (see Fig. 6a).

Because the street views are 360° panoramic images, the recording direction of the camera must be set when extracting the images. If the recording direction is not set, the image is automatically output in the direction of travel. However, the direction of travel does not always correspond to our definition of the starting and ending point of the street (see Fig. 6b and c). For each recording point, its recording direction is known. The recording direction displays the driver's viewing direction and is expressed as deviation from North as angle in degrees (see Fig. 6b and c). To set the recording direction of the camera so that the image is correctly aligned, we also compute the deviation from North of the street from the starting point to the ending point (see Fig. 6a). If the angles of the deviations are the same, the recording direction corresponds to our definition. If not, the image must be extracted against direction of travel, which can be done by adding 180 degrees to the recording direction.

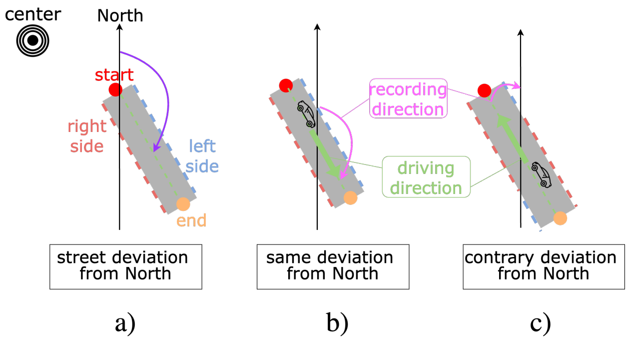

Figure 6. Deviation from North in street view images. a) Deviation from North for street b) Deviation from North of street is the same as the recording direction of the car. c) Deviation from North of street is contrary to the recording direction of the car.

Segments or iterations in aerial images are cut out from the overall image (see Fig. 7a). For our methods to process these images, they have to be rotated so that their edges are aligned parallel to the horizontal and vertical axis. That means the edges have to align with the display screen. To apply our street side definition to the aerial images and be able to assign the bounding boxes, the images must be rotated so that the left side of the image also represents the left side of the street and vice versa. To achieve both, we rotate the ending point of the street towards North (see Fig. 7b and c). The angle of rotation is again the deviation from North of the street.

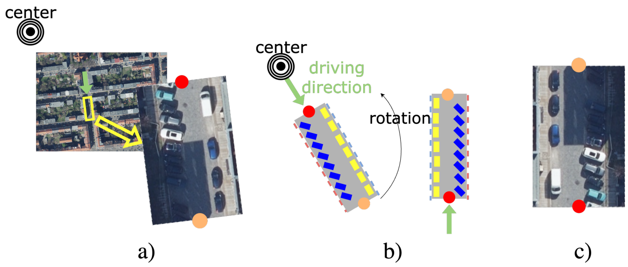

Figure 7. Rotation of aerial images.

https://doi.org/10.5194/isprs-annals-X-4-W4-2024-99-2024

4.4 Differentiating between driving and parked cars

To solve the problem of driving cars described in chapter 2 (see Fig. 1), driving cars must be recognized and excluded from further computations. To do so, we tested two methods. Assuming that parked cars will mostly be on the left and right edges of an image we experimented with clustering cars on each side. We computed the center points of all bounding boxes of detected cars and applied k-means clustering (with k=2) to sort out detected cars that do not fall within one of the clusters. Problems with this approach occurred with images, where there is parking only on one side of the street, which is why the approach with two fixed clusters does not work correctly. Other problems are uneven distribution of parked cars on each side, leading to skewed clusters and the false exclusion of cars. Methods to determine the optimal cluster size, such as *shilouette*, are not practical because they only work for a cluster size $>= 2$. The second method we tested is to insert a black area in the image (*no-detection area*) prior to object detection in which no cars are to be recognized. In street view images this area is added by drawing a black triangle into the center of the image and in aerial images this area is added by drawing a black rectangle starting from the centre of the picture with a flexible width of 15% of the image width. This naive method already provided good results. However, in future work we consider to train a separate classifier to distinguish moving from parking cars that takes the context information in the image as well as geo-spatial information about the street boundaries into account.

4.5 Result merging of detection results

The last step of the process is to evaluate all detection results to define one parking type per iteration and street side. For each iteration of a street segment, results from one aerial image need to be compared to several street view images (see Fig. 8). In a first step, the merging of detections is performed for each image type and street side separately.

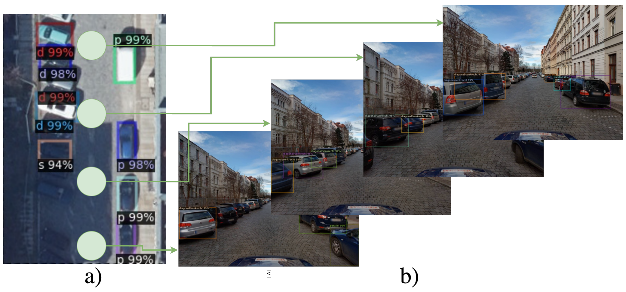

Figure 8. Two different perspectives on one iteration step of a segment a) aerial image with recording points, b) street view images of different recording points within the same iteration step.

Within each iteration step, all detected classes are assigned to one side of the street (see chapter 4.3), resulting in a list of detections d =[(*detected_class_1, detected_class_2*), ...] per side. To achieve one result per side, a majority decision is made on all entries in the list based on frequency of the detected classes and the most common class and an average class probability is saved. The average probability is computed by calculating the proportion of the most frequent class of all entries. At the end of this first step, for each image type the parking type is determined for each side of the street based on iterations (see Figure 9) resulting in a list of results r =*[(iteration: 0, left: (parallel, 99%), right: (diagonal, 87%)), (iteration: 1, left: ...]* for each image type. In a second step, the results from the different image types have to be compared to achieve one result per side for each iteration. The results from step one are therefore again compared using majority vote per iteration and side. If both image types came to the same resulting class, this class is chosen as result. If they produced different results, the class with the highest probability is chosen. If both image types conclude to different classes with the same probability, both results are saved and are marked as ties.

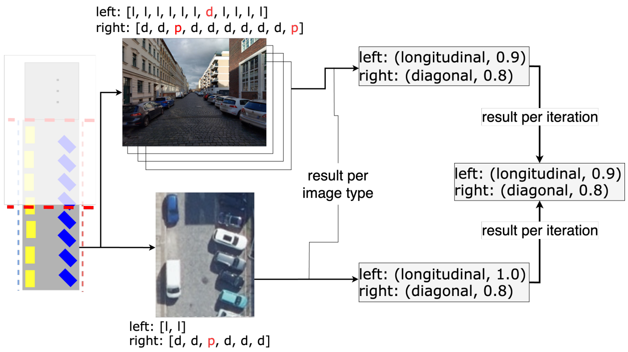

Figure 9. Merging detection results for each image type and iteration. Merging is first done on each image type separately before merging this interim result into a final one. Red letters represent different detection classes.
l - longitudinal, p - parallel, d - diagonal

5. EVALUATION

For the evaluation of our methods, we conduct several experiments in two districts (see Fig. 10).

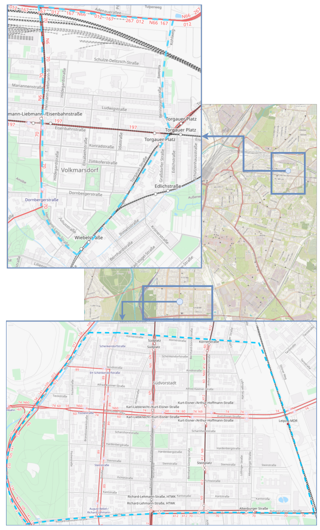

Figure 10. The study area consists of two districts, where district 1 can be seen on the lower excerpt and district 2 on the upper excerpt.

We look at

- results for both image types combined
- results for both image types separately
- results for both image types with and without *no-detection area*

- result differences for the two units segment or iteration
- results with ties, where two results were equally likely

In the process, we evaluate 338 segments in district 1. These segments are subdivided into iteration steps resulting in 1,220 iterations in district 1. We analyze a second district regarding iterations with a number of 998 iterations. Looking at the number of analyzed images, the number of aerial images is equal to the number of segments or iterations. The number of street view images analyzed for iterations in district 1 is 3,337 and in district 2 2,316. Within these images, an average number per image of 7 cars was detected in aerial images, and 4 cars in street view images.

5.1 Results

To evaluate the prediction result of our proposed method, we have to generate a manually labeled data set to test our results due to the lack of similar methods in current research. Our validation set contains labeled data of two suburbs of Leipzig, whereas one item contains the original street segment id, the segmentation number, iteration number, the street side and the parking value. The output of our method contains the same information so that we can compare result items one on one. Street segments where the annotators could not determine the parking type (for example, due to temporary roadworks or incorrectly parked cars) were ignored in the overall ranking.

Looking at the combined results from both image types for the unit *segments*, our method produces a good result with a rate for correctly classified parking types of 85.8% (see table 1). Between results with and without *no-detection area*, there is practically no difference. The results of the individual image sources separately are within the range of the overall result, with a slightly higher correct rate for aerial images and a slightly lower correct rate for street view images. Here also, the use of the no-detection area has no influence on the quality of the results.

By dividing a street segment into smaller units (*iterations*), it was our aim to recognize and localize different parking types within a segment allowing for the assignment of parking types to smaller, more nuanced units and thus improving the correct classification rate. Comparing the segment results to the unit of iterations, we can observe a higher correct classification rate for both data sources combined as well as for the individual data sources. The correct classification result for the combined data sources increases by approximately 1.7 %, the result for street view images increases by about 9.5 % and thus interestingly exceeds the combined result. As assumed, the smaller unit increases the overall result and solves the problem of different parking types within a segment. The usage of the no-detection area has a more significant impact for street view images. One explanation for this is the different number of cars recognised per image. For street view images, usually only the cars close to the camera are recognized, while the neural network does not recognize the vehicles further away. Since all cars in aerial images have the same distance from the camera, more (i.e. a higher number) cars are recognized. The fewer cars per image are included in the calculation, the more influence false detections such as moving or incorrectly parked cars have, which is why the no-detection area in street view images has more effect (see evaluation set specs at beginning of chapter 5).

Looking at the iteration results of the second district, we can observe similar numbers for the individual and combined image sources and similar behaviour for the use of the no-detection area. Interestingly, district 1 has more ties than district 2. This can be confirmed by the manual inspection of the data, which showed that in district 1 the parking type often changed from one type to another between the different recording years.

By merging the results by majority vote, individual false detections in a set of otherwise correct results do not lead to an incorrect overall result. The longer a segment, the more individual results are included in the majority decision and the fewer incorrect results distort the overall result. However, assigning only one result per segment overlooks the possibility of multiple parking types coexisting on a single segment and therefore distorts the result again. To address this, segments were subdivided into iterations, where an iteration length of 30 meters has proved to be a good choice.

5.2 Problems

By looking at faulty results, we were able to identify several factors leading to wrong parking type allocations which can be divided into context-related and methodological. Most methodological errors occur in crossroad areas due to the street division in the city data. In the data, the subdivision of streets is made in the midpoint of every intersection. This is why a part of the intersection is attached to each segment end (see Fig. 3). For street view in particular, because of this images are included in which the car is only just turning into the destination street, meaning that areas and cars that are not relevant are depicted. Secondly, there are cars in every branch of the intersection, which also leads to false detections by the neural network. It would be good to cut off these intersection areas from the segments before applying the methods. However, this task is not trivial, as the intersection areas are not always the same size and their size is unknown to the city. A small role plays the date of the recording. For a few streets, the parking type has changed over time. In this work, we evaluated two data sources with different recording years. If the parking method has changed, the method may produce a different but correct result for each image source, but the merging process produces the error. Another methodological error is produced by the different points of view in the data sources. Air images are cut out with exactly the boundaries of one segment or iteration, whereas street view images are collected by checking if their recording position lies within the segment/iteration. Street view images therefore depict a field of view several metres ahead of the current location and thus depict cars outside the currently inspected area. We would categorise the problem of differentiating between parking and moving cars between methodological and context-related errors since it is difficult to detect a dynamic state in stationary data. A mainly context-related error is the correct detection of cars in places with no parking and thus a wrong allocation of parking space (e.g. short-stay parking, parking for certain vehicles only such as taxis or e-cars, delivery vehicles, or cars in no-parking areas). For some of the segments or iterations, there is no valid image for either air or street view images, excluding them from the results.

6. DISCUSSION AND FUTURE WORK

Due to a lack of similar methods in current research, we proposed a novel method to identify the parking type in streets including the use of a mobile data source. This information is required within the city administration for a variety of urban planning processes: The parking pressure can be calculated by

unit	district	correct in %	ties in %	wrong in %	source	no-detection area
segment	district 1	**85.80**	3.55	10.65	combined	yes
segment	district 1	82.21	-	17.79	street view	yes
segment	district 1	88.76	-	11.24	air	yes
segment	district 1	85.50	3.25	11.24	combined	no
segment	district 1	82.52	-	17.46	street view	no
segment	district 1	88.76	-	11.24	air	no
iteration	district 1	**87.55**	3.09	9.36	combined	yes
iteration	district 1	**93.11**	-	6.89	street view	yes
iteration	district 1	89.99	-	10.01	air	yes
iteration	district 1	86.09	3.82	10.09	combined	no
iteration	district 1	89.44	-	10.56	street view	no
iteration	district 1	89.83	-	10.17	air	no
iteration	district 2	86.62	0.47	12.90	combined	yes
iteration	district 2	**88.07**	-	11.93	street view	yes
iteration	district 2	85.57	-	14.43	air	yes
iteration	district 2	82.73	0.76	16.51	combined	no
iteration	district 2	86.85	-	13.15	street view	no
iteration	district 2	85.87	-	14.13	air	no

Table 1. Results for different length units, data sources and no-detection area (grey)

intersecting the number of parking spaces with the number of car registrations in the same area. Based on this, parking concepts can be created for residential areas. Other use cases are the calculation of fairness of space allocation according to the different traffic participants or the planning of street greenery taking into account the parking situation. In comparison to research in this area, we do not use a locally permanently installed camera as data source, but movable data sources that move in or above the city. This gives us the opportunity to scale the application of our methods to the entire city and minimise privacy risks. Our method is intended to create an initial data basis, as up to now new recordings are made annually or biennially. It is foreseeable that the frequency of street view recordings will increase as soon as the recordings are no longer requested and purchased by the city authorities, but are automatically carried out by vehicles that are already on the road, such as buses or self-driving cars. We would like to encourage the usage of mobile data sources, because with increased recording frequency, detecting and transmitting free parking spaces, which in current research is limited by fixed cameras in parking lots, can also be implemented for street space. Moving data sources (cameras) make the installation of fixed cameras obsolete and allow the use case to scale to the whole city. Further, they represent a smaller privacy risk, as they do not permanently monitor individual locations. Prospectively this use case can be implemented with LiDAR data, a sensor that is also installed in self-driving cars, to even further reduce privacy concerns. For future work and as soon as recordings are available at shorter intervals, analyses can be extended to the time duration of parked cars. Thus, temporal parking patterns can be recognized and incorporated into the urban planning processes mentioned above. Furthermore, we plan to investigate multilane roads and include prior knowledge and rules on the range of validity of street signs. Moreover, we are already working on recognizing driveways in the images to include them in the calculation of the parking space estimation.

ACKNOWLEDGEMENTS

This paper was created as part of the ”Connected urban twins” project[4] funded by the German Federal Ministry of the Interior, Building and Community. The source of the street view images is *Cyclomedia*, the source of the aerial images is the City of Leipzig.

REFERENCES

Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., Vairo, C., 2017. Deep learning for decentralized parking lot occupancy detection. *Expert Systems with Applications*, 72, 327–334. https://www.sciencedirect.com/science/article/pii/S095741741630598X.

Bezirksamt Berlin-Neukölln, 2023. Parkraumbewirtschaftung. `https://www.berlin.de/ba-neukoelln/politik-und-verwaltung/aemter/ordnungsamt/parkraumbewirtschaftung/`. Accessed: 23.01.2024.

Biswas, S., Chandra, S., Ghosh, I., 2017. Effects of On-Street Parking in Urban Context: A Critical Review. *Transportation in Developing Economies*, 3(1), 10. http://link.springer.com/10.1007/s40890-017-0040-2.

Chen, Z., Xu, H., Zhao, J., Liu, H., 2023. Curbside Parking Monitoring With Roadside LiDAR. *Transportation Research Record: Journal of the Transportation Research Board*, 2677(10), 824–838. http://journals.sagepub.com/doi/10.1177/03611981231193410.

Fahim, A., Hasan, M., Chowdhury, M. A., 2021. Smart parking systems: comprehensive review based on various aspects. *Heliyon*, 7(5), e07050. https://linkinghub.elsevier.com/retrieve/pii/S2405844021011531.

GeoSN Saxony, 2024. `https://www.geodaten.sachsen.de/luftbild-produkte-3995.html?_cp=%7B%22accordion-content-4027%22%3A%7B%220%22%`

[4] `https://www.connectedurbantwins.de/en/`

3Atrue%7D%2C%22previousOpen%22%3A%7B%22group%22%3A%22accordion-content-4027%22%2C%22idx%22%3A0%7D%7D. Accessed last: 23.01.2024.

Goren, S., Oncevarlk, D. F., Yldz, K. D., Hakyemez, T. Z., 2019. On-Street Parking Spot Detection for Smart Cities. *2019 IEEE International Smart Cities Conference (ISC2)*, IEEE, Casablanca, Morocco, 292–295.

Lutz, C., 2014. *Cars and Transport: The Car-Made City*. John Wiley Sons, Ltd, chapter 8, 142–153.

Masmoudi, I., Wali, A., Alimi, A. M., 2014. Vision based System for Vacant Parking Lot Detection: VPLD:. *Proceedings of the 9th International Conference on Computer Vision Theory and Applications*, SCITEPRESS - Science and and Technology Publications, Lisbon, Portugal, 526–533.

Parmar, J., Das, P., M. Dave, S., 2020. Study on demand and characteristics of parking system in urban areas: A review, booktitle = journal of traffic and transportation engineering (english edition). 111–124.

Peng, X., Song, R., Cao, Q., Li, Y., Cui, D., Jia, X., Lin, Z., Huang, G.-B., 2022. Real-Time Illegal Parking Detection Algorithm in Urban Environments. *IEEE Transactions on Intelligent Transportation Systems*, 23(11), 20572–20587. https://ieeexplore.ieee.org/document/9797236/.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., Girshick, R., 2019. Detectron2. https://github.com/facebookresearch/detectron2.

Zhao, T., Nevatia, R., 2003. Car detection in low resolution aerial images. *Image and Vision Computing*, 21(8), 693–703. https://linkinghub.elsevier.com/retrieve/pii/S0262885603000647.