Development of a system to analyze the utilization of a bicycle parking space.

Part 2

T3100 / 5. Semester

of the Electrical Engineering program at the Dualen Hochschule Baden-Württemberg Stuttgart

Juan Sebastian Mejia Arenas

Delivery date: 12.05.24

Processing period:	10 Weeks
Matriculation number/ course:	9654407/ TEL21GR3
Training company:	Coperion GmbH. Stuttgart
Supervisor of the project:	M. Sc. Jerg Pfeil

Declaration

in accordance with§ 5 (3) f the "Study and Examination Regulations DHBW

Technik" dated 12.

May 2024.

I have written this thesis independently and have not used any sources or aids other than those sources and aids other than those specified.

Stuttgart. 12.05.2024	Sebastian Mejía
Place. date	Signature

Abstract

This project focuses on the development of a parking lot monitoring system utilizing object detection techniques. The project delves into the meticulous selection of object detection models and the optimization of parameters to achieve accurate and efficient detection of bicycles and motorcycles in a parking lot. Following the thorough evaluation of models, the YOLOv5n6 model is chosen for its suitability to the application.

The implementation of the system on a Raspberry Pi platform using Python is detailed, emphasizing the stages of the detection pipeline. These stages include real-time image capture, image preprocessing, object detection using the selected model, tracking of instances across frames, and data analysis for monitoring parking lot occupancy trends.

The culmination of the project is the initial deployment of the parking lot monitoring system on the Raspberry Pi platform, showcasing its functionality and potential for further development. Through meticulous model selection, parameter optimization, and detailed implementation, the project offers a solution for parking lot management by harnessing the power of computer vision technology.

Key words: Object Detection. Convolutional Neural Networks. YOLO. Raspberry Pi. Parking Lot Analysis. Computer Vision.

Kurzfassung

Dieses Projekt konzentriert sich auf die Entwicklung eines Parkplatzüberwachungssystems unter Verwendung von Objekterkennungstechniken. Es befasst sich mit der sorgfältigen Auswahl von Objekterkennungsmodellen und der Optimierung von Parametern, um eine genaue und effiziente Erkennung von Fahrrädern und Motorrädern auf einem Parkplatz zu ermöglichen. Das YOLOv5n6-Modell wurde nach einer gründlichen Bewertung aufgrund seiner Eignung für die Anwendung ausgewählt.

Die Implementierung des Systems auf einer Raspberry Pi-Plattform unter Verwendung von Python wird detailliert beschrieben. Die Phasen der Systems Pipeline, einschließlich Echtzeit-Bildaufnahme, Bildervorbearbeitung, Objekterkennung, Tracking und Datenanalyse zur Überwachung der Parkplatzbelegung, werden hervorgehoben.

Der Höhepunkt des Projekts ist die erstmalige Bereitstellung des Parkplatzüberwachungssystems auf der Raspberry Pi-Plattform. Es zeigt die Funktionalität und das Potenzial für weitere Entwicklungen auf. Durch sorgfältige Auswahl von Modellen, Optimierung von Parametern und detaillierte Implementierung bietet das Projekt eine Lösung für das Parkplatzmanagement durch die Nutzung der Leistungsfähigkeit der Computervisionstechnologie.

Schlüsselwörter: Objekterkennung. Faltungen neuronale Netzwerke. YOLO. Raspberry Pi. Parkplatzanalyse. Computer Vision.

Table of contents

II. List of tables

III. List of figures

1. Introduction

In recent years, the field of object detection has witnessed significant advancements, particularly in the realm of real-world applications such as surveillance, autonomous driving, and traffic monitoring. The ability to accurately detect and track objects in complex environments has become crucial for enhancing safety, efficiency, and decision-making processes.

This project aims to leverage state-of-the-art object detection techniques to analyze the usage of the parking lot at the dual Hochschule Baden Wurttemberg in Stuttgart, specifically focusing on the identification of bicycles and motorcycles. The objective is to develop a system capable of monitoring parking lot occupancy in real-time, providing valuable insights into usage patterns over time.

In the initial phase of the project, an exhaustive model selection process was conducted, culminating in the identification of four candidate models for training and evaluation in real-world scenarios. A custom dataset, meticulously curated to align with the specific requirements of the application, was prepared for the training phase. Subsequently, the selected models underwent rigorous training on this dataset, with their performance assessed based on results derived from the training and validation set. Qualitative evaluation of the models' accuracy, along with an analysis of implementation time per frame on Raspberry Pi, was undertaken to ascertain their suitability for the intended application.

Building upon the groundwork laid in the first phase, the second part of the project is dedicated to refining the parameters of the object detection task to ensure optimal performance of the models. To accomplish this, a comprehensive evaluation using a dataset consisting solely of labeled photos from the university parking lot is conducted. These photos represent the real-world application scenario, providing a thorough assessment of the models’ performance and the selection of appropriate thresholds for the intended environment.

In parallel with the evaluation of model performance, the second phase also involves the development and implementation of the detection system's codebase. This

process entails crafting the entire codebase from scratch, encompassing modules for real-time image capture, preprocessing, object detection, tracking and data analysis. Furthermore. the development of an user interface for visualizing and analyzing the collected data will provide insights into parking lot usage trends.

Following the evaluation and selection of the best-performing model, the initial deployment will commence. This will involve integrating the chosen model into the detection system's codebase and deploying it in the university parking lot environment. Real-time monitoring of parking lot occupancy is then achieved, with the system providing valuable insights.

2. Literature Review

2.1. Non-Maximum Suppression

Non-maximum suppression (NMS) is a pivotal technique widely employed in object detection algorithms[1]. The fundamental principle of NMS lies in its ability to efficiently select the most confident detections while discarding redundant or overlapping bounding boxes. This process is crucial for enhancing detection accuracy by eliminating false positives and ensuring that each object is represented by a single bounding box [2].

NMS operates as a post-processing step following the initial detection phase. By iteratively comparing overlapping bounding boxes and suppressing those with lower confidence scores, NMS effectively filters out redundant detections.[3]

One common method of NMS involves setting a fixed threshold, known as the Intersection over Union (IoU) threshold, to determine the extent of overlap allowed between bounding boxes. Specifically, during NMS, the algorithm retains the bounding box with the highest confidence score among overlapping boxes, while discarding the others.[4] YOLO model incorporates NMS as part of its object detection process and allow users to manually set the IoU threshold parameter to control the degree of overlap tolerated before suppression occurs[5].

In practical applications, NMS serves as a critical post-processing step to refine detection results and reduce false positives. By removing lower confidence detections that overlap excessively with higher confidence detections, NMS ensures that only the most reliable detections are retained[6]. In the project, the NMS technique from YOLO will be utilized to optimize detection results.

---

[1] F. Liang, S. Yang, T. Mai and Y. Yang (2018)
[2] M. Shi, P. Ouyang, S. Yin, L. Liu and S. Wei (2019)
[3] L. M. Brown, R. Feris and S. Pankanti (2014)
[4] X. Zhang, W. Su, J. Li, J. Li and X. Lou (2021)
[5] Ultralytics. (2024).
[6] S. B. Choi, S. -S. Lee, J. Park and S. -J. Jang (2021)

2.2. Tracking in Object Detection

Tracking is a fundamental aspect of object detection systems, serving to maintain temporal coherence and understand object dynamics over time. Tracking algorithms primarily focus on associating detections between consecutive frames, enabling tasks ranging from simple identification of object appearances and disappearances to more complex ones like predicting spatial trajectories and recognizing activities.[7]

Additionally, tracking enhances the robustness of object detection by providing temporal consistency and contextual information. Through the establishment of correspondences between detections, tracking algorithms overcome challenges such as occlusions, appearance changes, and intermittent detections.[8]

One approach for tracking tasks is the multi-object tracking with object bounding box association also known as overlap-based tracking. This method relies on measuring the spatial overlap between bounding boxes from consecutive frames, typically quantified using metrics like Intersection over Union (IoU). Higher IoU values indicate greater spatial correspondence between objects.[9]

By leveraging the IoU metric and overlap-based tracking, object detection systems can effectively track object appearances and disappearances over time, facilitating tasks such as occupancy monitoring. This approach offers a computationally efficient and robust method for temporal event tracking without the need for complex trajectory modeling or prediction algorithms.[10]

In this project, a modified adaptation of the overlap-based tracking method, is utilized. This approach is further explained and detailed in Section 4.4 Tracking, where the specific implementation and modifications made to suit the project's requirements are outlined.

---

[7] A. S. B. Sadkhan, S. R. Talebiyan and N. Farzaneh (2021).
[8] Klingler, N. (2024, April 18)
[9] N. Yang, Y. Wang and L. -P. Chau (2021)
[10] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2021)

3. Model Evaluation and Parameter Optimization

This section delves into the process of evaluating the performance of object detection models and optimizing key parameters to achieve superior results. The methodology encompasses analysis of confidence thresholds, IoU thresholds (for non-maximum suppression), and model selection to ensure the effectiveness of the detection system.

3.1. Dataset Presentation

The dataset used for model evaluation and parameter optimization comprises a total of 491 images captured from the parking lot at the dual Hochschule Baden Wurttemberg in Stuttgart. The composition of the test dataset is summarized in Table 3.1.

Table 3.1.: Test-dataset Composition

Feature	Details
Total Images	491
Total Bicycles	2549
Total Motorcycles	2
Images with Snow	250 (51%)
Images without Snow	241(49%)
Average Instances per Image	Aprox. 5
Total Parking spots	144
Occupancy rate	3.5%

The set encompasses a diverse range of environmental conditions, including varying light conditions and shadow effects. All photos are taken during daytime hours, within the university's operational hours, to faithfully represent typical parking lot conditions. Furthermore, photos featuring snow are included to ensure the model's robustness across different seasons.

It's worth noting that the low occupancy rate of 3.5% reflects typical parking lot conditions observe during the university's operational hours. Future studies may include footage from peak summer periods when higher occupancy rates are expected, providing further insights into parking lot dynamics. Additionally. the low frequency of motorcycles (only 2 instances) makes it not suitable to perform an

analysis of the threshold for this class. Therefore, the analysis will primarily focus on optimizing the detection of bicycles, which are more prevalent in the parking lot.

Each image in the dataset is meticulously labeled to delineate the presence and location of instances, providing ground truth annotations for subsequent evaluation. The inclusion of images with diverse environmental conditions ensures robust evaluation of the object detection models, enabling thorough assessment across different lighting and shadow scenarios. This diversity enhances the generalizability of the models, ensuring their effectiveness in real-world deployment scenarios.

For visual reference, a selection of representative photos from the dataset can be found in the Appendix section.

3.2. Confidence Threshold Analysis

The labeled dataset prepared in the previous section served as the foundation for model evaluation and parameter optimization. The focus lies on optimizing two crucial parameters of the detection task: the confidence threshold and the IoU threshold for non-maximum suppression (NMS).

Initially, for the confidence threshold analysis and optimization, the IoU parameter remains at its default value of 0.7(default by YOLO), while systematically varying the confidence threshold. The models perform detections on the complete dataset, these detections are evaluated to obtain a list of true positives (TP), false positives (FP), and false negatives (FN) for each image. These metrics are obtained by comparing the detection results with the ground truth annotations.

The TP, FP, and FN values are then aggregated to calculate the precision, recall, and F1-score for the entire dataset. This process is repeated with different confidence thresholds ranging from 0.01 to 0.775. The objective is to construct a precision-recall (PR) curve to visualize the model's behavior across different confidence thresholds and identify the appropriate one for the application. The results of this analysis are presented in Table 3.2.

Table 3.2.: Performance Metrics at Different Confidence Thresholds

Confidence threshold	TP	FP	FN	Precision [%]	Recall [%]	F1-Score [%]
0.01	2133	1203	365	63.9	85.4	73.1
0.025	2078	681	431	75.3	82.8	78.9
0.05	2028	449	488	81.9	80.6	81.2
0.075	1993	332	526	85.7	79.1	82.3
0.1	1954	249	569	88.7	77.4	82.7
0.125	1934	200	592	90.6	76.6	83.0
0.15	1914	161	616	92.2	75.7	83.1
0.175	1899	123	632	93.9	75.0	83.4
0.2	1872	101	662	94.9	73.9	83.1
0.225	1858	79	676	95.9	73.3	83.1
0.25	1843	57	692	97.0	72.7	83.1
0.275	1815	47	721	97.5	71.6	82.5
0.3	1794	41	742	97.8	70.7	82.1
0.35	1747	18	796	99.0	68.7	81.1
0.4	1693	5	854	99.7	66.5	79.8
0.45	1619	0	930	100	63.5	77.7
0.5	1546	0	1003	100	60.7	75.5
0.55	1457	0	1092	100	57.2	72.7
0.6	1349	0	1200	100	52.9	69.2
0.65	1226	0	1323	100	48.1	65.0
0.7	1082	0	1467	100	42.4	59.6
0.75	884	0	1665	100	34.7	51.5

As observed in Table 3.2, the detection with the highest F1-score occurs with a confidence threshold of 0.175 (green highlighted). The decision regarding the threshold selection is based on the desired trade-off between precision and recall in the application.

Precision is crucial for accurately localizing and identifying instances in the parking lot. However, recall holds greater importance as it determines the ability to recognize the maximum number of instances, providing a comprehensive occupancy report. Considering that precision can also be improved through tracking and software

solutions, it is preferable to prioritize recall. Therefore, a confidence threshold of 0.05, which exhibits a slightly higher recall value while maintaining a high F1-score, is chosen (blue highlighted).

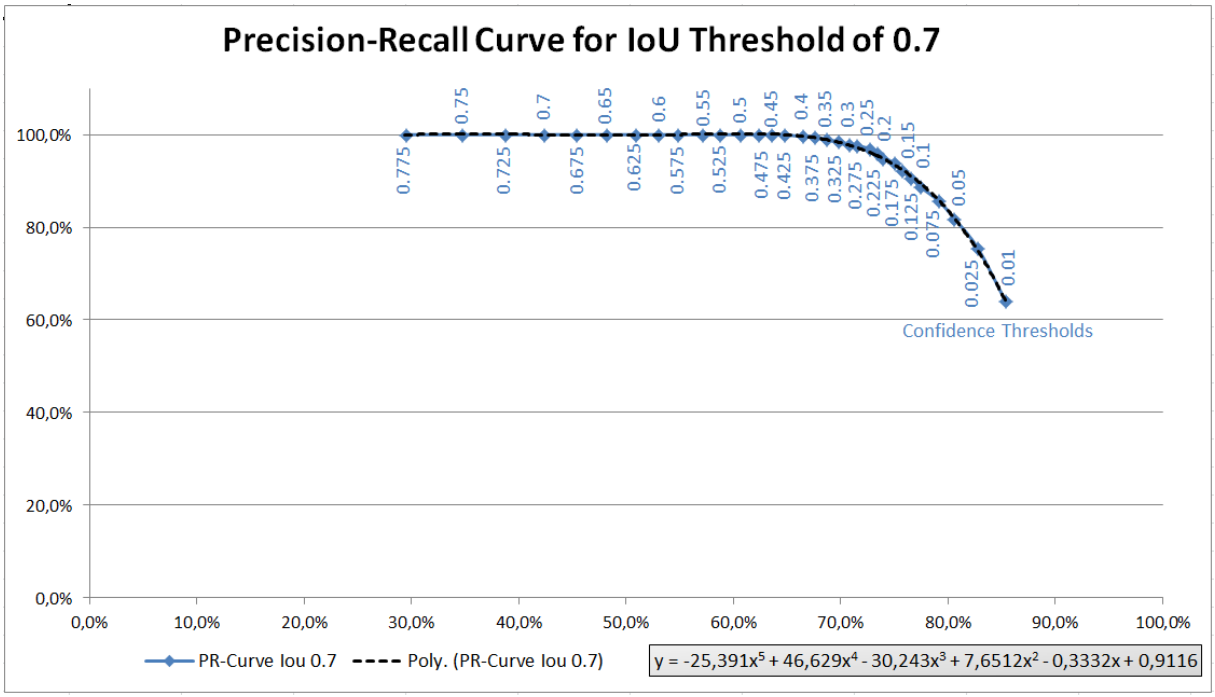

Figure 1.Precision-Recall Curve for IoU Threshold of 0.7

The Precision-Recall (PR) curve, depicted in Figure 1, follows the typical characteristics of a PR curve, with high precision and low recall at high confidence threshold values, and high recall but low precision at lower thresholds. This consistency in behavior instills confidence in the evaluation process.

Furthermore, the area under the curve represents the mean Average Precision (mAP) of the model. This value is calculated from the integral of the PR curve's polynomial trendline, providing a comprehensive measure of the model's performance across all confidence thresholds.

3.3. IoU-Threshold analysis for Non-Maximal Suppression

Having optimized the confidence threshold, the focus now shifts to the IoU threshold for non-maximal suppression (NMS). The objective remains the same: to find the most optimal value for our application.

The process begins by systematically reducing the IoU threshold and performing confidence threshold selection with the new NMS value. This iterative process continues with different IoU values, and the results are summarized in Table 3.3.

Table 3.3.: Performance Metrics at Different IoU Thresholds

IoU-Threshold	Best Confidence Threshold	TP	FP	FN	Precision [%]	Recall [%]	F1-Score [%]	mAP50 [%]
0.7	0.1	1954	249	569	88.7	77.4	82.7	53.88
0.5	0.05	2032	219	490	90.3	80.6	85.1	54.82
0.4	0.05	2033	136	490	93.7	80.6	86.7	55.13
0.3	0.05	2024	109	494	94.9	80.4	87	55.08
0.2	0.05	1967	119	535	94.3	78.6	85.7	54.15
0.1	0.05	1933	115	571	94.4	77.2	84.9	52.88

As observed in the table, the F1-Score and mAP increase as the IoU threshold decreases from 0.7 to 0.3. This is attributed to the reduction in overlapping boxes (duplicates), which were generating false positives. By reducing the IoU threshold, the number of false positives decreases, leading to an improvement in precision. The recall remains almost constant, indicating the elimination of just duplicate detections. However, below an IoU threshold of 0.3, the recall starts to decline due to the elimination of unique detections close to each other.

Based on the analysis, the optimal IoU threshold is determined to be 0.3. This threshold value yields the highest F1-score, mAP, precision, and recall among the values considered in the table.

3.4. Model Selection

The evaluation of the best performing confidence threshold and IoU threshold for NMS was conducted for all four models reviewed in the first part of the project. This quantitative analysis serves as a comprehensive assessment of each model's performance in the context of the project's objective: analyzing occupancy in a parking lot using object detection techniques. The results of this evaluation are summarized in table 3.4.

Table 3.4.: Best performing metrics of the Object Detection Models

Model	IoU-Threshold	Best Confidence Threshold	Precision [%]	Recall [%]	F1-Score [%]	mAP50 [%]
YOLOv5n6	0.3	0.05	94.9	80.4	87	55.08
YOLOv8n_Large	0.3	0.05	98.3	78.1	87.3	66.68
YOLOv5n	0.3	0.05	96.4	54.5	69.7	48.11
YOLOv8n	0.3	0.05	94.1	61.6	74.4	66.93

Notably, the YOLOv5n6 model emerges as the most promising candidate for the application at hand, boasting the highest recall rate among all models. While the YOLOv8n_Large model also demonstrates a commendable performance, with a higher mAP50 value, its inference speed is slower compared to YOLOv5n6.

After analyzing the models with parameter optimization, it's appreciable that YOLOv8 performs better overall than YOLOv5. However, for the specific application, what matters most is how well the model performs at a particular threshold, especially in terms of recall. YOLOv5n6 stood out in this regard, showing superior recall and lower interference speeds. Consequently, YOLOv5n6 was chosen as it aligns more closely with the project's requirements.

However, it's essential to acknowledge the broader benefits of YOLOv8's generalization and overall performance. YOLOv8's higher mean Average Precision (mAP50) signifies its superior ability to balance precision and recall across various thresholds, making it advantageous in scenarios where overall detection quality is critical. The advancement in both YOLOv5 and YOLOv8 versions underscores the continuous improvement in object detection models, offering options tailored to specific application needs.

3.5. Error Analysis and Practical Implications

The analysis of the selected model performance reveals insights into the associated errors, primarily measured through precision and recall metrics.

With a precision of 94.9%, the error associated with precision is relatively low, calculated as 5.1%. This implies that the instances detected by the system are highly certain to be true positive detections. Furthermore, these errors in precision can be mitigated through software validation mechanisms, where detections are validated over time, further reducing this factor. Therefore, the reported occupancy status for detected spots can be considered reliable and accurate.

The recall metric quantifies the system's ability to detect all instances of occupancy within the parking lot. With a recall of 80.4%, the error associated with recall is 19.6%. This suggests that from every five instances detected by the system, one instance could have been overlooked or not detected. While this error rate indicates a slight deficiency in the system's ability to capture all instances of occupancy, it remains within acceptable bounds considering the overall accuracy achieved.

It's important to interpret this error rate within the specific context of the application. In a parking lot with relatively low usage and ample open parking spots, the error associated with undetected instances is not critical.

Moreover, the system's performance allows for a comprehensive analysis of occupancy trends over time, providing valuable insights into parking lot usage patterns. While the error may be slightly more critical for users seeking real-time occupancy information or searching for parking spots, the abundance of available parking spots mitigates its impact on practical use.

Additionally, the potential for improving detection accuracy through preprocessing mechanisms, such as enhancing brightness in low-light areas, offers opportunities to further refine the system's performance. By addressing specific challenges identified during evaluation, such as instances not detected under low-light conditions, the system can enhance its ability to accurately capture occupancy data in various lighting conditions.

Overall, the error analysis provides valuable insights into the system's performance and its practical implications for occupancy analysis in parking lots. Despite minor

deficiencies, the system remains a useful tool for monitoring and managing parking lot occupancy, supporting efficient utilization of parking resources.

4. Codebase Development

Following the thorough evaluation of object detection models and the selection of optimal parameters for the detection task, the next phase of the project involves the implementation of the chosen model into a detection system. This section focuses on the development of the codebase required to analyze parking lot occupancy using object detection techniques.

The codebase is designed to facilitate real-time analysis of parking lot images captured at regular intervals. This analysis is achieved through a systematic pipeline that encompasses various stages, including image capture, preprocessing, object detection, tracking and data analysis. The flowchart depicting this pipeline is presented in Figure 2.

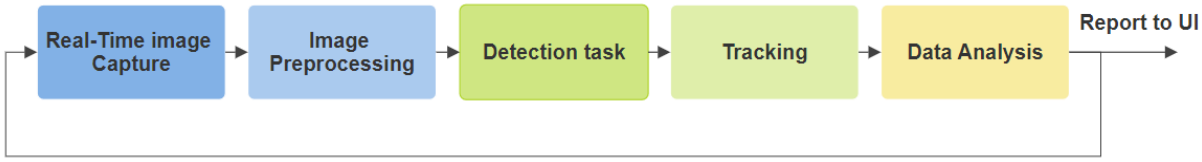

Figure 2. Parking Lot Analysis Pipeline

The codebase is capable of efficiently processing parking lot images and extracting valuable insights regarding occupancy patterns, providing accurate and timely information for monitoring parking lot usage.

The code will be executed on a Raspberry Pi 4, utilizing Python for implementation, ensuring compatibility and ease of deployment in the target environment.

4.1. Real-Time Image Capture

The real-time image capture step serves as the initial stage in the detection system pipeline. This process involves capturing images of the parking lot environment using a Raspberry Pi Camera Module 3.

The purpose of this step is to gather visual data of the parking lot at regular intervals, which serves as the input for subsequent stages in the pipeline. An example of a captured image of the parking lot environment is shown in Figure 3.

Figure 3. Example of captured image of the parking lot environment using the Raspberry pi Camera Module 3.

4.2. Preprocessing

Preprocessing plays a vital role in enhancing the quality of images before they undergo object detection. In this stage, the captured images are manipulated to isolate the region of interest (ROI) and apply a mask to focus solely on the desired areas for detection. This process is essential for improving the accuracy and efficiency of the detection system.

The application of a mask helps exclude unnecessary areas from the detection process, such as background elements or irrelevant objects, allowing the model to concentrate on the parking lot environment specifically. By cropping the image to the ROI and applying a mask, the detection system can effectively reduce computational overhead and enhance the precision of object detection. Additionally, the preprocessed image is properly resized to meet the requirements of the used model, ensuring the proper functioning of the detection task. An example of the preprocessed image is shown in Figure 4.

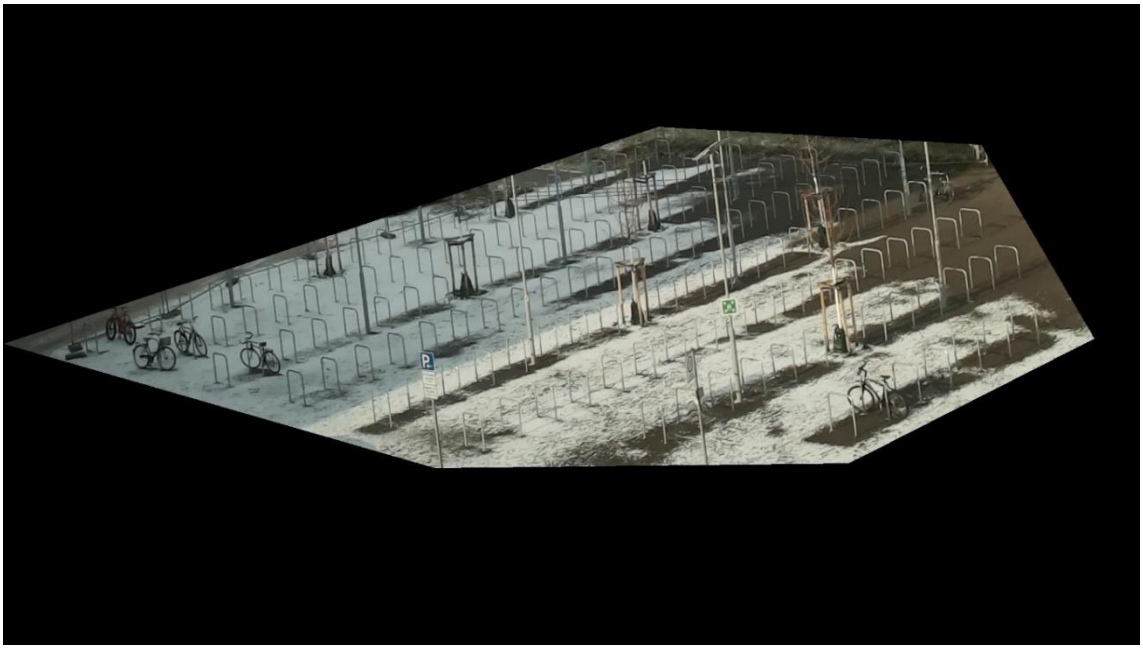

Figure 4. Example of Preprocessed image showing ROI with applied mask

Furthermore, while the current preprocessing techniques focus on isolating the ROI, additional preprocessing methods could be beneficial for further improving the detection task. Techniques aimed at enhancing image brightness, could significantly enhance the model's performance.

Throughout the day, the parking lot experiences different lighting conditions, including partial shadow, complete shadow, and full sunlight. While images captured under direct sunlight are well-detected, the model encounter challenges with images affected by shadows. Dynamic brightness adjustment techniques could address these challenges by adaptively altering the image's brightness based on the prevailing lighting conditions.

Although not implemented in the current project due to time constraints, the incorporation of dynamic brightness adjustment techniques is highly recommended for improving accuracy and ensuring optimal performance across different times of the year. By addressing variations in lighting conditions, these techniques can enhance the robustness and reliability of the detection system, providing accurate insights into parking lot occupancy.

4.3. Detection Task

In the detection task stage, the preprocessed images are passed through the trained object detection model to identify and localize objects of interest within the parking lot. This step involves utilizing the selected model, YOLOv5n6, which has been trained on a custom dataset to detect bicycles and motorcycles. The detection task is performed with the optimized parameters including an IoU threshold of 0.3 for non-maximum suppression (NMS) and a confidence threshold of 0.05.

Utilizing the trained model, the detection task analyzes each preprocessed image to identify objects of interest. The model evaluates regions of the image to generate bounding boxes around detected objects, assigning class identifiers and confidence scores to each detection. The optimized parameters ensure the detection system's effectiveness in minimizing false positives and false negatives while maximizing detection accuracy.

Upon completion of the detection task, the preprocessed images are promptly deleted to maintain anonymity and protect privacy. Only the information pertaining to the detected instances, such as class identifiers, bounding box coordinates, and associated scores, is retained. This ensures compliance with privacy regulations and safeguards the anonymity of individuals present within the parking lot.

These results serve as the basis for further analysis and tracking of objects within the parking lot.

4.4. Tracking

In this phase of the pipeline, the focus shifts to maintaining temporal coherence by effectively tracking object instances across consecutive frames. The systematic tracking process ensures accurate association of detections with specific parking spots and evaluates the consistency of the detections over the time, thereby improving the overall accuracy and reliability of the detection system. The tracking process plays a pivotal role in evaluating object presence and facilitating subsequent data analysis of occupancy.

The process begins with the detection list generated by the detection task for the current frame, follows by a Parking Spot Assignment, a Consistency Check, and the Determination of Occupancy. The tracking process is visually represented in Figure 5.

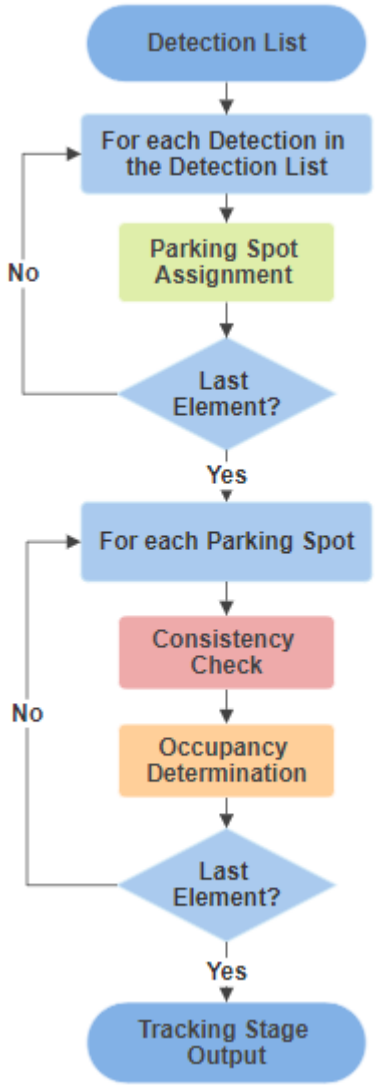

Figure 5. Tracking Process Flowchart.

4.4.1. Parking Spot Assignment

In the parking spot assignment phase, each detection from the current frame undergoes comparison with the coordinates of manually mapped parking spots within the parking lot environment. The mapped parking lot, depicted in Figure 7, showcases each parking spot encapsulated within green bounding boxes. Each green bounding box represents the spatial coordinates of a parking spot, which are uniquely identified using an alphanumeric system (e.g., A1, A2, B1, etc.).

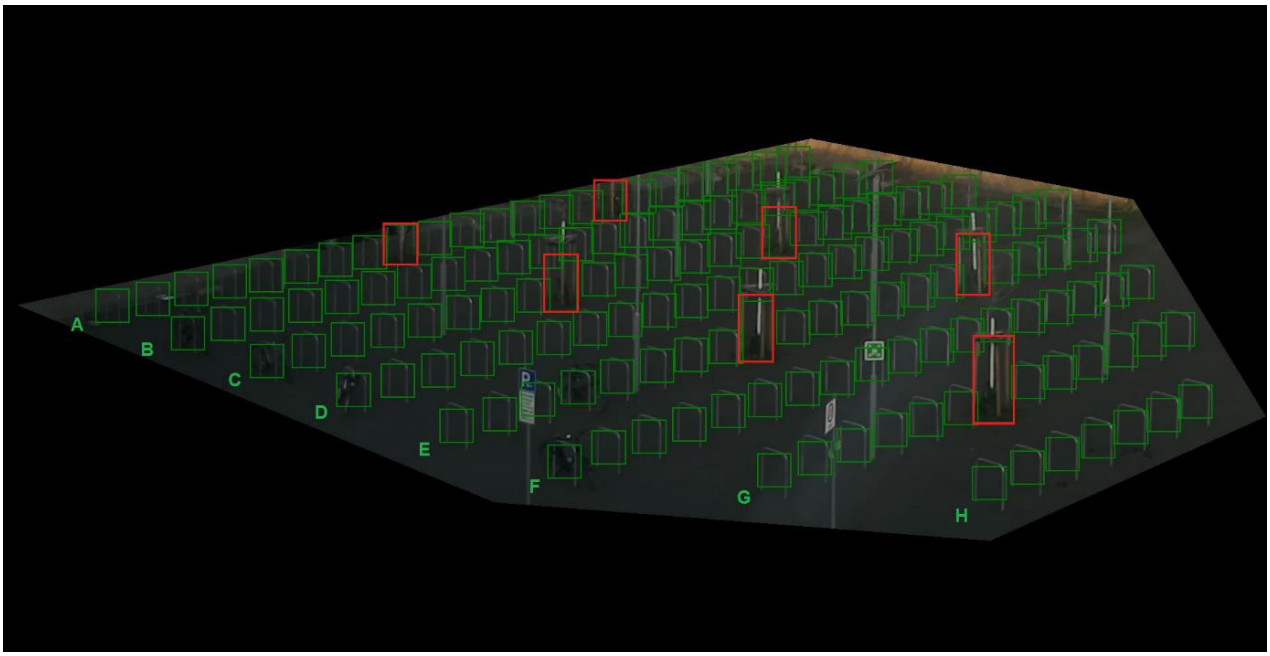

Figure 6. Mapped Parking Lot with Bounding Boxes.

The parking spot assignment process is visually outlined in the flowchart depicted in Figure 8 and consist of the following central steps:

- **Comparison of Bounding Boxes:** *Each* bounding box detected in the current frame is compared with *every* parking spot bounding box.[11]
- **IoU Calculation:** The comparisons are performed using the Intersection over Union metric. The bounding box pair with the highest IoU among the comparisons is selected and compared with a threshold.[12]
- **Threshold Evaluation:** The threshold value helps to determine whether a detection corresponds to a parking spot. Bounding boxes with IoU values above the threshold are considered matches completing the parking spot assignment process[13]. Detections that cannot be matched with a parking spot are disregarded as potential error detections, a point that warrants further analysis in future improvements to the program.
- **Detection Record:** Once a detection is successfully matched with a parking spot, it is added to the detection record of the respective spot. This detection record is subsequently used to track instances across multiple frames, enabling

[11] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2021).
[12] N. Yang, Y. Wang and L. -P. Chau (2021)
[13] N. Yang, Y. Wang and L. -P. Chau (2021)

the system to monitor the occupancy status of individual parking spots over time. The detection record has a maximal length of "***n***", constituting the level of consistency explained in the next section.

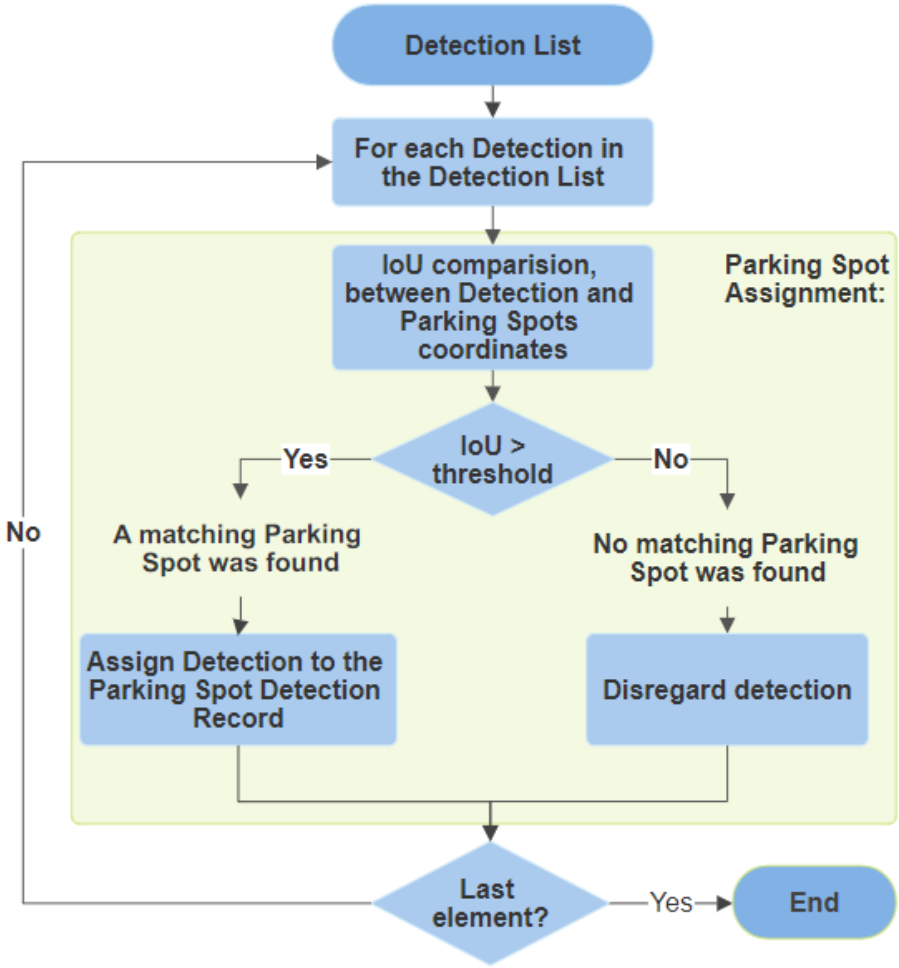

Figure 7. Parking Spot Assignment Flowchart

It is important to note that red bounding boxes within the mapped parking lot (Figure 6) indicate areas where obstacles are present. Within these areas, no bicycles or motorcycles should be detected. To mitigate erroneous detections within these regions, detections assigned to the red bounding boxes are disregarded.

This systematic approach not only enables accurate association of detections with specific parking spots but also ensures the mitigation of false positives within designated areas of the parking lot.

4.4.2. Consistency Check and Occupancy Determination

Once the parking spot assignment is completed, the tracking process proceeds by incorporating temporal consistency measures that enhance object detection performance. Temporal consistency is evaluated by checking detections across multiple frames to address challenges inherent in the detection task. For instance, objects that disappear in one frame but reappear in subsequent frames may be tracked across short temporal gaps. A visualization of the Consistency Check and Occupancy Determination process is depicted in Figure 8.

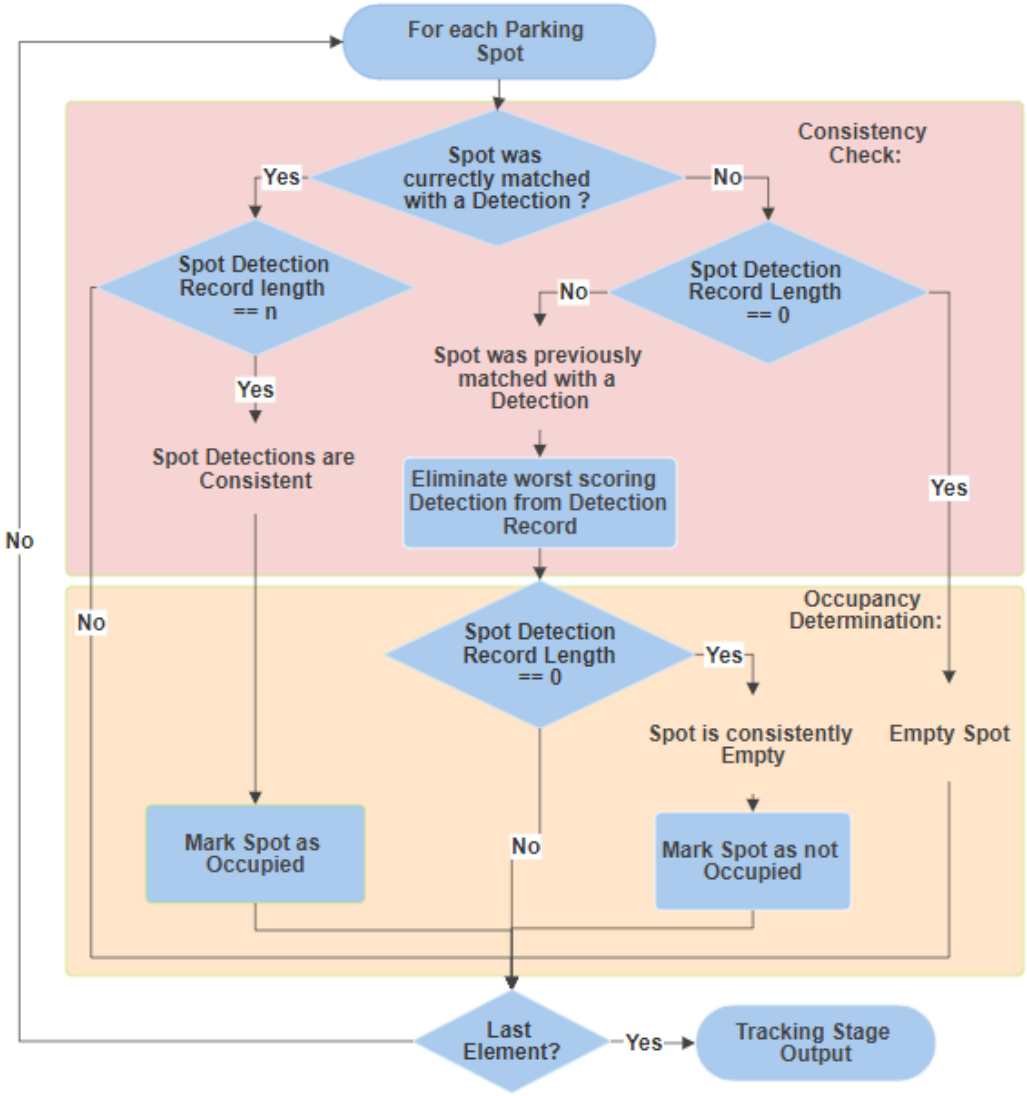

Figure 8. Consistency Check and Occupancy Determination Flowchart

To conduct the consistency check, the system iterates through each parking spot and examines its detection record. If the detection record reaches a predefined maximum length, denoted as "***n***", it signifies that the instance associated with that spot has been

consistently detected across multiple frames. This maximum length, or consistency level, ensures that only instances consistently detected over a certain duration are considered present in the parking spot, thereby marking it as occupied.

Conversely, to determine if an occupied parking spot becomes vacant, the system monitors the detection record. If a spot's detection record does not receive a new entry in the current frame, it suggests that no detection was found in that spot. In such cases, the system removes one entry from the detection record (if possible). By systematically erasing one detection from the record, the system checks for the consistent absence of the instance associated with that spot. If, over subsequent frames, no detections are matched with the spot, further entries are removed until the detection record reaches a length of 0. This indicates that the instance has been consistently absent, and the parking spot can be marked as unoccupied.

This process ensures precise occupancy determination, allowing the system to accurately identify both occupied and vacant parking spots. Once all occupied parking spots are identified, the system can analyze the data, providing valuable insights for parking lot management and utilization.

4.4.3. Determining Detection Record length “*n*” and Frame Capture Frecuency “*f*”

The length of the detection record plays a crucial role in shaping the robustness and accuracy of the tracking process. A longer detection record provides a larger dataset for performing consistency assessments, ultimately enhancing the system's accuracy, however, longer detection records also increase computational overhead. On the other hand, shorter detection records reduce the level of consistency checks but result in lower computational usage.

Another important aspect to consider is the frame capture frequency, which determines the time interval between frames in the Detection Record. This frequency also dictates how often the entire pipeline, from image capture to reporting, is repeated. Therefore, the length "***n***" and frequency "***f***" are closely related parameters that must be chosen judiciously to optimize system performance.

The selection of appropriate values involves careful consideration of several factors:

- **Spatial Trajectory Tracking**: Since this project does not require tracking spatial trajectories, extremely short frame capture frequencies (in the range of milliseconds or seconds) are deemed unnecessary.
- **Parking Lot Environment:** Instances in the parking lot typically exhibit stable occupancy patterns, most instances tend to persist for extended periods, often exceeding half an hour. To strike a balance between responsiveness and efficiency, a frame capture frequency of up to 15 minutes (***f*** < 15 min) is chosen. While a higher frequency, such as 30 minutes, could also capture occupancy changes effectively, opting for a 15-minute interval provides an additional margin of assurance, ensuring that no crucial details are overlooked.
- **Reporting Latency**: In this project, reporting latency refers to the time it takes for the system to consistently detect an instance across multiple frames before designating the parking spot as occupied and reporting it. This latency is determined by the length of the detection record, and the frame capture frequency. The reporting latency formula is expressed as:

  $$\boldsymbol{report\ latency} = \boldsymbol{f} * (\boldsymbol{n} - \mathbf{1}) \qquad \boldsymbol{for}\ \boldsymbol{n} > \mathbf{1}$$

  Essentially, this formula quantifies the time required to accumulate enough data across multiple frames to confidently report occupancy. To ensure minimal reporting latency and comprehensive coverage of the parking lot, the maximum reporting latency is set at 15 minutes.

Based on these considerations, a proper tracking system can be stablished. Is desired to have a "***f***" and "***n***" configuration that does not exceed the reporting Latency of 15 minutes. The possibilities for the parameter selection are shown in the table 4.1.

The goal is to achieve the highest frame capture frequency possible to minimize computational usage while ensuring timely occupancy reporting. With this objective in mind, a detection record length of 2 provides a reporting latency of 15 minutes, which is the maximum acceptable threshold. This configuration will be initially tested in the program.

Table 4.1: Detection Record length and Frame Capture Frequency configurations

Length of detection record ***n***	***f*** = Latency / (***n***-1)	Max frequency (Latency = 15 min)
2	***f*** = Latency / (1)	***f*** = 15 min
3	***f*** = Latency / (2)	***f*** = 7.5 min
4	***f*** = Latency / (3)	***f*** = 5 min
5	***f*** = Latency / (4)	***f*** = 3.75 min
6	***f*** = Latency / (5)	***f*** = 3 min
7	***f*** = Latency / (6)	***f*** = 2,5 min
8	***f*** = Latency / (7)	***f*** = 2,2 min

However, alternative configurations with longer detection record lengths, such as "n=3" or "n=4," also offer long frame capture frequencies, potentially enhancing the level of consistency and accuracy of detections. These alternative configurations present valid options for testing and optimization in future system improvements. By experimenting with different "*n*" values and assessing their impact on system performance, the tracking system can be further refined.

4.5. Data Analysis

In this phase of the project, a lightweight data analysis framework was implemented to showcase the functionality of the object detection model and provide basic insights into parking lot occupancy. While the primary focus of the project was on object detection, a simple database and local metrics display were integrated to demonstrate the potential for future development of a comprehensive parking lot management system.

4.5.1. Data Collection and Storage

The database consists of a structured schema to record key attributes related to parking spot occupancy. Each entry in the database represents an occupied parking spot found in a frame and report, enabling comprehensive occupancy analysis. The following attributes are stored for each entry:

- **Id**: A unique identifier for each entry in the database.
- **Spot id**: Identifier for the parking spot where occupancy is detected.
- **Report timestamp:** Timestamp indicating the time of the report when the occupancy was detected.

- **Occupied by**: Indicator of the type of vehicle occupying the spot. (0 for bicycles, 1 for motorcycles).
- **Occupied timestamp**: The timestamp indicating when the parking spot was first occupied.
- **Average occupation duration** Average duration of occupancy for the parking spot.
- **Confidence score**: Confidence score associated with the best detection on the detection record.

4.5.2. Occupancy Analysis

A set of key occupancy metrics were calculated to showcase the functionality and capabilities of the parking lot management system. These metrics offer insights into parking lot utilization and provide a foundation for future enhancements and developments.

- **Current Occupancy Count**: Provides real-time counts of bicycles, motorcycles, and total instances currently occupying the parking lot.
- **Live Occupancy Monitoring:** Displays a dynamic graphical representation of parking lot occupancy levels over the current day, offering insights into temporal patterns and usage trends.
- **Occupied Spots List**: Presents an inventory of parking spots currently occupied by instances, facilitating spot identification and navigation for users.

By calculating these metrics, the performance and functionality of the object detection model can be effectively showcased. While this data analysis represents a lightweight approach, it offers valuable insights into the capabilities of the model and its application in parking lot management. The focus remains primarily on the object detection pipeline, with a proper web connection and visualization left for future developments.

With the completion of this data analysis and the presentation of results, the entire system, from image capture to data analysis and reporting, is comprehensively explained. This provides a clear overview of the workflow involved in the parking lot management system, demonstrating its potential for further development and refinement.

5. Initial Deployment on Raspberry Pi

This section unveils the inaugural deployment of the parking lot monitoring system on the Raspberry Pi platform. Initially, the system's functionalities underwent development and validation on a computer, leveraging pre-existing photos of the parking lot. However, to transition the project to its intended platform, the system is implemented on the Raspberry Pi, utilizing live photos directly captured from the parking lot.

The system is configured and deployed on the Raspberry Pi, with specific adjustments made to ensure compatibility with the platform. The code functionalities, which were previously tested on the computer, are adapted for execution on the Raspberry Pi environment. While the code was developed with Raspberry Pi compatibility in mind, practical constraints limited extensive testing and optimization on the actual hardware.

To demonstrate the system's functionality and present key metrics derived from the parking lot monitoring data, a local visualization interface is developed. This user interface (UI) provides a straightforward yet effective means to visualize the metrics obtained from the data analysis. Figure 9 illustrates the implemented user interface and the results obtained during one day of operation.

The system is configured with a detection record length (***n***) of 2 and a frame capture frequency (***f***) of 15 minutes. As evident from the occupancy monitoring graph, the system effectively tracked instances over time, accurately reflecting the appearance and disappearance of vehicles in the parking lot. The labels under the graphic display the current occupancy status.

The generated metrics provide valuable insights into the current occupancy status, offering a glimpse into the system's real-time functionality. Moreover, these metrics lay the foundation for further analysis and optimization, demonstrating the potential to delve deeper into parking lot management and utilization patterns. With the data extracted through object detection and the system pipeline, future enhancements and refinements can be implemented to enhance the overall efficiency and effectiveness of the parking lot monitoring system.

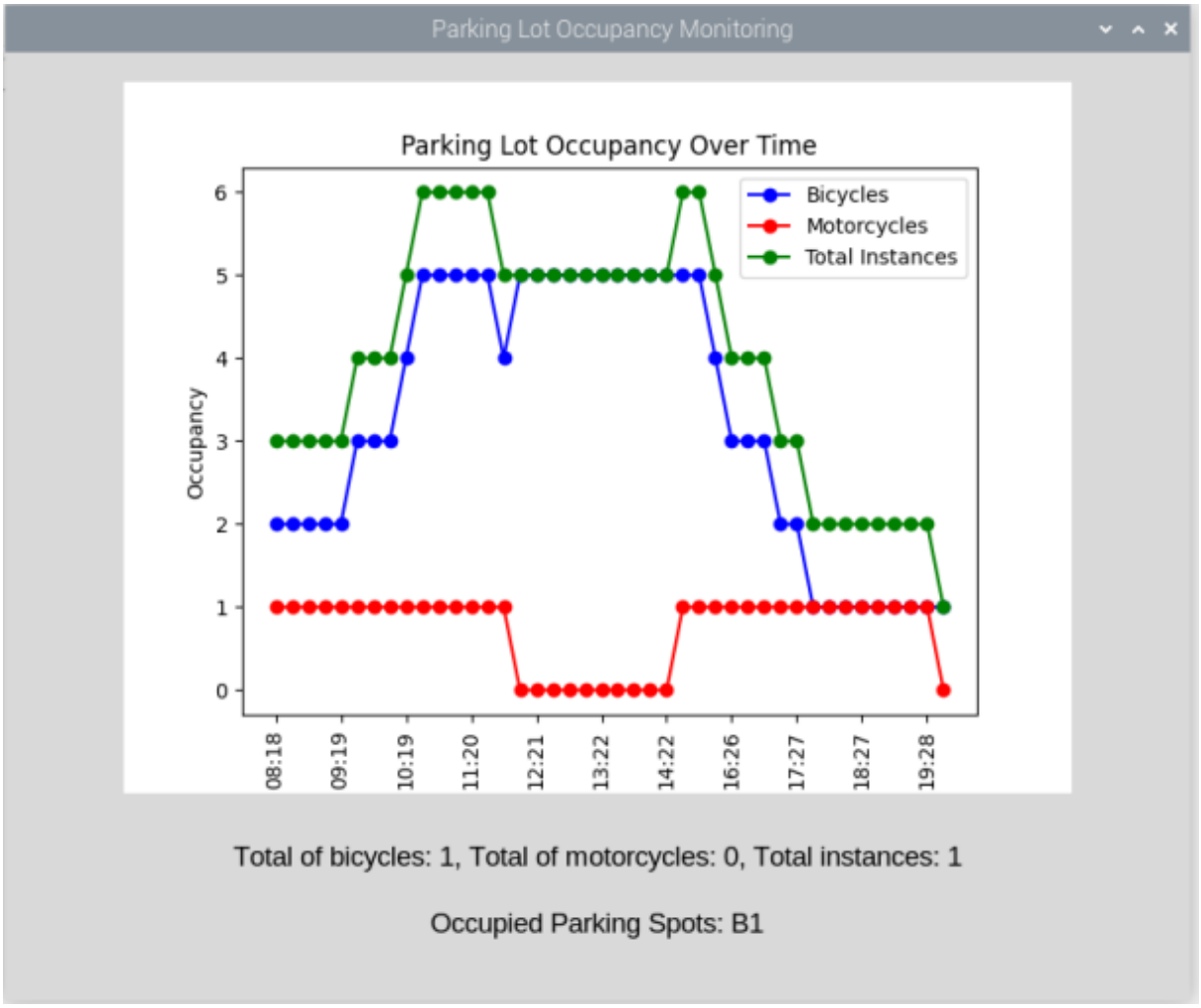

Figure 9. Occupancy Monitoring Interface.

6. Future Developments Possibilities

6.1. Enhanced Metrics Integration

As the parking lot monitoring system evolves, there are numerous opportunities to expand the range of metrics available for analysis and visualization. By leveraging the data collected through the system pipeline and store in the database, additional metrics can be calculated and integrated into the user interface (UI) for enhanced insights and decision-making. Some of the key metrics that could be implemented include:

- **Occupied Spots Visualization:** Visualizing the spatial distribution of currently occupied spots on the parking lot map. Aiding in spot monitoring and management by highlighting occupied spots in real-time.

- **Average Occupancy Duration:** Calculation of the average duration of occupancy for each parking spot and for the hole parking lot. Understanding parking spot turnover rates and assessing parking space utilization efficiency.
- **Hourly Occupancy Analysis**: By aggregating occupancy data into hourly periods, insights can be gained into peak usage hours and patterns of parking lot activity. Identifying the most utilized hours of the day can inform decisions regarding staffing, resource allocation, and parking management strategies.
- **Daily and Monthly Summaries**: Summarizing occupancy data by day or month enables higher-level analysis of occupancy trends over longer time periods. For example, analyzing monthly occupancy patterns can reveal seasonal variations in parking demand, helping to optimize staffing and resource allocation throughout the year.
- **Historical Analysis:** Historical occupancy data can be accessed and analyzed to identify long-term trends and patterns, informing strategic decision-making and future planning efforts.

By leveraging the existing data stored in the database and implementing calculations and visualizations within the user interface, is possible to integrate these enhanced metrics. This without the need for changes to the system architecture while maximizing the utilization of available data for comprehensive analysis and decision support.

6.2. Parameter Optimization

The configuration parameters of the system pipeline, specifically the detection record length (***n***) and frame capture frequency (***f***), play a critical role in determining the system's performance and accuracy. While the initial configuration of ***n*** = 2 and ***f*** = 15 minutes has demonstrated functionality, further investigation is warranted to determine the optimal parameter values for enhanced system performance.

A parameter optimization study could be conducted to systematically evaluate the impact of varying ***n*** and ***f*** values on the system's accuracy and efficiency. By testing different combinations and assessing their effects on key performance metrics such as detection precision, recall, and F1 score, as well as tracking consistency and computational efficiency, the performance of each parameter configuration can be

quantified. The study aims to identify the configuration that maximizes system performance.

6.3. Development of Web Dashboard for Data Visualization

An alternative direction for further development involves the implementation of a web-based dashboard to visualize the metrics and insights generated by the parking lot monitoring system. This option offers an opportunity to enhance accessibility and usability by providing users with remote access to real-time and historical parking lot occupancy data. Below are considerations for this development option:

- **Implementation of Web Database**:

To enable the functionality of the web dashboard, a new database system must be created to store the monitoring data securely and efficiently. Unlike the local database utilized for preliminary testing and demonstration purposes, this web-based database requires engineering to seamlessly integrate with web applications and dashboards. In this step, adjustments to the database structure can be made to accommodate the enhanced user interface requirements. However, it is important to note that there is no need to alter the pipeline structure.

- **Development of Web Dashboard:**

A web dashboard can serve as the primary interface for visualizing and analyzing parking lot occupancy data. It can feature interactive graphs, charts, and maps to present key metrics derived from the database storing occupancy data. Users can have the ability to interact with the platform, allowing them to search for specific information regarding the parking lot's occupancy trends over time or to delve into individual parking spots to view detailed statistics.

- **Considerations for Authorization and Security:**

As a web dashboard operates online, considerations must be made for authorization and security constraints. Options include implementing authentication mechanisms to control access and enforcing security measures to protect sensitive data. Furthermore, decisions may be deliberated regarding access restrictions, such as limiting access to university students or making the dashboard publicly accessible.

6.4. Daylight-Based System Operation

The system currently operates for a fixed duration, based on university operating hours from 8 AM to 7 PM. However, this fixed schedule doesn't account for changes in daylight hours throughout the year. As a result, the system may continue to capture photos even during periods of low light, leading to potentially inaccurate data collection. To address this limitation, an improvement can be made by integrating a brightness or light sensor with the Raspberry Pi. This addition would allow the system to dynamically adjust its operation based on ambient light levels. This enhancement offers several benefits, including improved accuracy by preventing data collection during low-light conditions, energy conservation through the suspension of monitoring activities in inadequate lighting, and greater adaptability to varying light conditions across different times of the day and seasons.

6.5. Recognition of Multiple Bicycles per Spot

Currently, the system is designed to recognize and count only one instance per parking spot. However, the layout of the parking spots allows for the accommodation of two instances, one on each side, as the parking lot is designed for this purpose.

To identify multiple instances parked in the same spot, it's crucial to enhance the system's performance, especially regarding recall. This improvement would entail increasing the confidence threshold for the detection task, aiming to reduce duplicate detections. By minimizing duplicates, the IoU threshold for Non-Maximum Suppression (NMS) could be raised, enabling the system to identify cases where bicycles are parked together in one spot. Although this scenario of multiple bicycles per spot is currently infrequent due to the low parking lot occupancy rate, enhancing the system in this way lays the groundwork for accurately recognizing such instances in the future.

In pursuit of improving the accuracy of the detection task, two key strategies can be implemented. Firstly, a preprocessing method can be introduced to dynamically enhance the quality of the photos captured by the system. Specifically, focusing on brightness enhancement can be particularly beneficial, as parking lots often exhibit varying and changing shadows that may obscure or partially block instances, leading to detection inaccuracies. By improving brightness, the visibility of objects in the parking lot can be enhanced, thereby improving detection accuracy.

Secondly, considering that the average execution time of the pipeline on the Raspberry Pi is approximately 18 seconds, and the frame capture frequency operates in the order of minutes, there is ample room to explore the implementation of heavier object detection models. Options such as YOLOv5s6 or YOLOv5m6, despite their increased computational demands and processing time, could potentially offer substantial accuracy improvements. This creates an opportunity to evaluate and compare the performance of models of varying sizes within the system's operational constraints.

7. Conclusions

In conclusion, this project underscores the potential of object detection technology in parking lot monitoring. By meticulously designing and implementing a detection pipeline, the system has demonstrated its ability to accurately identify and track objects within parking lots. Despite the system's high precision in reporting occupied spots, it may overlook approximately one instance for every five instances detected. Nonetheless, these occasional oversights do not significantly impact the system's overall ability to provide real-time occupancy information, offering valuable insights into parking lot utilization.

The meticulous selection of parameters, particularly the confidence threshold and Intersection over Union for Non-Maximum Suppression (NMS), was crucial in identifying the most performant model. Customizing these settings according to specific requirements ensures the effectiveness of an object detection model. However, this precision approach resulted in limitations, particularly in the system's ability to detect multiple instances within the same parking spot.

To address this limitation and enable the identification of multiple instances per spot, enhancing the accuracy of the model is essential. This involves increasing the confidence threshold to eliminate the need for duplicate elimination, thereby allowing instances close to each other to be accurately recognized. Furthermore, refining the preprocessing of images and integrating a heavier model like YOLOv5s6 offer promising avenues for improving detection accuracy, aligning perfectly with the intended design of parking spots.

Notably, given the system's average execution time on the Raspberry Pi of approximately 18 seconds and the frame frequency (f) operating on the order of minutes, there is ample room to implement a heavier model with increased inference speed while maintaining accuracy. These enhancements not only promise to enhance overall detection accuracy but also ensure the recognition of multiple instances per spot, aligning with the system's objectives.

While the current database serves its purpose, future enhancements are necessary to facilitate remote monitoring and accessibility. Integrating a web dashboard would allow students to view occupancy reports remotely, meeting modern demands for flexible data access. Implementing this improvement would require modifications to the database structure, user interface, and data analysis methods, without altering the object detection pipeline's backend infrastructure.

In summary, despite minor imperfections, the system's overall performance and reliability make it a viable solution for parking lot management. The integration of object detection technology offers a scalable and efficient approach to monitoring parking lot occupancy, paving the way for enhanced parking resource management in various settings.

IX. Bibliography

F. Liang, S. Yang, T. Mai and Y. Yang (2018). *The Design of Objects Bounding Boxes Non-Maximum Suppression and Visualization Module Based on FPGA.* IEEE 23rd International Conference on Digital Signal Processing (DSP*)*, Shanghai, China, 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631668. Retrieved 14.04.2024 from https://ieeexplore.ieee.org/document/8631668

M. Shi, P. Ouyang, S. Yin, L. Liu and S. Wei (2019). *A Fast and Power-Efficient Hardware Architecture for Non-Maximum Suppression.* IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 66, no. 11, pp. 1870-1874, Nov. 2019, doi: 10.1109/TCSII.2019.2893527. Retrieved 14.04.2024 from https://ieeexplore.ieee.org/document/8616871

L. M. Brown, R. Feris and S. Pankanti (2014), *Temporal Non-maximum Suppression for Pedestrian Detection Using Self-Calibration*. 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 2014, pp. 2239-2244, doi: 10.1109/ICPR.2014.389. Retrieved 14.04.2024 from https://ieeexplore.ieee.org/document/6977101

X. Zhang, W. Su, J. Li, J. Li and X. Lou (2021). *Spatial Non-Maximum Suppression for Object Detection using Correlation and Dynamic Thresholds.* 18th International SoC Design Conference (ISOCC), Jeju Island, Korea, Republic of, 2021, pp. 264-265, doi: 10.1109/ISOCC53507.2021.9614023. Retrieved 14.04.2024 from https://ieeexplore.ieee.org/document/9614023

Ultralytics. (2024). Configuration. Ultralytics YOLOv8 Docs. Retrieved 14.04.2024 from https://docs.ultralytics.com/usage/cfg/#predict-settings

S. B. Choi, S. -S. Lee, J. Park and S. -J. Jang (2021), *Standard Greedy Non Maximum Suppression Optimization for Efficient and High speed Inference.* IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Gangwon, Korea, Republic of, 2021, pp. 1-4, doi: 10.1109/ICCE-Asia53811.2021.9641977. Retrieved 14.04.2024 from https://ieeexplore.ieee.org/document/9641977

Klingler, N. (2024, April 18). *Object Tracking in Computer Vision (2024 Guide)*. https://viso.ai/deep-learning/object-tracking/

A. S. B. Sadkhan, S. R. Talebiyan and N. Farzaneh (2021). *An Investigate on Moving Object Tracking and Detection in Images*. 1st Babylon International Conference on Information Technology and Science (BICITS*)*, Babil, Iraq, 2021, pp. 69-75, doi: 10.1109/BICITS51482.2021.9509887. Retrieved 18.04.2024 from https://ieeexplore.ieee.org/document/9641977

N. Yang, Y. Wang and L. -P. Chau (2021). *Multi-Object Tracking with Tracked Object Bounding Box Association*. IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Shenzhen, China, 2021, pp. 1-6, doi: 10.1109/ICMEW53276.2021.9455993. Retrieved 18.04.2024 from https://ieeexplore.ieee.org/document/9455993

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., & Wang, X. (2021). *ByteTrack: Multi-Object tracking by associating every detection box*. Retrieved 18.04.2024 from https://arxiv.org/abs/2110.06864

X. Appendix

X.I. Test-Dataset Sample