Using Deep Neural Networks to Quantify Parking Dwell Time

Marcelo Marques Ribas*, Heloisa Benedet Mendes*, Luiz Eduardo de Oliveira*,
Luiz A. Zanlorensi[†], Paulo Lisboa de Almeida*

*Departamento de Informática (DInf), Universidade Federal do Paraná, Curitiba, PR - Brazil

{marcelomarques,heloisamendes,luiz.oliveira,paulorla}@ufpr.br

†DeepNeuronic, Covilhã - Portugal
luiz.zanlorensi@deepneuronic.com

Abstract—In smart cities, it is common practice to define a maximum length of stay for a given parking space to increase the space's rotativity and discourage the usage of individual transportation solutions. However, automatically determining individual car dwell times from images faces challenges, such as images collected from low-resolution cameras, lighting variations, and weather effects. In this work, we propose a method that combines two deep neural networks to compute the dwell time of each car in a parking lot. The proposed method first defines the parking space status between occupied and empty using a deep classification network. Then, it uses a Siamese network to check if the parked car is the same as the previous image. Using an experimental protocol that focuses on a cross-dataset scenario, we show that if a perfect classifier is used, the proposed system generates 75% of perfect dwell time predictions, where the predicted value matched exactly the time the car stayed parked. Nevertheless, our experiments show a drop in prediction quality when a real-world classifier is used to predict the parking space statuses, reaching 49% of perfect predictions, showing that the proposed Siamese network is promising but impacted by the quality of the classifier used at the beginning of the pipeline.

Index Terms—Deep Learning, Smart Cities, Siamese Network.

I. INTRODUCTION

Developments regarding parking space monitoring through images have been proposed in the recent past, including the development of datasets to be used as benchmarks [1]–[3], deep learning-based approaches to classify the individual parking spaces between empty/occupied [2]–[4], and the automatic segmentation of the parking spaces positions [5], [6].

Following the thread of these deep learning-based innovations, this paper proposes an approach to counting the time a car stays parked in a parking space. To accomplish this, we use images from cameras and process these images using siamese networks. By collecting data about the time each car stays parked, we may generate helpful information about, for instance, the usage of cities' parking areas.

Furthermore, it is a common practice to have parking spaces with a maximum length of stay (e.g., 15 minutes), and a system that counts the time a driver stays parked may help authorities

This work has been supported by the Brazilian National Council for Scientific and Technological Development (CNPq) – Grant 405511/2022-1, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001.

detect offenders. Other usages of such a system may include detecting abandoned or broken cars (e.g., a car parked for too long may be abandoned or broken) and the detection of illegally parked cars (e.g., a car that stayed stationary for too long in an area that is not defined as a parking space).

Quantifying the parking time is a complex problem since 1 – the vehicles' images are often collected at a distance, with low-resolution cameras (e.g., in the PKLot dataset, the bounding box of a car is 56 x 51 pixels in size, on average); 2 – vehicles may park in several positions relative to the camera; 3 – images are often collected with a low frame rate, where many minutes may pass between consecutive images collection (e.g., in the PKLot dataset one image is taken for every 5 minutes); 4 – changes in luminosity, occlusions, and shadows can make it difficult to compare the vehicles between different images. See an example of the same car parked in the same position at different times in Figure 1. Figure 2 shows a variety of car angles and luminosity differences in a parking lot.

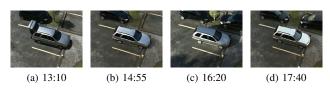


Fig. 1. A car parked in the same position at different times - PKLot dataset.

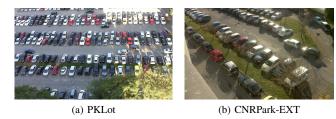


Fig. 2. Image examples from the PKLot and CNRPark-EXT datasets.

In this work, we focus on the following contributions:

- We define a siamese network to compare cars.
- We define a complete pipeline to check the parking status and then verify the dwell time of each car.

The complete pipeline includes a classification network to classify the parking spaces, a Convolutional Neural Network (CNN)-based siamese network to compare cars, and an algorithm that combines the information of both networks to update the dwell time of each parked car.

The remainder of this work is structured as follows: We show the related works in Section II. The proposed approach, including the description of the classification and siamese networks used and the algorithm used to compute the car's dwell time is given in Section III. In Section IV, we define the experimental protocol, where we define a cross-dataset scenario where no train samples from the target parking lot are given for the system. Section V shows that the proposed method can reach an Mean Absolute Error (MAE) of 46 and 54 minutes in the PKLot and CNRPark-EXT datasets, respectively. Finally, in Section VI, we present our conclusions.

II. RELATED WORKS

Calculating the duration a car remains parked in a specific location requires using two computer vision tasks: detecting and tracking objects. Deep learning-based object detection stands out as one of the most widely adopted tasks in computer vision. It can be categorized into two types: single-stage and two-stage methods. Single-stage methods, exemplified by YOLO [7], abstain from the use of a Region Proposal Network (RPN) [8] for selecting regions of interest (ROI). Consequently, they demonstrate higher speed, with object locations generated by a single CNN network. In contrast, two-stage methods like Faster R-CNN [8] and Mask R-CNN [9] rely on an RPN network for ROI generation, typically yielding more precise bounding boxes.

Deep learning techniques are also often used in tracking tasks, broadly divided into two categories – Single-Object Tracking (SOT) and Multiple-Object Tracking (MOT). The key distinction lies in that, in MOT, multiple objects are present within the target scene. Consequently, the model must address various complexities, such as object occlusion and the presence of objects with similar appearances.

The evolution of object tracking can be delineated through three stages. The initial stage, which occurred around the year 2000, primarily witnessed the application of classic algorithms and machine learning in target tracking. While these algorithms boasted features like low computational complexity, swift execution, and minimal hardware requirements, their robustness and accuracy were relatively modest.

Between 2010 and 2016, the second phase of object tracking development saw the rise of methods like MOOSE [10] and SORT [11], drawing substantial attention. This prompted researchers to delve into trackers based on correlation filtering, revealing notable advantages in speed and accuracy across various evaluation datasets. SORT has undergone further enhancements in DeepSort [12] and OCSORT [13]. Recently, Sharma et al. [14] utilized a combination of these two methods along with YoloV8 to create a real-time algorithm for tracking parking time violations using closed-circuit cameras.

The ongoing third stage, from 2016 to the present, is characterized by advancements in target tracking driven by deep learning algorithms, particularly the Siamese network. Continual enhancements in robustness and accuracy have been achieved, facilitated by the utilization of increasingly abundant datasets. This progress underscores the formidable end-to-end learning capabilities of deep learning in the realm of object tracking. A comprehensive survey of deep learning for visual tracking can be found in [15].

III. PROPOSED APPROACH

We propose using two deep neural networks in a pipeline. First, a classification network is used to classify the parking spaces between occupied and empty. The second network is a siamese comparison network, which will be used to compare cars. For both the classification and siamese Networks, we employ the large version of the MobileNetV3 [16] as the backbone (about 5M parameters) due to its tradeoff between accuracy and computational cost [17].

We defined the input size of the images for both networks as $128 \times 128 \times 3$. To compute the similarity between the input images pair for the siamese network, we use the Contrastive Loss [18], [19]. The classification network uses the Crossentropy loss function. Since the parking space positions are fixed, we use the strategy described in [1], where for each image, each parking space position is cropped as a rotated rectangle. This cropped image is fed to the networks to define its status (empty/occupied) and to check if the parked car is the same between two consecutive images.

The cropped training images and their statuses are used to train the classification network. To train the siamese networks, we consider each individual car in the training set to create the training pairs. To generate a negative pair, where the images are from different cars, we combine the image of the current car with a random image from another car. To generate a positive pair, where both images belong to the same car, we combine the car's image with an image of the same car taken at a different random time. Examples of negative and positive training samples are given in Figure 3.







(a) Negative

(b) Positive

Fig. 3. Siamese network training samples examples.

The proposed pipeline is straightforward. As we are interested in counting the time a car stays parked, we must consider the time between acquiring consecutive images. Considering that one image is taken for every k seconds, and the image of a parking space collected at the current time t, we first check if the parking space is occupied or empty using the classification network. If the parking space is occupied, we check the status of the parking space in the previous image, collected at t-k.

If the parking space was empty in the previous image, the detected car is considered a new one (the car that has just parked). Otherwise, the image of the parking space (occupied by a car) collected at time t is compared with the image of the same parking space collected at t - k using the comparison (siamese) network. If the network deems the car to be the same, the time counter for the car is increased by k. Otherwise, the car is considered as a new one.

Figure 4 illustrates this pipeline for two occupied parking spaces between times t - k and t. The complete procedure is depicted in Algorithm 1. The algorithm assumes that the classification network classified the images between empty or occupied.

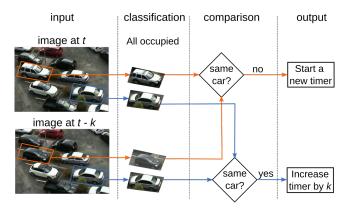


Fig. 4. Flowchart example for two parking spaces. Both parking spaces were occupied between t-k and t. The parked car in the orange parking space is not the same between images, and a new timer is created. The car in the blue parking space remained the same, and its timer was increased.

Algorithm 1: DWELLTIME(p, q, k, N).

Input: p: parking space information at the current time t; q: parking space information at the previous time t - k; k: the number of seconds between p and q; N: a trained network to compare cars.

Result: The car dwell time is updated

```
1 if p is empty then
```

return

3 if q is empty then

p.time = 0// Car just parked

return

6 result = compare(N, p.image, q.image)

7 if result is same car then

p.time = q.time + k

9 else

p.time = 0

IV. EXPERIMENTAL PROTOCOL

A. Datasets and Metrics

We use the well-known PKLot [1] and CNRPark-EXT [2] datasets for the experiments. These datasets assume that the camera is at a fixed position and that the parking space positions do not change over time. Image examples from the PKLot and CNRPark-EXT datasets can be seen in Figure 2. The time k taken between consecutive images is 5 minutes for the PKLot, whereas for the CNRPark-EXT, k is 30 minutes. A value of 30 minutes between images may be considered too long (e.g., a car may park for 20 minutes and leave without the system noticing its existence). However, we use the CNRPark-EXT for its robustness and to include two different datasets in the tests.

To create a ground truth for the experiments, we manually annotated each car with an individual identifier to track it over multiple images. We labeled 16,694 images from the PKLot and CNRPark-EXT datasets to include these new identifiers, making it possible to track 21,538 individual cars. During the experiments, the developed approach must count how long a car stays parked in a parking space. To compare the prediction with the ground truth, we use the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) values, defined as

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}_i|$$
 (1)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2}$$
 (2)

where N is the number of predictions, y_i is the actual amount of time the car stayed parked according to the ground truth, and \hat{y}_i is the predicted time.

When a single car is deemed as a new one multiple times, we use its first prediction \hat{y} to compare with the ground truth y, and the remaining predictions are compared to zero (i.e., $0 - \hat{y}$ in the equations). A similar approach is used when a car is not detected, where the ground truth will be compared with zero, as the system computed no time for the car. We also show the accuracy achieved, defined as the number of cars where the predicted parking time matched exactly with the ground-truth, divided by the total number of parked cars in the dataset.

B. Networks and Training Procedure

Both the classification and siamese (comparison) deep networks were pre-trained in the ImageNet dataset. We employed the Adam optimizer with an initial learning rate of 0.001 and a batch size of 32 for both networks. As suggested in [20], both the classification and siamese networks are trained in a cross-dataset fashion. This is important since we are putting the system to the test without considering training samples from the target parking lot, creating a realistic scenario where the system can be deployed without training it in a target scenario. First, we trained the networks using images from the PKLot dataset and tested them in the CNRPark-EXT dataset. Then, we reversed the training and test sets, training using the CNRPark-EXT images and testing using the PKLot images.

The first 70% of day images of each camera from the training dataset are used for training, and the remaining 30% for validation. Table I shows the training, validation, and testing splits¹. For each epoch, the complete training set is used

¹The number of samples available in each dataset can be different from the original datasets [1], [2] since we manually labeled new samples.

to train the classification network. Considering the siamese comparison network, since there is a quadratic number of possible training and validation pairs relative to the number of available cars in the datasets, we select 20,000 random training pairs and 20,000 random validation pairs for each epoch.

Training using PKLot test using CNRPark-EXT				
#Training	#Validation	#Test 83,655 occupied 67,972 empty		
309,381 occupied 360,459 empty	143,529 occupied 113,754 empty			
Training using CNRPark-EXT test using PKLot				
#Training	#Validation	#Test		
58,625 occupied 52,958 empty	25,030 occupied 15,014 empty	452,730 occupied 474,213 empty		

The classification and siamese networks are fine-tuned for 30 and 100 epochs, respectively. We select the model with the lowest loss in the validation set as the final model². For the classification network, for each epoch, we define a 10% chance of applying one of the following data augmentation techniques in each training image: Horizontal Flip, Gaussian Blur with a 5×9 kernel and standard deviation $\in [0.1,5]$, Random Crop, and Random Autocontrast. For the siamese network, we define a 10% chance of applying a brightness change $\in [-0.2,0.2]$ for each training pair.

We use the validation sets for both the classification and comparison networks to define the classification threshold. We defined the optimal threshold as the one that generates the Equal Error Rate (EER) point in the Receiver Operating Characteristic Curve (ROC) for the classification network. For the siamese Networks, we defined the classification threshold as the biggest threshold value with at most a 5% chance of mistaking a different pair of cars as a pair of the same car.

V. EXPERIMENTS

In Section V-A, we begin presenting the results considering a perfect parking spaces classifier (that predicts if a parking space is empty or occupied) to check the ability of the comparison (siamese) network to track cars over multiple images. Then, in Section V-B, we present the results considering the complete pipeline, considering a classification network followed by a comparison network, showing the expected behavior of the proposed approach in a real-world scenario. All results presented are an average of 5 runs.

A. Measuring the Dwell Time using the Ground Truth

For the first set of experiments, we want to analyze the ability of the trained siamese network to correctly track the presence of the same car over multiple images of a parking space. To accomplish this, we check if the parking space is occupied or empty using the information available in the ground truth since we are interested in checking the performance of the siamese network without the interference of other factors.

First, to measure the competence of the siamese networks to compare cars, for each car^3 in the testing set, we randomly selected an image of the same car (in a different moment) to create a positive pair, and a random image from a different car to generate a negative pair. Thus, in a test dataset containing n individual cars, we generated 2n testing pairs. The results of the trained networks in these testing images are shown in Table II.

TABLE II SIAMESE NETWORK RESULTS.

Training Set	Testing Set	# Test Pairs	Accuracy (stdev)
PKLot CNPark-EXT	CNPark-EXT PKLot	15,482 27,594	93.8% (1.6) 96.2% (1.8)
Total		43,076	95.3%*

^{*}Weighted average.

The results in Table II show that the networks achieved good generalizations, with accuracies of 95.3%, on average. This is an interesting result since, despite no image samples from the target parking lot being given for training, most of the car pairs were correctly recognized by the networks.

In Table III we show the results achieved by using the trained siamese networks to define the parking dwell time of each car in the test sets, using the approach discussed in Section III. As one can observe, the proposed approach achieved promising results, reaching perfect predictions for 69.1% and 77.8% in the PKLot and CNRPark-EXT tests, respectively. The MAE and RMSE results are shown in minutes. The MAE values show that, on average, the proposed approach misses the correct dwell time with an average error of 40.2 and 47 minutes for the PKLot and CNRPark-EXT datasets, respectively. It is worth remembering that in the CNRPark-EXT dataset, the time difference between two consecutive images is 30 minutes. Thus, one single mistake in detecting the same car in two consecutive images will generate an error of at least 30 minutes.

It is worth noticing that the high number of perfect predictions combined with the relatively high MAE suggests that when the system cannot correctly predict the dwell time, the errors generated tend to be large. This result is corroborated by the high RMSE values since, due to the quadratic nature of the RMSE computation, it penalizes higher differences between the ground truth and the predictions.

B. Measuring the Dwell Time using a Classifier

In this Section, we show the results of the complete proposed approach (Section III), where a network is first used to classify the parking spaces between occupied and empty,

²We also defined that the siamese networks must be trained for at least 15 epochs since the training/validation samples are taken at random for each epoch, which can create loss instabilities.

³The same car may appear over multiple images. We randomly selected just one image of the car.

TABLE III
RESULTS USING THE GROUND-TRUTH TO GET EACH PARKING SPACE
STATUS (EMPTY OR OCCUPIED).

Test Set	C	MAE (-41)	RMSE (stdev)	% perfect
	Camera	MAE (stdev)		predictions (stdev)
	UFPR04	52.9 (1.5)	111.6 (2.1)	59.4 (1.0)
PKLot	UFPR05	58.1 (1.7)	125.1 (1.3)	58.1 (1.5)
PKLOt	PUC	29.4 (1.1)	89.4 (1.9)	76.3 (0.6)
	Average	40.2 (1.3)	103.0 (1.7)	69.1 (0.8)
	cam1	47.8 (11.7)	117.1 (13.7)	76.6 (5.9)
	cam2	33.2 (20.4)	93.0 (30.3)	82.7 (9.1)
CNRPark-EXT	cam3	42.2 (12.7)	107.9 (13.8)	78.7 (7.2)
	cam4	44.4 (11.2)	114.0 (12.2)	78.5 (6.1)
	cam5	43.4 (9.9)	114.5 (9.6)	80.1 (5.7)
	cam6	47.5 (9.1)	121.5 (8.1)	77.8 (5.5)
	cam7	53.7 (7.5)	127.5 (6.9)	75.7 (4.5)
	cam8	44.9 (9.2)	113.4 (10.1)	78.6 (4.9)
	cam9	57.5 (12.4)	129.8 (11.2)	73.5 (6.4)
	average	47.0 (10.4)	117.8 (10.8)	77.8 (5.8)

and the siamese network is used to compare the cars. This test was developed to answer how a classifier that predicts the parking space status (and may incorrectly classify it) can hinder vehicle dwell time accounting.

First, in Table IV, we show the averaged classification results of the classification networks when put to the test in the target (testing set) parking lot to classify its parking spaces between occupied and empty. As one can observe, we reached an average classification accuracy of 92.4%, which is similar to the results achieved by most of the methods in the state-of-the-art when we consider a cross-dataset scenario [17], [20].

TABLE IV CLASSIFICATION NETWORK RESULTS.

Training Set	Testing Set	# Test Samples	Accuracy (stdev)
PKLot CNPark-EXT	CNPark-EXT PKLot	160,298 926,943	93.0% (0.24) 92.3% (0.15)
Total		1,087,241	92.4%*

^{*}Weighted average.

In Table V, we show the results of the complete pipeline, using the classification networks to verify if the parking space is occupied or empty, to only then use the siamese networks to check if the car is the same between images and increase the dwell time of the vehicle if necessary. As one can observe, despite the high accuracy of the classification networks, there was a substantial drop in all metrics when compared with the results available in Section V-A.

This severe drop in the metrics can be explained by the fact that, on average, a car can be parked for several hours in both datasets. On average, the same car is present in 25 (2 hours and 5 minutes) and 12 (6 hours) consecutive images in the PKLot and CNRPark-EXT datasets, respectively. One example is given in Figure 6. Considering the example, if the classifier incorrectly predicts a single image of the car as a free parking space, the dwell time computation can be severely impacted, especially if the misclassified image is near the beginning of the sequence, since the timer for the current car will be stopped

TABLE V
RESULTS USING THE COMPLETE PIPELINE.

Test Set	Camera MAE (std	MARKAL	r) RMSE (stdev)	% perfect
		MAE (stdev)		predictions (stdev)
PKLot	UFPR04	68.7 (5.5)	141.8 (9.2)	21.7 (5.4)
	UFPR05	75.2 (13.9)	151.6 (21.5)	18.3 (5.5)
	PUC	34.5 (2.6)	96.2 (5.9)	56.9 (6.6)
	Average	53.3 (5.4)	124.6 (9.2)	38.4 (4.8)
CNRPark-EXT	cam1	107.9 (6.4)	175.2 (6.5)	43.8 (2.2)
	cam2	46.2 (3.9)	114.2 (5.6)	75.6 (2.3)
	cam3	82.8 (3.9)	158.2 (3.5)	57.3 (2.5)
	cam4	66.5 (3.5)	136.5 (4.4)	61.5 (1.1)
	cam5	86.2 (4.2)	156.2 (4.9)	52.0 (1.5)
	cam6	80.2 (3.6)	151.6 (4.1)	56.8 (0.9)
	cam7	101.6 (5.0)	170.0 (4.5)	46.5 (2.6)
	cam8	74.4 (1.7)	144.0 (2.4)	58.8 (0.7)
	cam9	88.4 (5.7)	156.9 (7.4)	51.9 (1.7)
	average	84.5 (1.4)	155.0 (1.8)	54.3 (0.7)

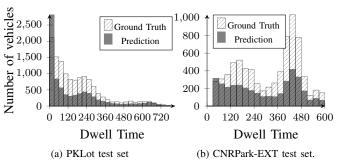


Fig. 5. Car's Dwell Time Histogram using the a) PKLot and b) CNRPark-EXT datasets as test sets.

much earlier than it should and a timer may (incorrectly) be started for a new car.



Fig. 6. A car that stay parked for several hours - CNRPark-EXT dataset.

This behavior can be seen in Figure 5, where we show histograms of the number of cars parked for different time spans considering both the PKLot and CNRPark-EXT datasets. The proposed method overestimated the number of cars with a low stay times, between 0 and 30 minutes, for both the PKLot and CNRPark-EXT datasets. This can be explained by the car's chronometers being stopped prematurely due to a misclassification of the classification network or an error generated by the siamese network.

C. Lessons Learned

The time span between consecutive images impacts the results: As discussed in this work, we consider the time span between two consecutive images taken as k. Large values of k can lead to more lightweight systems since the number of images taken and processed by minute can be reduced.

Nevertheless, large values of k can lead to larger errors since one misclassified image can lead to an error of at least k seconds. Moreover, the greater the value of k, the more changes between consecutive images may occur, hindering the ability of the siamese networks to compare cars.

The classification network is a key point in generating better results: Even a classification network with a relatively small error (7.6% in our experiments) can severely decrease the estimation of the dwell time since a car stays parked for several hours, and one single misclassification can prematurely stop the chronometer for a given car, and start a chronometer for a new car that does not exist in reality. As discussed in [20], a key point that should be tackled is the development of generic classifiers with accuracies near 100%, which could significantly improve the performance of our system. Classifiers that achieve accuracies of nearly 100% do exist [1], [2], [20]. Nevertheless, such classifiers need training samples from the target parking lot. This work focused on a general approach that does not need training samples from the target parking lot. When training samples from the target parking lot are available, we should expect results near the ones shown in Section V-A, where a hypothetical perfect classifier was used.

When the system gets things wrong, it really gets it wrong: The RMSE values are much bigger than the MAE values in all experiments. This indicates that for some difficult samples, the system can generate big errors, while for most samples, the errors are relatively small (the RMSE penalizes bigger errors due to its quadratic nature). These difficult samples should be taken into account in future works.

VI. CONCLUSION

In this work, we showed a complete pipeline that includes a deep classification network followed by a siamese network to compute the dwell time of vehicles in parking spaces. We evaluated the proposed approach in a cross-dataset scenario, which is the most challenging scenario for vision-based parking spaces monitoring systems [20], where no training samples from the target parking lot are given. To the best of our knowledge, this is the first work that proposes a pipeline to compute the dwell time of cars using images that define a robust protocol, including a cross-dataset scenario and experiments using large volumes of data and the use of wellknown metrics, such as MAE and RMSE.

Our experiments showed promising results for the proposed approach regarding the Siamese networks responsible for comparing cars between images. Our results show that such a system can perfectly compute the dwell time for 69% and 78% of the cars for the PKLot and CNRPark-EXT datasets, given that a hypothetical perfect classification network can tell if each parking space is occupied or empty. It is worth mentioning that many classification networks can reach accuracies near 100%, given that training samples from the target parking lot are given [1], [2], [20].

As we focus on cross-dataset scenarios, we included the experiments regarding the complete pipeline, where a classification network that did not receive training samples from the

target parking lot is employed to classify each parking space before computing the dwell time. Our results show that even a network with a relatively small error (7.6% in our tests) can severely deteriorate the results since the chronometer of the cars can be stopped much earlier than they should. This indicates that, as suggested in [1], parking space classification networks that do not rely on train instances of the target parking lot should be improved to reach accuracies, at least near to the ones trained in the target parking space.

For future work, we plan to improve the training of both the siamese and classification networks to reduce the error in the cross-dataset scenario and the total error of the complete pipeline, taking into consideration the Algorithm III.

REFERENCES

- [1] P. R. Almeida, L. S. Oliveira, A. S. Britto Jr, E. J. Silva Jr, and A. L. Koerich, "Pklot-a robust dataset for parking lot classification," Expert Systems with Applications, vol. 42, no. 11, pp. 4937-4949, 2015.
- [2] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," Expert Systems with Applications, vol. 72, pp. 327-334, 2017.
- [3] R. M. Nieto, Á. García-Martín, A. G. Hauptmann, and J. M. Martínez, "Automatic vacant parking places management system using multicamera vehicle detection," IEEE ITS, vol. 20, no. 3, pp. 1069-1080, 2018.
- [4] V. Dhuri, A. Khan, Y. Kamtekar, D. Patel, and I. Jaiswal, "Real-time parking lot occupancy detection system with vgg16 deep neural network using decentralized processing for public, private parking facilities," in ICAECT, 2021, pp. 1-8.
- [5] P. Almeida, L. S. Oliveira, and J. Honorio, "Vehicle occurrence-based parking space detectio," in 2023 IEEE International Conference on Systems, Man, and Cybernetics, 2023, pp. 3603-3608.
- [6] R. Grbić and B. Koch, "Automatic vision-based parking slot detection and occupancy classification," Expert Systems with Applications, vol. 225, p. 120147, 2023.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi., "You only look once: Unified, real-time object detection," in IEEE CVPR, 2016, pp.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards realtime object detection with region proposal networks." in NIPS, 2015, pp. 91-99
- [9] K. He, P. D. G. Gkioxari, , and R. Girshick, "Masked R-CNN," in Int. Conf. on Compuer Vision, 2017, p. 2961-2969.
- [10] D. Bolme, J. Beveridge, B. Draper, and Y. Lui, "Simple online and realtime tracking," in *CVPR*, 2010, pp. 2544–2550.

 A. Bewley, F. R. Z. Ge, L. Ott, and B. Upcroft., "Simple online and
- realtime tracking," in ICIP, 2016.
- [12] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in ICIP, 2017, pp. 3645-3649.
- [13] J. Cao, J. Pang, X. Weng, R. Khirodkar, and K. Kitani, "Observationcentric sort: Rethinking sort for robust multi-object tracking," in CVPR, 2023, pp. 9686-9696.
- N. Sharma, S. Baral, M. P. Paing, and R. Chawuthai, "Parking time violation tracking using yolov8 and tracking algorithms," Sensors, vol. 23, 2023
- [15] S. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," IEEE ITS, vol. 23, pp. 3943-3968, 2021.
- A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan et al., "Searching for mobilenetv3," in Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 1314-1324.
- A. G. Hochuli, J. P. Barddal, G. C. Palhano, L. M. Mendes, and P. R. Lisboa de Almeida, "Deep single models vs. ensembles: Insights for a fast deployment of parking monitoring systems," in ICMLA, 2023, pp.
- [18] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 2005, pp. 539-546.

- [19] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2006, pp. 1735–1742.
 [20] P. Almeida, J. Alves, R. Parpinelli, and J. Barddal, "A systematic review on computer vision-based parking lot management applied on public datasets," *Expert Systems With Applications*, vol. 198, p. 116731, 2022.