An Efficient YOLOv7x Based Automated Street Parking Space Detection for Smart Cities

Tala Bazzaza, Hamid Reza Tohidypour, Yixiao Wang, and Panos Nasiopoulos
Electrical & Computer Engineering, Univeristy of British Columbia
Vancouver, BC, Canada
email: tbazzaza@student.ubc.ca, {htohidyp, yixiaow, panosn}@ece.ubc.ca

***Abstract*— Finding available street parking spots is a cause of increased traffic in metropolitan cities. To address this challenge, in this paper, we propose a unique real-time street parking detection scheme that utilizes visual information and object recognition to accurately detect empty street parking spots. We also introduce a comprehensive video dataset that is captured specifically for this task and is used for training our networks. Among several network options for localization, our tests on YOLOv7 achieved the highest accuracy and speed, making it an ideal choice for real-time street parking detection for human driven as well as autonomous vehicles.**

Keywords: street parking detection; deep learning, YOLOv7; real-time performance; object recognition.

I. INTRODUCTION

As cities continue to grow and urbanize, traffic congestion has become an increasingly common problem. In metropolitan cities, it is estimated that 30-50% of traffic congestion is caused by drivers searching for parking spots during peak hours [1][2]. This not only leads to waste of fuel and increased levels of pollution, but also to a significant reduction in productivity due to driving aimlessly multiple times around city blocks in search of a vacant parking spot [3]. To address this issue and to improve traffic management, an efficient and functional street parking detection system is needed to deploy the vision of smart cities. This system will direct drivers towards vacant parking spots around the block, therefore reducing unnecessary delays which worsen traffic conditions [4][5]. Previously, methods that rely on ultrasonic sensors were used to quantify the target area by using a virtual grid map and establishing a coordinate system for parking spot detection [6]. Other methods have utilized autonomous sensor nodes with Wireless Sensor Networks (WSNs) for monitoring parking occupancy in lots [7][8]. A previous study utilizes video surveillance camera data to detect parking spots using Support Vector Machines (SVM) and k-nearest neighbors algorithms [9]. Although this method produced results with high accuracy, it is not practical for on-street parking detection as it uses aerial views captured by video surveillance cameras which do not necessarily cover the city streets. Nowadays, many researchers are using computer vision techniques and deep learning methods to detect available parking spots. In [10], the authors use instance segmentation algorithms and convolutional neural networks to perform real time processing on data to determine if the parking spot is vacant. These methods are also more scalable and robust than the traditional sensor-based systems, as they can work under different weather conditions, lighting conditions, and camera angles. However, there are still some challenges that need to be addressed, such as occlusions, different parking spot sizes and shapes, and variations in parking spot markings.

This paper explores the use of the latest You Only Look Once (YOLO) network architecture, namely YOLOv7, for accurately detecting available street parking spaces. To achieve this, we have created a new and extensive video dataset of city street parking spaces and we have trained and fine-tuned the network on this dataset. We compare the performance of our network against the state-of-the-art approach presented in [11], which is based on YOLOv4. Evaluation results show that our YOLOv7 outperforms YOLOv4 in terms of Mean Average Precision (mAP) at different threshold levels of overlap between the predicted bounding box and the ground-truth bounding box. More specifically, YOLOv7 reached a mAP of 89.9% at a threshold of 50% overlap, while YOLOv4 reached a mAP of 83.3% at the same threshold. Additionally, YOLOv7 showed faster inference time and better generalization ability than YOLOv4. These results demonstrate the superiority of YOLOv7 and pave the way for its use in real-time street parking detection for human driven as well as autonomous vehicles.

The rest of this paper is organized as follows. Section II describes the methodology of our proposed method. Section III describes the results and evaluation. In Section IV, conclusion and future work are presented. The acknowledgement closes the article.

II. OUR PROPOSED METHOD

A. Dataset and labelling

In this study, we used a dataset consisting of 55 videos captured by our team in the city of Vancouver, Canada. The dataset was carefully curated to include a diverse range of weather conditions (sunny, cloudy, rainy, snowy) and location scenarios. To prepare the dataset for training and evaluation, we utilized the Fast Forward Moving Picture Experts Group (ffmpeg) tool to extract frames from the videos, which were then labeled using the Computer Vision Annotation Tool (CVAT) software [12][13]. This allowed us to create a dataset that is representative of real-world scenarios and provides a robust evaluation of the performance of the object detection models.

In our labeling technique, we decided to implement a single class, labeling only available parking spaces. We labeled parking spots that are within a distance of 5 meters from the car, and only focused on the right side of the street. This approach helps eliminate double counting parking spots and more accurately determining if the parking space is long enough to fit a car. In addition, using unlabeled data during training can introduce the model to learn features that are not related to vacant parking spots such as intersections, bus stops, yellow curbs, and non-vacant parking spots. These unlabeled frames

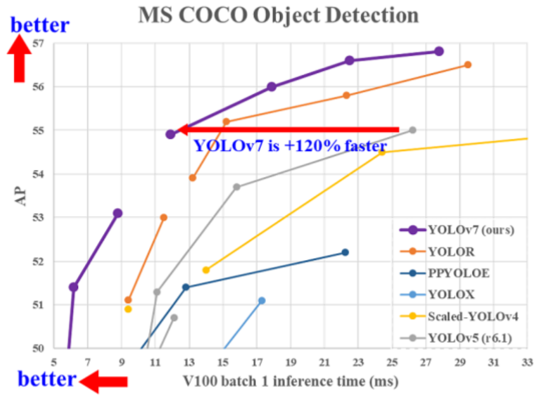


Figure 1. Comparison of YOLOv7 with previous object detection networks [16].

were included along with empty text files as there are no coordinates highlighted. We created a dataset that contains a total of 3381 frames of city street parking spaces. It is well established that a balanced dataset of labeled and empty (no-label present) frames yields the best training results for object detection [14][15]. For this reason, the dataset was composed of 1776 frames that were labeled as available parking spots and 1605 frames that were left unlabeled (empty text files) to be used during the training process. This dataset was split in 85% for training and 15% for validation. Additionally, a different and unseen dataset consisting of 278 frames from the city of Vancouver was used for testing. The testing dataset is used to evaluate the performance of our proposed model.

B. *Our network*

We chose the YOLOv7 family architecture as the basis for our network. It is worth noting that YOLOv7 has made significant advancements to the previous YOLO models both on the architectural level and at the trainable bag-of-freebies level, which improves the accuracy of the model without increasing the cost of training. Overall, YOLOv7 has a more efficient architecture that reduces the number of parameters by 75%, thus requiring 36% less computation and achieving 1.5% higher Average Precision (AP) compared to the previous models [15]. Figure 1 compares YOLOv7 with other real-time object detectors. We observe that YOLOv7 achieves state-of-the-art performance with improved accuracy and lower complexity.

YOLOv7, like the entire family of this architecture, uses data augmentation techniques to increase the size of the training dataset and improve the generalization of the model, which as a result avoids possible overfitting. However, since in our implementation we are mainly focused on the right side of the street when locating parking spots, some augmentation techniques like flipping and rotation are not applicable as they will produce erroneous images (i.e., parking on the left side or cars upside down) and for this reason they were disabled. Nonetheless, we implemented a modified version of the Mosaic data augmentation introduced by the inventors of YOLOv4, selectively combining four images to generate a new one, but making sure that we are not violating our requirement to have parking spots only on the right side [14]. This technique proved to help our YOLOv7 based network to learn more features and become robust to different lighting conditions, camera angles, and object scales.

We decided to train two versions of the YOLOv7 family, the original YOLOv7 and the latest YOLOv7x. The main differences between the two is that YOLOv7 uses the method of stack scaling on the neck, which is a technique to increase the capacity of a model by adding more layers to it. In this case, this technique is applied to the "neck" of the model, which is a key component of the architecture that helps to extract features from the input image. By stacking more layers on the neck, the model is made more powerful and able to detect more complex objects.

On the other hand, YOLOv7x in addition to the neck scaling scheme of YOLOv7 performs compound scaling on the neck, which increases the depth and width of the entire model simultaneously, as opposed to only increasing one or the other, leading to an improved performance.

Anchor boxes are a key component of object detection algorithms, as they are used to predict the location and size of objects in an image. YOLOv7 uses an auto-anchor algorithm borrowed from YOLOv5, which adapts to the scale of the objects in an image by using a single anchor box that can adjust to different scales [15][17]. To this end, before the training process begins, the suitability of the provided anchors for the dataset is evaluated. If the fit is not optimal, new anchors are recalculated that are more suitable for the data. The model is then trained using these newly generated, more appropriate anchors.

III. RESULTS AND EVALUATION

We compared our suggested network with the state-of-the-art method presented in [11]. In order to fairly evaluate the performance of the YOLOv4 network used in [11], we had to retrain it using our new and more comprehensive dataset.

We had to address the limitation of YOLOv4 in generating anchor boxes by using k-means clustering. Since we are using a custom dataset, we generated anchor boxes based on the aspect ratio and scale of the objects in our dataset before starting the training process in YOLOv4. The new calculated anchor boxes were added manually in each of the yolo-layers while configuring our model.

Regarding our proposed approach, we first trained our YOLOv7 and YOLOv7x networks using the computing clusters available by Compute Canada [18]. We started training YOLOv4, YOLOv7 and YOLOv7x with the pretrained weights of the darknet framework and the pretrained weights of YOLOv7 and YOLOv7x [14][15].

Performance evaluation and accuracy of three models is done using the mean Average Precision (mAP) metric. Average precision is calculated by measuring the precision and recall of the model at different intersection-over-union (IoU) thresholds,

TABLE I. VALIDATION RESULTS OF ALL THE MODELS

Model	mAP @ 0.5
YOLOv4	82%
YOLOv7	84%
YOLOv7x	90%

TABLE II. TESTING RESULTS OF ALL THE MODELS

Model	Precision	Recall	mAP @ 0.5
YOLOv4	0.84	0.81	83.3%
YOLOv7	0.90	0.79	86.5%
YOLOv7x	0.91	0.81	89.9%

which is the ratio of the area of overlap between the predicted bounding box and the ground-truth bounding box to the area of the union of the two boxes. In this paper, we compare the models at an IoU threshold of 0.5. Table I shows the performance of our trained models on the validation set.

We observe that the YOLOv4 model scored a mean Average Precision of 82% for the validation set. On the other hand, both versions of YOLOv7 outperformed YOLOv4, achieving mAP@0.5 of 84% and 90%, respectively.

We also observed that YOLOv7x achieved the best weights at (mean Average Precision) mAP@0.5 with levels of 90% - see Fig. 2 that shows the precision-recall curve.

In order to evaluate the performance of our models against that of YOLOv4 for unseen test data, we tested all of them on 278 previously unseen frames captured by our lab in the city of Vancouver. Table II shows results of detecting vacant parking spots. We observe that YOLOv7x achieves the best performance, reaching a mAP of 89% at a threshold of 50% overlap between the labeling bounding box and the predicted one. Overall, our models performed significantly well in all different areas of the road such as main street, crosswalks, and intersections. and other type of side entrances that could be confusing even for a human. Figure 2 below shows the precision-recall curve for the testing set of YOLOv7x.

Figs. 3a and 3b show two representative examples of street parking detection performed by YOLOv4 and YOLOv7x respectively. It is obvious from Fig.3a that the YOLOv4 model was unable to detect parking spots in some instances where the car was driving on the same lane as the parking lane (no bounding box is present). However, Fig. 2b shows that the YOLOv7x model accurately detected the available parking spaces (purple bounding box). The Intersection of Union (IoU) score displayed above the bounding boxes represents the degree of overlap between the labeled bounding box and the one predicted by the model. As our goal is to detect street parking

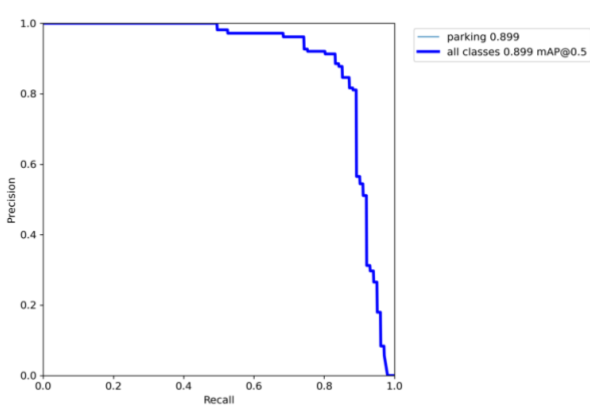
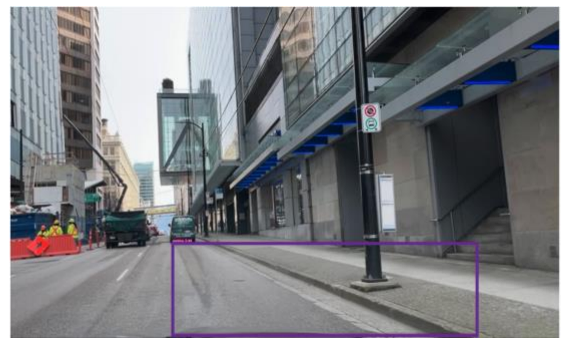


Figure 2. YOLOv7x Precision-Recall Curve.

(a)



(b)

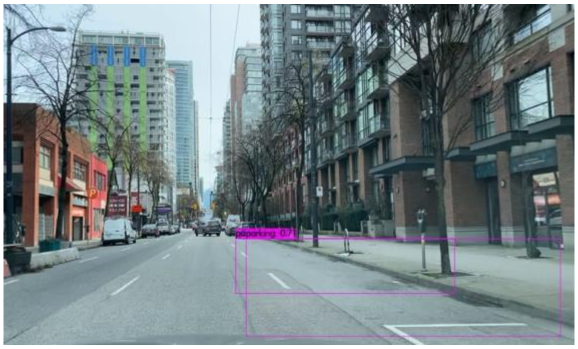
Figure 3. a) YOLOv4 testing failed to identify some parking spots; b) YOLOv7x successfully detected parking spaces.

spaces, rather than the precision of the bounding box placement, we can infer that the model is highly effective at identifying available parking spaces in the given frame.

Fig. 4a shows an example where the YOLOv4 model detects a parking spot twice, unlike YOLOv7x (see Fig. 4b) where the model correctly detects the available parking spaces as one.

IV. CONCLUSION

In this paper, we proposed a new and innovative real-time street parking detection scheme that is based on the latest YOLO architecture, namely YOLOv7x. The network was trained on a new dataset mainly captured by our team and was designed to receive video input from a car mounted camera. We labeled parking spots that are within a distance of 5 meters from the car, and only focused on the right side of the street. This approach helps eliminate double counting parking spots and more accurately determining if the parking space is long enough to fit a car. Performance evaluations have shown that the YOLOv7x model outperforms the state-of-the-art YOLOv4 based approach in terms of both accuracy and detection. The performance of our model could be significantly improved by increasing the size and variety of our dataset. Future work will include new motion detection techniques that calculate how much a frame has shifted using global motion vectors to help analyze how many frames should be skipped after detecting a parking spot to find the next processing frame that has a parking spot. Additionally, we plan to add a separate network for detecting parking signs to provide a comprehensive solution that can be integrated into smart city infrastructures.



(a)

(b)

Figure 4. a) YOLOv4 detecting a parking spot more than once. b) YOLOv7x accurately detecting a single parking spot.

ACKNOWLEDGMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC – PG 11R12450), and TELUS (PG 11R10321).

REFERENCES

[1] R. Arnott and E. Inci, "An integrated model of downtown parking and traffic congestion," Journal of Urban Economics, vol. 60, no. 3, pp. 418–442, 2006.

[2] J. Polak and P. Vythoulkas, "An assessment of the state-of-the-art in the modelling of parking behaviour," TSU REF, vol. 752, p. 31, 1993.

[3] MD. Meyer and M. McShane, "Parking policy and downtown economic development," J Urban Plan Dev ASCE, vol.109, no.1, pp. 27-43, 1983.

[4] V. Paidi, H. Fleyeh, J. H˚akansson, and R. G. Nyberg, "Smart parking sensors, technologies and applications for open parking lots: a review," IET Intelligent Transport Systems, vol. 12, no. 8, pp. 735–741, 2018.

[5] D. G. Chatmana and M. Manville, "Theory versus implementation in congestion-priced parking: An evaluation of SFpark, 2011–2012," Research in Transportation Economics, vol.44, pp. 52-60, 2014.

[6] Y. Shao, P. Chen, and T. Cao, "A Grid Projection Method Based on Ultrasonic Sensor for Parking Space Detection," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, pp. 3378-3381, 2018.

[7] E. Sifuentes, O. Casas, and R. Pallas-Areny, "Wireless Magnetic Sensor Node for Vehicle Detection with Optical Wake-Up," in IEEE Sensors Journal, vol. 11, no. 8, pp. 1669-1676, Aug. 2011.

[8] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: An application of optical wireless sensor network," in Proc. Int. Symp. Appl. Internet Workshops, p. 66, 2007.

[9] X. Sevillano, E. Màrmol, and V. Fernandez-Arguedas, "Towards smart traffic management systems: Vacant on-street parking spot detection based on video analytics." in 17th International Conference on Information Fusion (FUSION), pp. 1-8, 2014.

[10] B. Sairam, A. Agrawal, G. Krishna, and S. P. Sahu, "Automated Vehicle Parking Slot Detection System Using Deep Learning," *2020 Fourth* International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 750-755. 2020.

[11] T. Bazzaza *et al*., "Automatic Street Parking Space Detection Using Visual Information and Convolutional Neural Networks," *2022 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, pp. 01-02, 2022.

[12] FFmpeg Developers. (2016). ffmpeg tool (Version be1d324) [Software]. Available from http://ffmpeg.org [retrieved February, 2023].

[13] OpenVINO Toolkit, 2020. GitHub repository. Available from https://github.com/openvinotoolkit/cvat. [retrieved February, 2023].

[14] AlexeyAB version of Darknet, 2020. GitHub repository. Available from https://github.com/AlexeyAB/darknet. [retrieved January, 2023].

[15] C.-Y. Wang and A. AB implementation of YOLOv7. GitHub repository. Available from https://github.com/WongKinYiu/yolov7 [retrieved February, 2023].

[16] G. Boesch, "YOLOv7: The Most Powerful Object Detection Algorithm (2023 Guide)," Available from https://viso.ai/deep-learning/yolov7-guide/. https://viso.ai/deep-learning/yolov7-guide/ [retrieved February, 2023].

[17] G. Jocher ultralytics version of YOLOv5. GitHub repository. Available from https://github.com/ultralytics/yolov5. [retrieved February, 2023]

[18] Compute Canada state-of-the-art advanced research computing network. Available from: https://www.computecanada.ca [retrieved February, 2023].