International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2020 IJSRCSEIT | Volume 6 | Issue 2 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT206276

Cost effective Parking System Using Computer Vision

Kaushal Shah, Shivang Rajbhoi, Nikhil Prasad, Charmi Patel, Roushan Raj
Computer Science and Engineering, Parul Institute of Technology, Vadodara, Gujarat, India

ABSTRACT

This paper presents an approach for detecting real-time parking slots which includes vision-based techniques. Traditional sensor-based systems are not cost effective as 'n' number of sensors are required for 'n' parking slots. Transmitting sensor data to central system is done by hardwiring or installing dedicated wireless system which is again costly. Our technique will overcome this problem by using camera instead of number of sensors which is expensive. For detection we are using a Convolutional Neural Networks (CNN) classifier which is custom trained. It is more robust and effective in changing light conditions and weather. The following system do not require high processing as detections are done on static images not on video stream. We have also demonstrated real-time parking scenario by constructing a small prototype which shows practical implementation of our system.
Keywords : Convolutional Neural Networks, You Only Look Once, Deep learning.

I. INTRODUCTION

Since last 10 years, there is a huge increase in number of vehicles. But the current transportation infrastructure and car park facilities are insufficient in sustaining the influx of vehicles on the road. Therefore, problems such as traffic congestion and insufficient parking space inevitably crops up [1]. Car parking occupancy detection are of great importance for an effective management of car parking lots. Knowing in real-time availability of free parking spaces and communicating to the users can be of great help in reducing the queues, improve scalability and the time required to find an empty space in a parking lot [2]. An important requirement for these systems is the ability to detect nearby parking slots automatically, which is becoming an increasingly difficult and critical task and therefore the traffic increases [3]. Various measures have been taken in the attempts to overcome the traffic problems. Some methods like sensor-based and computer vision classifiers are functional nowadays to reduce the problems regarding parking. But these systems are either very costly or less efficient and less robust. Current detection systems repurpose classifiers to perform detection. To detect an object, these systems take a classifier for that object and evaluate it at various locations and scales in a test image. Systems like deformable parts models (DPM) use a sliding window approach where the classifier is run at evenly spaced locations over the entire image [4].

We are using YOLO (You Only Look Once), which is an object detection algorithm based on Convolutional Neural Network (CNN). YOLO is refreshingly fast as well as simple and YOLO trains on full images and directly optimizes detection performance.

Since we frame detection as a regression problem, we don’t need a complex pipeline. We simply run our neural network on a new image at test time to predict detections [5]. YOLO reasons globally about the image when making predictions. Unlike sliding window and region proposal-based techniques, YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance [6]. After digging a little more we found the concept of neural networks which can be used in detection of objects. YOLO (you only look once) is an object detection algorithm which is based on convolutional neural networks. Which is more robust and accurate than classifiers. So we have implemented YOLO in our project for the detection of cars. After the detection of cars it is checked whether a specific car is present at a particular parking slot or not.

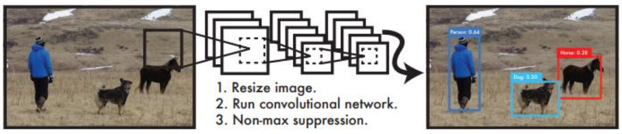

Figure 1

The YOLO Detection System. Processing images with YOLO is simple and straightforward. The system (1) resizes the input image to BLOB size, (2) runs a single convolutional network on the image, and (3) thresholds the resulting detections by the model's confidence [5].

With YOLO which gives a robust solution and it is very helpful in situations like disturbances due to partial occlusions, presence of shadows, variation of light conditions, and exhibits a good generalization property. In fact, the quality of the results is maintained when we considered parking lots and scenarios significantly different from the ones used during the training phase. Convolutional Neural Networks architecture (CNN) are similar to the human neural network build with synapses (weights) and neurons. From this point of view, complex tasks can be learned through the network. This uses CNN with pre-existing architectures to detect in real-time the vacancy of a parking spot [7].

II. YOLO OBJECT DETECTION PERFORMANCE

There are number of object detectors like Faster R-CNN and retina nets but YOLOv3 outperforms almost all the object detectors. Faster R-CNN currently is the most accurate one but it is not faster so YOLOv3 is considered first.

AP (Average precision) is a popular metric in measuring the accuracy of object detectors like Faster R-CNN, SSD, etc. Average precision computes the average precision value for recall value over 0 to 1. Precision measures how accurate is your predictions. i.e. the percentage of your predictions are correct [8].

Table – 1. Performance of each version of YOLO [9]

Model	Layers	FLOPS (B)	FPS	mAP	Dataset
YOLOv1	26	not reported	45	63.4	VOC
YOLOv1-Tiny	9	not reported	155	52.7	VOC
YOLOv2	32	62.94	40	48.1	COCO
YOLOv2-Tiny	16	5.41	244	23.7	COCO
YOLOv3	106	140.69	20	57.9	COCO
YOLOv3-Tiny	24	5.56	220	33.1	COCO

III. CNN IN OBJECT DETECTION

CNN is a concept of Deep Learning and its work on neural network. Convolutional neural network is one of the main categories to do images recognition, images classifications. Objects detections, recognition faces etc., are some of the areas where CNNs are widely used. There are some layers present in CNN in which each input image passed.

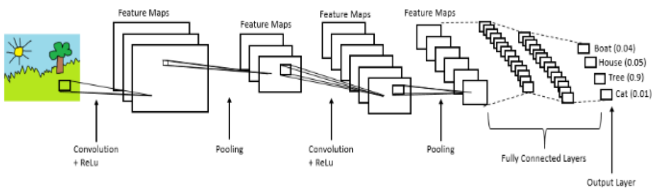

Figure 3. Architecture of CNN [10]

In the above diagram architecture of CNN shown. First, we Provide input image into convolution layer then Choose parameters, apply filters with strides, padding if requires. Perform pooling to reduce dimensionality size, add as many convolutional layers until satisfied then flatten the output and feed into a fully connected layer and at last output the class using activation function.

IV. TRAINING OF MODEL

Two files are required for object detection for this approach. One is yolo.weights file and other is cfg file. A CFG file stores settings and configuration information about the layers. Initially our system should be tested with YOLO pre-trained weights which is trained on huge COCO dataset. The COCO dataset is an excellent object detection dataset with 80 classes, 80,000 training images and 40,000 validation images. And if the model does not satisfy the required system the model should be custom trained as per YOLOv3 method.

Image should be collected and different atmospheric conditions as well as lightening conditions should be taken in consideration while capturing the images. For more accurate model maximum dataset images are required (as much as possible).

Figure 4. Training Model [11]

Once image acquisition is completed then images are labelled using labelImg-master tool. The above image is a snapshot of labelImg-master tool which is used to label car image which is a part of our prototype. After the labelled images and annotations file should be converted in darknet format and trained. Our prototype model is trained via Google Colab.

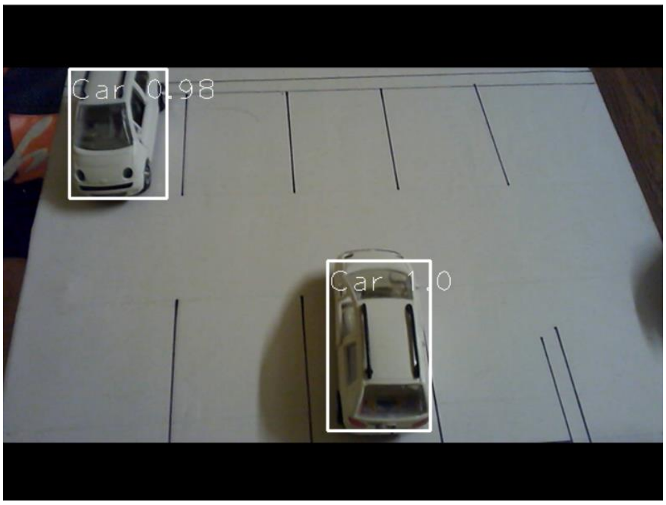

Figure 5

Figure shows the confidence of our model after we trained it to 3000 epochs. The confidence is satisfactory as it is detecting our car accurately. As this is only a prototype 35 images are trained and it is enough for this particular environment irrespective of lightening conditions. But in practice, as discussed earlier more the training images the better will be the accuracy.

V. METHODOLOGY

Computer vision for frame retrieval: Computer vision focuses on making sense of what a machine sees. A computer vision system inputs an image and outputs task-specific knowledge, such as object labels and coordinates. With the help of CV libraries Image from the surveillance cameras are captured. After the collection of static images, it is then sent for further processing. Image processing mainly involves the filtration and transformation. The algorithm takes static size images so it is mandatory to transform image in particular width and height.

Figure 6

A frame retrieval from real parking lot video [12]

Image Transformation: The static images which are acquired from the camera are then requires transformation to be able to feed in the YOLO model. For that a particular blob size is fixed.

A **Binary Large Object** (**BLOB**) is a collection of binary data stored as a single entity in a database management system. Blobs are typically images, audio or other multimedia objects, though sometimes binary executable code is stored as a blob [13].

The information of blob sizes is shown below [14].

1. 320 x 320 (Smallest size blob and faster processing but low accuracy)
2. 416 x 416 (Medium size blob and average processing speed and moderate accuracy)
3. 608 x 608 (Largest size blob and slow processing but higher accuracy)

Detection: The transformed image is then fed in YOLO algorithm. After, YOLO will give multiple [x, y, w, h] coordinates of the objects (car) that is detected in particular image. Where x and y are the centers of bounding box of object, w and h are width and height of bounding box of object.

Comparison: Static parking slots points are already available in a file. We are comparing the static points to the points given by YOLO algorithm. If the points are inside the parking points then car is present and the space is occupied otherwise no car their in-parking slot and space is free.

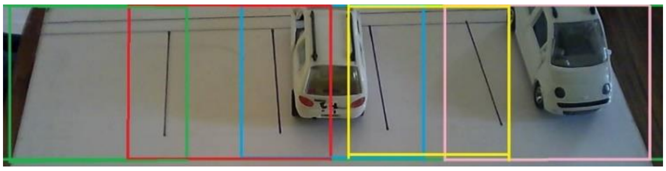

Figure 7: Static points plot visualization.

In the above figure the visualization of static points are plotted which will be used and compared with the points which yolo will give. For example the first slot have green box so maximum area is considered where car can be parked in that spot. According to it the points are noted down. Further the output of yolo will give object coordinates and this coordinates will be checked with the noted coordinates if yolo coordinates lies in the plotted coordinates then the car is present in that particular spot. In this example first parking spot. Similarly, all the slots are noted and checked in algorithm.

VI. RESULT

For, considering deployment scenario we have included real parking slot video. And in that every car is detected and when parking structure changes our algorithm updates GUI accordingly.

Figure 8. Real-time GUI Display of Deployment video

Figure 9 is the prototype image final output. It is totally working at every spot and updates GUI as you change cars. The video to this prototype is available here [15].

Figure 9

Table – 2

Method	mAP	FPS
Faster R-CNN	51.9	12
Fast YOLO	52.7	155
SSD 300	41.2	46
SSD 512	46.5	19
YOLO v2	48.1	40
YOLO v3	55.3	35

The above table shows different methods for object detection and shows mean average precision and frames per second to detect a particular object.[16]

V. CONCLUSION

We developed an approach for parking slots detection, which uses Computer vision and neural networks object detection techniques that classify parking space occupancy more accurate and faster. In practice each camera is able to cover a large portion of a parking lot by monitoring simultaneously about 50-100 parking spaces.

In order to validate our approach, we have taken various video samples of parking lots in different conditions. These videos we used as our dataset and this dataset are used in our system for checking the accuracy of system and also processing time to detect the slots. Moreover, we compare the presented approach with Haar and other classifiers and we found our system is more accurate and faster and the accuracy is far better.

As traditional parking lots uses sensor based detection techniques which require number of sensors. Initial cost and maintenance of those sensors becomes quite costly. As we have used camera based approach which will nullify the initial cost of sensor based systems. Also the maintenance is remarkably low. The following system will work in low light conditions or infrared vision cameras as well. The proposed method is also robust to non- standard parking behaviors, such as cars occupying neighbor parking slot.

VI. REFERENCES

[1] M. Y. I. Idris , Y.Y.Leng ,"car park system :review of smart parking system and its technology",Malaysia,Information technology journal 8(2):101-113,2009,ISSN 1812

[2] Giuseppe Amato,"Car Parking Occupancy Detection Using Smart Camera Networks and Deep Learning",978-1-5090-0679-3/16/$31.00 ©2016 IEEE

[3] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, ”You Only Look Once: Unified, Real-Time Object Detection”, University of Washington, Allen Institute for AI, Facebook AI Research, PDF: https://arxiv.org/pdf/1506.02640.pdf

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9):1627–1645, 2010

[5] Joseph Redmon, Santosh Divvala,"You Only Look Once:Unified, Real-Time Object Detection",University of Washington,Allen Institute for AI, Facebook AI Research

[6] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015

[7] Julien Nyambal,Richard Klein,"Automated Parking Space Detection Using Convolutional Neural Networks",University of the Witwatersrand Johannesburg, South Africa,978-1-5386-2313-8/17/$31.00 ©2017 IEEE

[8] jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

[9] Rachel Huang, Jonathan Pedoeem ,YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers, 978-1-5386-5035-6/18/$31.00 ©2018 IEEE

[10] RaghavPrabhu/understanding-of-convolutional-neural- network-cnn-deep-learning-99760835f148

[11] labelImg -master tool - https://github.com/tzutalin/labelImg , Google Colab - https://colab.research.google.com/

[12] A.Khare,"Automatic ParkingManagement Video", https://www.youtube.com/watch?v=y1M5dNkvCJc 02.01.2020, 13:00

[13] Available: www-cse.ucsd.edu/classes/wi07/cse190/reports/ntrue.pdf

[14] Available: http://www.ini.rub.de/data/documents/tschentssenparking fbi2012

[15] Available: https://www.youtube.com/watch?v=UYo-hYBmZAo

[16] Available: https://pjreddie.com/darknet/yolo

Cite this article as :

Kaushal Shah, Shivang Rajbhoi, Nikhil Prasad, Charmi Patel, Roushan Raj, "Cost effective Parking System Using Computer Vision", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 6 Issue 2, pp. 241-246, March-April 2020. Available at doi : https://doi.org/10.32628/CSEIT206276

Journal URL : http://ijsrcseit.com/CSEIT206276