One Convolutional Layer Model For Parking Occupancy Detection

Soumia Goumiri
Systems Design Methods Laboratory (LMCS)
National School of Computer Science (ESI)
Algiers, Algeria
s_goumiri@esi.dz

Dalila Benboudjema
Systems Design Methods Laboratory (LMCS)
National School of Computer Science (ESI)
Algiers, Algeria
d_benboudjema@esi.dz

Wojciech Pieczynski
SAMOVAR, Telecom Sudparis
Institut Polytechnique de Paris
Evry, France,
Wojciech.Pieczynski@telecom-sudparis.eu

2021 IEEE International Smart Cities Conference (ISC2) | 978-1-6654-4919-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISC253183.2021.9562946

***Abstract*—Convolutional Neural Networks (CNN) have recently performed wonders in image recognition tasks. In this paper, we propose a new CNN model composed of one convolutional layer, which we called 1Conv. We apply 1Conv for the problem of parking space detection. We used the most popular datasets to evaluate the performance of our model that are: National Research Council Park (CNRPark), National Research Council Park Extension (CNRPark+EXT), and Parking Lot (PKLot). We compared the results with mAlexNet, a CNN model similar to 1Conv. The results show that our model outperforms mAlexNet in terms of accuracy, Area Under the Curve (AUC), and execution time. The better accuracy of 1Conv compared to mAlexNet was 99.06% against 90.71% using CNRPark dataset. Which means that our model outperforms mAlexNet by 9% in term of accuracy. Execution time of mAlexNet is double compared to 1Conv.**

***Index Terms*—Convolutional Neural Networks (CNN) model, deep learning, parking space detection**

I. Introduction

The demand for parking is increasing as cities become increasingly overcrowded. Drivers circulate for hours looking for parking. Thus, consomme fuel, pollute the air by CO_2 emission, beside the psychological impact on the drivers. Consequently, smart parking management has become a must in every city in the world.

The traditional method to collect information about parking spots is sensors, but this solution suffers from the problem of installation and maintenance cost [1]. Recently researchers are using cameras to collect parking information. This arises some new application for drivers such as tracking their vehicles using plate number. Crowd-sourcing is also another data sources of parking data collection. In which users provide information such as using their smartphone. All these amount of data need a lot of time to be processed. The appearance of deep learning methods such as Convolutional Neural Network has facilitate the process of large amount of data.

Convolutional Neural Network models are doing wonders in computer vision tasks, especially image classification. This competitive advantage has motivated us to propose a model that uses CNN for predicting parking availability. We are not proposing a complete architecture that collects images of parking spots, process them, and then provide the results for the drivers. But, a model that can be integrated into any architecture that used images of parking lot as an input and provide a binary classification of them. This architecture tells the driver whether the input image is a free place or an occupied one.

The remainder of this paper is organized as follow: in Section II, we present the state-of-art works. Section III describes our approach. We provide an experimental evaluation in Section IV. Finally, we conclude our paper in Section V.

II. Related work

Several solutions have been proposed to solve the problem of parking occupancy detection. In this section, we focus on those based on CNN models such as: AlexNet [2], VGGNet [3], R-CNN [4], and Faster R-CNN [5].

The authors of [1] have extended the method proposed in [6]. They proposed a model that is inspired by the well-known CNN model AlexNet. The proposed model was named mAlexNet for mini AlexNet. It is composed of three blocks of convolutional layers and max-pooling layers. The flattening is followed by one fully connected layer. Finally, a softmax function is applied to get the classification of the input image. The authors also have introduced a new dataset CNRPark+EXT. The proposed method has been tested on three datasets: CNRPark, CNRPark+EXT, and PKLot using, different splits of them.

A Faster Region-Based Convolutional Neural Networks (R-CNN), aiming to detect parking occupancy using images collected from nearby parking spaces, was proposed in [7]. The considered images were taken from four onboard cameras in vehicles. In this solution, the authors have considered quadrilateral shapes of parking spaces, unlike the usual methods that work with rectangular shapes. The labeling of the dataset was done manually for about 400 images. This solution is composed of three modules: *a features extractor backbone*, *a Region Proposal Network (RPN)*, and *a detection head*. However, the proposed method is slow because it is based on two stages approaches.

Another method using Faster R-CNN was proposed in [8]. Authors propose a parking occupancy detection and vehicles count system. The two subsystems run in real-time. In the first step, the system extracts convolution features by using Resnet50 as a backbone network. Then, the results are used as an input to an RPN. Finally, the outputs of the previous step are sent to a CNN to classify them. The final results are binary classification and vehicle type (car, trucks, van, bus, and rickshaw). The tests were run using 20000 epochs on the PKLot dataset. The accuracy of this system was 90%.

A smart parking detection using object detection and object classification was proposed in [9]. The proposed system is composed of two networks for data collection: a network of cameras and a network of LIDARs. LIDARs are sensors capable of sensing about seven to eight spots status. The system provides different services such as the location of the spot, parking payment, and the location of the user's vehicle. User location is done by tracking the vehicle capturing its license plate number. A CNN is applied to determine the occupancy of a spot and the result is passed to a web application.

In [10] a Fully Convolutional Networks (FCN) Visual Geometry Group (VGG-Net) was used to obtain parking spot occupancy. FCN-VGG Net is a real-time fully automated method. It uses object detection and road segmentation methods. Segmentation is done using FCN-VGG Net, while Object detection is ensured by Faster R-CNN. This approach was designed for outdoor parking. The results reached 83% for crowded traffic and up to 92% otherwise.

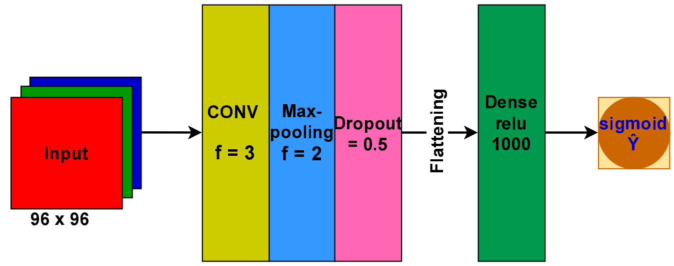

Fig. 1. Architecture of the proposed model.

III. Our Approach

We propose a new CNN model and apply it to the parking occupancy detection problem. The architecture of our model is illustrated in Fig. 1. The inputs of the model are RGB images of size 96x96. The architecture of the proposed model is composed of one convolutional layer. Followed by a max-pooling layer, then we drop out 50% of neurons. We employed the convolutional layer with filter of size 3x3 and channel size 16. We use a window of size 2x2 for pooling layer. The output of the previous block is flattened to become the input to the fully connected layers. One fully connected layer with units 1000 is applied. The last fully-connected layer represents the output of the model with one value belonging to {0,1}, where "0" means that the input image is a vacant place and "1" means that it is an occupied one. To all the layers is applied a ReLU activation function except for the output layer, for which we apply a sigmoid function, which is the most suitable for binary classification. First, we recall the definition of a ReLU is $h = max(0, a)$ where $a = Wx + b$ and W is the weight, b is the bias, and x is the input. The reasons behind the choice of ReLU are sparsity and a reduced likelihood of vanishing gradient [11]. The sigmoid function is defined by:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

IV. Evaluation Experiments

Tests were carried out on an NVIDIA GeForce RTX 2080 ti GPU card with 11 GB of memory. We note that we have access to only one GPU.

A. Datasets

To have a fair comparison the method proposed in [1], we used the same three datasets: CNRPark[1], CNRPark+EXT[2],

[1]http://claudiotest.isti.cnr.it/park-datasets/CNRPark
[2]http://claudiotest.isti.cnr.it/park-datasets/CNR-EXT/

TABLE I
DETAILS OF CNRPARK, CNRPARK+EXT, AND PKLOT DATASETS [1]

Datasets	Free	Occupied	Total
CNRPark	4181	8403	12584
CNRPark+EXT	65658	79307	144965
PKLot	337780	358119	695899

TABLE II
DETAILS OF THE SPLITS DATASETS USED IN THE EXPERIMENTS, WITH THE NUMBER OF FREE AND OCCUPIED SPACES

	Train		Test	
	Free	Occupied	Free	Occupied
CNRPark	3375	6691	807	1710
CNRPark+EXT	52450	63521	13208	15785
PKLot	286534	270182	71582	67598

and PKLot[3] [12]. In Table I we give the details of the three datasets used in the experiments, including the number of free, occupied and total spaces in each one of them.

In Table II we give the details of the splits datasets used in the experiments, with the number of free and occupied spaces.

B. Implementation details

To evaluate the performances of our model we used Keras for TensorFlow. This library provides a multitude of predefined algorithms, methods, and functions.

We used Adaptive Moment Estimation (Adam) optimizer to train our model. It is an algorithm for the gradient descent optimization technique. We have chosen this method because it is effective when dealing with problems involving a lot of data or parameters. In particular, it requires a reasonable deal of memory. Adam is used with the following parameters: learning rate=0.001, beta_1=0.9, beta_2=0.999, epsilon=1e-07, and amsgrad=False.

Since we are dealing with a classification problem and the output is binary, the most suitable loss function is Binary Cross Entropy (BCE). The BCE function is defined by:

$$-\frac{1}{output\ \ size}\sum_{1}^{output\ \ size} y_i.log(\hat{y_i}) + (1-y_i).log(1-\hat{y_i}) \tag{2}$$

[3]https://www.kaggle.com/blanderbuss/parking-lot-dataset

Where:

$output\ \ size$: number of output of the model;
y_i: is the real output equals to 1 or 0;
$\hat{y_i}$: is the predicted output by the model, $\hat{y}$ in $\{0,1\}$.

Metrics used to evaluate the performances of our model are, accuracy, Area Under the Curve (AUC), and execution time. The accuracy of a model is determined by the formula:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{3}$$

Where, *tp* and tn are true positives, and true negatives, *fp* and *fn* are false positives and false negatives. The objective of the training is to minimize the loss and maximize the accuracy and the AUC.

The AUC value measures the entire two-dimensional area under the entire Receiver Operating Characteristic curve, or ROC curve (by integral calculations) from (0,0) to (1,1). The ROC curve plots the rate of true positives as a function of the rate of false positives:

The True Positive Rate (TPR) is the equivalent of the recall. It is therefore defined as follows:

$$TPR = \frac{tp}{tp + fn} \tag{4}$$

The False Positive Rate (FPR) is defined as follows:

$$FPR = \frac{fn}{fn + tn} \tag{5}$$

We run our model 10 epochs with a batch size equal to 64. The data are split in two parts with 80% used for training and the remaining 20% used for testing. The splitting was done using *model_selection.train_test_split* [13] module of the python *scikit-learn* [14] tool which split arrays or matrices into random train and test subsets. During the experiments we have faced the issue of insufficient memory when working with big datasets. To faced this issue we load the datasets by batch of size 64, train the model on them, and then compute the average of the results accuracy and AUCs. We did the same process to test the model on the unseen images.

C. Results and Discussion

Experiment results are reported in Table III. We compared our method with mAlexNet [1]. We have used two metrics to compare the two methods, accuracy and AUC. In most cases, our model outperforms mAlexNet. 1Conv gave accuracy of 99.06% when tested on CNRPark against 90.71% for mAlexNet, and an AUC of 0.9989 against 0.9200 for mAlexNet. Our model improved the accuracy by almost

TABLE III
EXPERIMENT RESULTS

Models	**Dataset**	**Accuracy**	**AUC**	**Execution time**
1Conv	CNRPark	99.06%	0.9989	6.41
mAlexNet	CNRPark	90.71%	0.9200	12.63
1Conv	CNRPark+EXT	98.75%	0.9962	67.40
mAlexNet	CNRPark+EXT	97.71%	0.9967	145.00
1Conv	PKLot	99.84%	0.9996	237.55
mAlexNet	PKLot	98.07%	0.9967	522.14

9% when tested on CNRPark dataset. The second dataset we used is CNRPark+EXT, which is an extension of the previous dataset with more data. With this dataset, 1Conv has reached 98.75% accuracy and an AUC equal to 0.9962, whereas the accuracy of mAlexNet is lower by 1% i.e. 97.71% and its AUC is equal to 0.9967. The last dataset that we have used is PKLot, commonly used dataset in the literature. It is composed of more than 600 000 images of parking spots which allow us to evaluate our model with more precision. 1Conv has improved the accuracy of mAlexNet by 1% which gives a value of 99.04% for 1Conv and 98.07% for mAlexNet. AUC is 0.9996 and 0.9967 for 1Conv and mAlexNet respectively. The execution time of 1Conv is 6.41 minutes versus 12.63 minutes for mAlexNet using CNRPark dataset. With CNRPark+EXT dataset 1Conv took 67.40 minutes against 145.00 minutes for mAlexNet. 237.55 minutes is the execution time of 1Conv using PKLot dataset versus 522.14 minutes of mAlexNet. As we can see our model is doubly faster than mAlexNet.

From the above results, we can see that our model is more efficient than mAlexNet when tested with different datasets in terms of accuracy, AUC and execution time. Besides, 1Conv is composed of one convolutional layer against three for mAlexNet. This implies that our model requires less convolutions and pooling operations, and thus is faster, while requiring less storage on disk, and needs less memory. We note that the solution proposed in [9] is also composed of one convolutional layer, similarly to our model. This solution gives an accuracy of 99.51% which is less than the accuracy of 1Conv. We draw attention to the fact that 1Conv uses input of shape 96x96 contrary to shape 256x256 used in [9], which make 1Conv faster. the information provided by authors of [9] was not enough to reproduce the results and use their approach as a competitor of 1Conv.

V. CONCLUSION

In this paper, we have proposed a new CNN model, called 1Conv, composed of one convolutional layer. To evaluate 1Conv, we apply it to the problem of parking space detection. We have compared 1Conv with mAlexNet. The two models were tested on CNRPark, CNRPark+EXT, and PKLot datasets. The results show that our model gives, in most cases, better results than mAlexNet. 1Conv gives an accuracy of 99.06%, 98.75% and 99.84% against 90.71%, 97.71% and 98.07% of mAlexNet using CNRPark, CNRPark+EXT and PKLot respectively. The accuracy of 1Conv is always better than mAlexNet of all the tested datasets. For CNRPark, 1Conv outperforms mAlexNet by 9% in term of accuracy.

The execution time of 1Conv is 6.41, 67.40, and 237.55 minutes using CNRPark, CNRPark+EXT and PKLot respectively versus 12.63, 145.00, and 522.14 minutes for mAlexNet using the same datasets. As we can see our model is doubly faster than mAlexNet.

REFERENCES

[1] 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.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[4] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[5] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[6] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016, pp. 1212–1217.

[7] A. Zinelli, L. Musto, and F. Pizzati, "A deep-learning approach for parking slot detection on surround-view images," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 683–688.

[8] G. Khan, M. A. Farooq, Z. Tariq, and M. U. G. Khan, "Deep-learning based vehicle count and free parking slot detection system," in *2019 22nd International Multitopic Conference (INMIC)*. IEEE, 2019, pp. 1–7.

[9] H. Bura, N. Lin, N. Kumar, S. Malekar, S. Nagaraj, and K. Liu, "An edge based smart parking solution using camera networks and deep learning," in *2018 IEEE International Conference on Cognitive Computing (ICCC)*. IEEE, 2018, pp. 17–24.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[11] R. Biswas and S. Roy, "Botnet traffic identification using neural networks," *Multimedia Tools and Applications*, pp. 1–25, 2021.

[12] P. R. De 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.
[13] sklearn.model_selection.train_test_split. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
[14] scikitlearn: Machine learning in python. [Online]. Available: https://scikit-learn.org/stable/