PARKING LOTS CLASSIFICATION USING CNN

Assit.Lec Rasha Amer Kadhim Al- Khalissi
MS.c.. Computer science
College of Agriculture
Animal Production Department
Diyala University
Email : rashaakm@agriculture.uodiyala.edu.iq

***Abstract*—** The issue of finding an accessible parking spot moves toward becoming apparent because of the increasing the number owned cars . This paper manages the check of empty parking spots, by utilizing by using computer vision by using a binary classification system and Convolutional Neural Network that is equipped for deciding whether a parking spot is possessed or not. A benchmark database

Convolutional Neural Networks (ConvNets or CNNs) are a part of Neural Networks that have been exceptionally compelling in areas of image recognition and classification. ConvNets have been effective in recognizing faces, objects and traffic signs robots and self driving cars .[1]

Keywords*— *Convolutional Neural Networks; ConvNets ; CNN; classification; image recognition ; Layer; ReLU; Pooling; Backpropagation; LeNet ; AlexNet ; ZF Net ;GoogLeNet; VGGNet; ResNets; DenseNet; Caffe; Deeplearning4j; TensorFlow; Torch; Keras; MXNET. Deep Learning,parking lots,empty spot, binary classification.

1-INTRODUCTION

Finding a parking spot, especially in a city or during busy hours, can be a tedious process. According to a study by Donald Shoup at UCLA,' people in Westwood Village (with a population around 50,000) drive over 30 extra kilometers each year to find vacant parking spaces' (Shoup 2006)[15]. On average, people spend 8 minutes finding a spot (Shoup 2008)[15]. It would be not only more convenient, but also friendly to the environment, if drivers had update information on where he can find a nearest empty parking spots.I used Convolutional Neural Networks to build my model

2-RELATEDWORK

1. (Funck et al., 2004), how it work

It work by create an average image by using a diverse images of the empty parking area, under different conditions.

The system only rating the occupancy of the whole parking area and the average error rate of 10 %.

2. (True, 2007) : - It used a labeled Regions-Of-Interest (ROI). The system into two sections, in the initial segment a color histogram is created made for each parking spot and is then classified using utilizing either k- Nearest Neighbour (kNN) or a Support Vector Machine (SVM). The system only rating the occupancy of the whole parking area and the average error rate of 10 %., and 51 % error rate for feature detection [19].

3. (Bhaskar et al., 2011) : : it used a rectangle detection and Scale Invariant Feature Transform (SIFT). It using a classifier that based on threshold, they achieved an accuracy of 96.9 %, but the system depends on the lines of the parking space and the parking spaces should be a rectangular to get a good result[3].

4. (Huang and You, 2016) :

   using 3D point clouds . It work by segment unwanted information, buildings, ground and curb and use three Orthogonal-Views as input to a CNN . Using the method they achieved an accuracy of 83.8 %.[8]

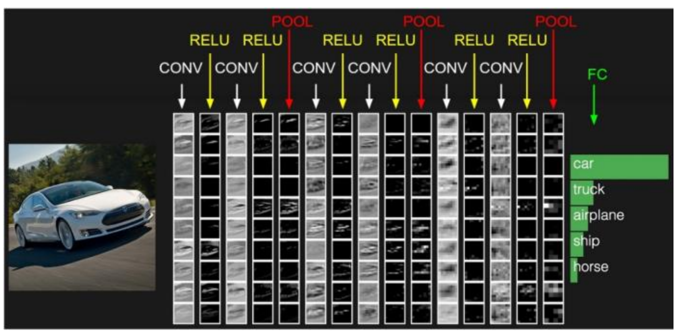
I will get away from the feature based approaches that used in most of the work that I described it above. I will do a binary classification system using a CNN instead(figure1), because CNN have been some of the most powerful innovations in the field of computer vision.

class 1

class 0

figure1

The first year that neural nets grew when Alex Krizhevsky used them in year's ImageNet competition dropping the classification error record from 26% to 15% and he win the competition that was in 2012 ,since then, a lot of companies have been using deep learning at their services. Such us Facebook uses neural nets for their automatic tagging algorithms, Google for their photo search, Amazon for their product recommendations, Pinterest, and Instagram.[2] CNN have figured out how to sort pictures into classes far better than people in some cases (figure 2). It is easy to understand especially when we break them down into their essential parts.



(figure 2)

3 - How CNN worke

CNNs are active at processing data in the form of arrays, which makes it ideal for computer vision tasks (Lecun et al., 2015). CNNs are based on Multilayer Perceptrons (MLP), since these consist of fully connected layers, they do not scale well with image sizes. Interestingly a CNN tries to exploit the spatially neighborhood relationship in pictures, by stacking the component maps and just interfacing every neuron to a little locale of the information volume, this is also called the receptive field of the convolutional layer. For each feature map, the weight and bias will be shared, this is possible by assuming that a feature which is useful to compute at one position, is also useful to compute at another spatial position.[21].

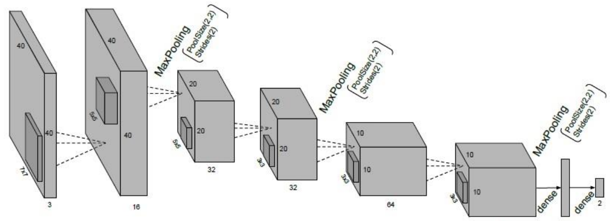


figure 3 CNN steps

The conv layer receives the raw image data. It uses its filters to recognize the features that it look important . Each filter produces a 2 dimensional array of dot products . this arrays called an activation map. The activation maps for all the filters in one layer are put together into a 3 dimensional activation volume. This volume will be the output of the conv layer. After the conv layer and ReLU the data passes through a max pool layer This cycle (Conv layer, ReLU layer, then Maxpool layer) happens several times, depending on the size of your CNN. The results from the last conv fed into a fully connected layers. Finally, the output of this last fully connected layer goes through a softmax layer which converts the fully connected layer's output into a set of class probabilities, one foreach class.[21].. (figure 4)

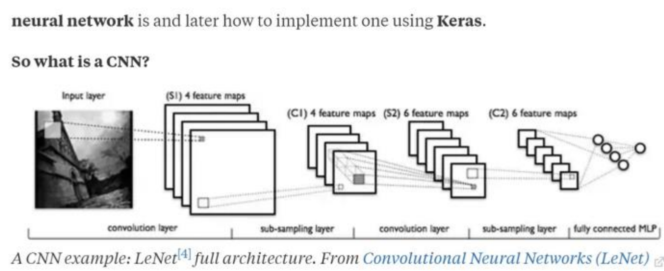


figure 4(CNN Architectures)

In the first layers a CNN detects simple features, such as edges, then corners. In the later layers, the network starts to learn more complex features, which it is seem t to the human eye. Activation function CNNs are developed of neurons, these have learnable weights and biases and can be communicated by the linear function:

$$y = w .x +b$$

w is the weight, x the input and b is the bias.

A-Convolutional Layer :- this is the most important part. It takes an image as an input and produces a smaller image where each pixel are a results from a mathematical convolution(process of a filter) with neighboring pixels also called kernel .In one convolution layer, the network will apply multiply filters [10] .

A CNN comprises of multiple convolutional layers. These layers scan over the image, using a small, M * M pixel feature detector called a filter, and mark the regions of the image that align most closely with that filter. It does this by sliding a M*M window over the image and computing the dot product between that window's pixel values and the filter's pixel values. A CNN normally consist of several convolutional layers, an activation function, pooling layers and lastly the classification layer, which is normally afully connected Neural Network [10].

"To calculate the match of a feature to a patch of the image, simply multiply each pixel in the feature by the value of the corresponding pixel in the image. Then add up the answers and divide by the total number of pixels in the feature(figure 3). If both pixels are white (a value of 1) then 1 * 1 = 1. If both are black, then (-1) * (-1) = 1. Either way, every matching pixel results in a 1. Similarly, any mismatch is a -1. If all the pixels in a feature match, then adding them up and dividing by the total number of pixels gives a 1. Similarly, if none of the pixels in a feature match the image patch, then the answer is a -1 .The next step is to repeat the convolution process in its entirety for each of the other features. The result is a set of filtered images, one for each of our filters."as [10]

4- CNN ARCHITECTURES

For CNN architecture I used ResNets (2015) . ResNets feed the output of two successive convolutional layer AND also by pass the input to the next layers.

"It features special skip connections and a heavy use of batch normalization. The architecture is also missing fully connected layers at the end of the network. The reader is also referred to Kaiming's presentation (video, slides), and some recent experiments that reproduce these networks in Torch.

ResNets are currently by far state of the art Convolutional Neural Network models and are the default choice for using ConvNets in practice (as of May 10, 2016). In particular, also see more recent developments that tweak the original architecture from Kaiming He et al. Identity Mappings in Deep Residual Networks (published March 2016)."

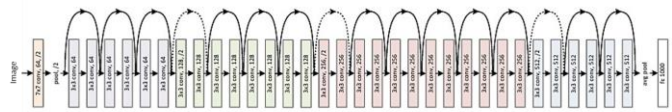
- developed by Kaiming . was the winner of ILSVRC 2015.

In a traditional network the activation at a layer is known as follows:

Y=f(x)

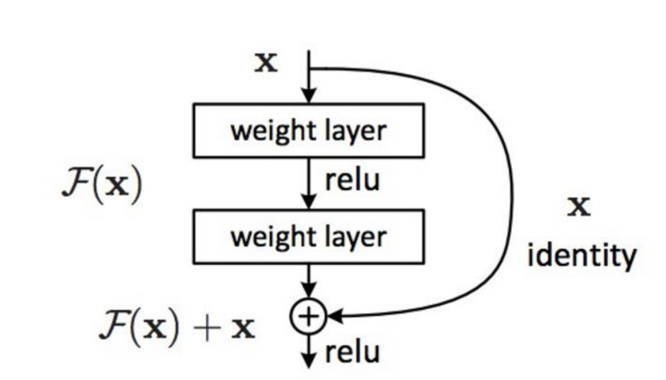
- Where f(x) is our convolution, matrix multiplication. At each layer the ResNet implements(figure 5):

Y=f(x)+ x



(figure 5) ResNets

It allows the slope to pass backwards directly. By amass these layers, the gradient could skip over all the intermediate layers and reach the bottom without being minimize (figure 6).



(figure 6) ResNets

5- TECHNOLOGIES USED

A. Anaconda 2 , Python 2.7 version.
B. I used Pillow library for processing the images
C. Tensorflow : An open-source software library for Machine Intelligence.

it's supported by Google. Creator: Google Brain Team
Interface: Python, C/C++

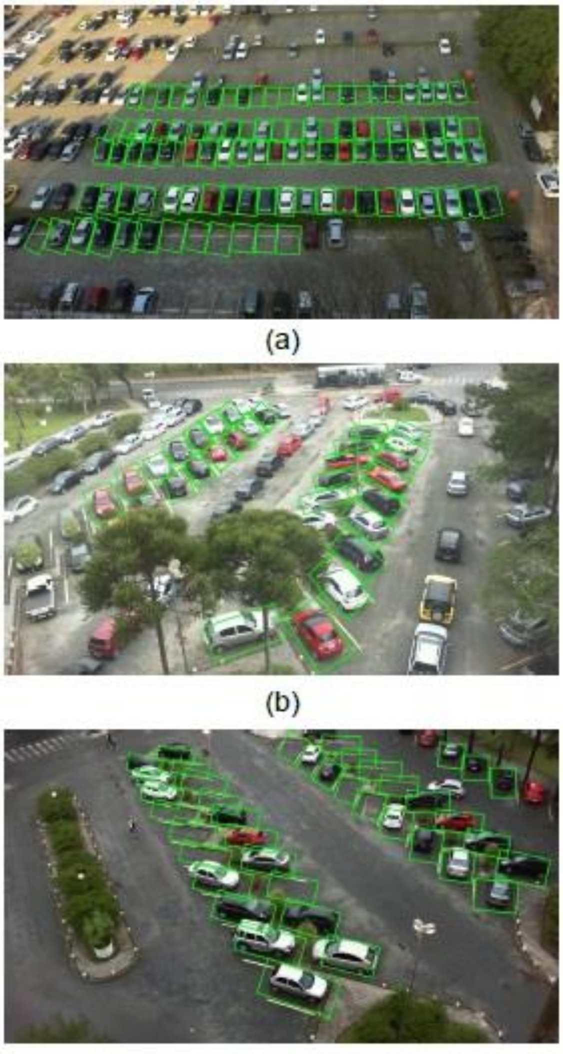
D. Tflearn : a deep learning software library featuring a higher-level API for TensorFlow. It is a high-level neural networks library, written in Python and capable of running on top of either TensorFlow [18]

6- DATASET

I use ("PKLot - A Robust Dataset for Parking Lot Classication")[4] :-This database consists of 12.417 images of the three parking areas, captured at a resolution of 1280x720 px. In total there is 695.900 images of parking spaces captured throughout the day and in three weather conditions, sunny, rainy and cloudy(figure 7).

In addition I used pictures that tacked from one of troy university's parking lots(figure 8) . I used a raspberry pi 2 camera to take the images .

PKLot - A Robust Dataset for Parking Lot Classication 'It contains 12,417 images of three parking lots and 695,899 segmented parking spaces in these lots. Two of the parking lots are in the Federal University of Parana (UFPR) and the third is in Pontifical Catholic University of Parana (PUCPR) resulting in three sets of data (UFPR04, UFPR05, PUC).'[4]



three parking areas

segmented parking spaces

(figure 7) PKLot - A Robust Dataset

(figure 8) troy university's parking lots

7- TRAINING THE DATA

For the training data I used The segmented images (figure 9) into over 600,000 sub images of individual parking spaces in size (32,32). Each space is hand labeled as Empty or occupied .I divided the training data in to 65% training 35% for testing . I resized the images to (32,32) then I saved the path of the image in to txt file with the label for each image . After loading the txt file I convert the images in to numpy array shape (number of Images,32,32,3) by using TFlearn data_utils [18]

tflearn.data_utils.image_preloader (target_path, image_shape, mode='file', normalize=True, grayscale=False, categorical_labels=True,files_extension=None,filter_channel=False)

tflearn.data_utils.build_hdf5_image_dataset (target_path, image_shape, output_path='dataset.h5', mode='file', categorical_labels=True, normalize=True, grayscale=False, files_extension=None, chunks=False)

figure 9

It take more than 24 hour to training in a nvidia GPU processor and with using the server.

8- The result

I used (70819) for training data from one parking lot in different weather condition and (35024) for testing data (figure 11)
.

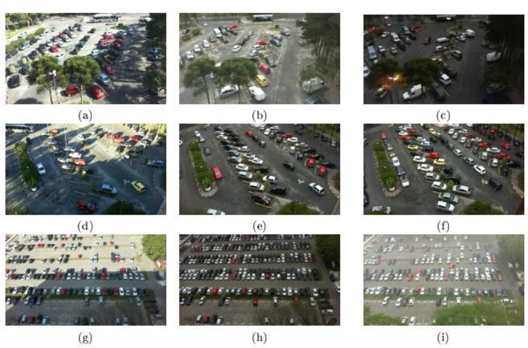
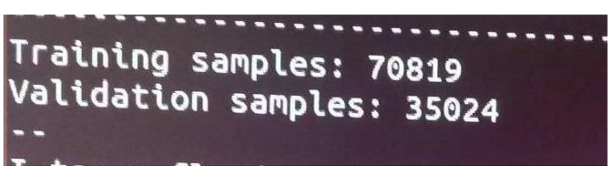


Figure 1: Images captured under different weather conditions: (a) sunny (b) overcast, and (c) rainy from UFPR04; (d) sunny (e) overcast, and (f) rainy from UFPR05; and (g) sunny (h) overcast, and (i) rainy from PUCPR

(figure 10)



(figure 11)

- I get about 100 accuracy for training and 99,90 for testing

- I get about (99,98)accuracy after training and testing
.

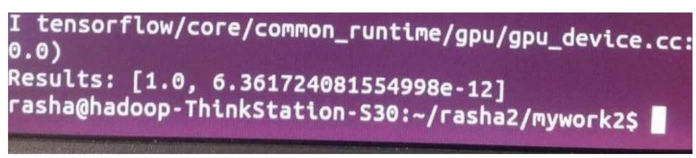
9 - Test the model passing one image

I succeeded to build my model . I test that model in my images that I collected . I cropped the parking spot from the parking lot pictures . Then rotated the cropped image and resizes them (32,32) using pillow then pass it to the model for classification .Then I convert the image in to numpy array be shape (1,32,32,3) to fit the model . After loading the saved model I run this code (figure 12):-

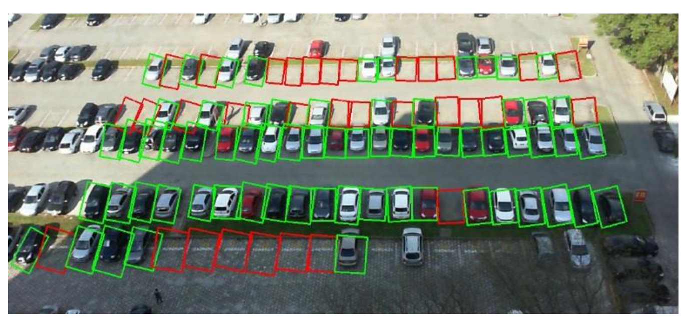
model.load('/home/rasha/rasha2/mywork2/parking_lots.tflearn')

model.predict (X)

X is np_array for the input image



(figure 12)



(figure 13)

10- Future work

The PKLot database used in this work, does not provide images at the night time . So we can improved my model by used some images in the night.to solve this problem we can lit the parking lots by street light.

Other improvements to the system could involve automatic detection of the parking spaces, as this would ease the process of installing the system at a new location.

One method to do this, could be by assuming all parking spaces are bound by two lines and are parallel to each other. Or by crap each spot of parking lot .

11- Conclusion

Convolutional Neural Networks give the best execution in image recognition problems and even beat people in specific cases .

The using of CNN has shown improvement in performance which is better than previous efforts.

it shown high accuracy when introduced to new parking spaces, with the lowest accuracy achieved being 95.45 % and the highest 99.98 %.

Besides this the system have shown to perform well under different illuminations, as the results when training the system on all of the training data I get (99.98 %) .

REFERENCES

1. Ahrnbom, M., Astrom, K., and Nilsson, M. (2016). Fast classification of empty and occupied parking spaces using integral channel features. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.
2. Baroffio, L., Bondi, L., Cesana, M., Redondi, A. E., and Tagliasacchi, M. (2015). A visual sensor network for parking lot occupancy detection in smart cities. In Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, pages 745–750.
3. Bhaskar, H., Werghi, N., and Al-Mansoori, S. (2011). Rectangular empty parking space detection using siftbased classification. In VISAPP, pages 214–220.
4. De Almeida, P. R. L., Oliveira, L. S., Britto, A. S.,Silva, E. J., and Koerich, A. L. (2015). PKLot-A robust dataset for parking lot classification. Expert Systems with Applications, 42(11):4937–4949.
5. Funck, S., Mohler, N., and Oertel, W. (2004). Determining Car-Park Occupancy from Single Images. Intelligent Vehicles Symposium, 2004 IEEE, pages 325–328.
6. Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In Gordon, G. J. and Dunson, D. B., editors, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11), volume 15, pages 315–323.
7. Journal of Machine Learning Research – Workshop and Conference Proceedings. Huang, C. C. and Vu, H. T. (2015). A multi-layer discriminative framework for parking space detection. In 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), pages 1–6.
8. Huang, J. and You, S. (2016). Vehicle detection in urban point clouds with orthogonal-view convolutional neural network. In 2016 IEEE International Conference on Image Processing (ICIP), pages 2593–2597.
9. Klosowski, M., Wojcikowski, M., and Czyzewski, A. (2015). Vision-based parking lot occupancy evaluationsystem using 2D separable discrete wavelet transform. Bull. Polish Acad. Sci. Tech. Sci., 63(3):569–573.
10. Krizhevsky, A., Sutskever, I., and Hinton, G. E.(2012). ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst., pages 1–9.
11. Lecun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature, 521(1):436–444.
12. Masmoudi, I., Wali, A., Jamoussi, A., and Alimi, A. M. (2014). Vision based system for vacant parking lot detection: Vpld. In Proceedings of the 9th International Conference on Computer Vision Theory and Applications (VISIGRAPP 2014), pages 526–533.
13. Masmoudi, I., Wali, A., Jamoussi, A., and Alimi, M. A. (2016). Trajectory analysis for parking lot vacancy detection system. IET Intelligent Transport Systems, 10(7):461–468.
14. Russakovsky, O., Deng, J., Su, H., Krause, Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3):211–252.
15. Shoup, D. C. (2006). Cruising for parking. Transport Policy, 13(6):479–486.
16. Simard, P., Steinkraus, D., and Platt, J. C. (2003). Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. Proc. 7th Int. Conf. Doc.Anal. Recognit., pages 958–963.
17. Sukhinskiy, I. V., Nepovinnykh, E. A., and Radchenko,G. I. (2016). Developing a parking monitoring system based on the analysis of images from an outdoor surveillance camera. In 2016 39th International Convention on Information and Communication Technology,Electronics and Microelectronics (MIPRO),pages 1603–1607.
18. Tflearn Development Team (2017). Tflearn: A Python framework for fast computation of mathematical expressions.
19. True, N. (2007). Vacant parking space detection in static images. University of California, San Diego. Tschentscher, M., Koch, C., Konig, M., Salmen, J., and Schlipsing, M. (2015). Scalable real-time parking lot classification: An evaluation of image features and supervised learning algorithms. In 2015 InternationalJoint Conference on Neural Networks (IJCNN), pages1–8.
20. Valipour, S., Siam, M., Stroulia, E., and Jagersand, M.(2016). Parking stall vacancy indicator system basedon deep convolutional neural networks.
21. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenetclassification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.