Park Slot Detection: Park Slot Detection: A Deep Learning Approach

André Bargas
FGA - Faculdade do Gama
University of Brasilia
Brasilia, Brazil
120110237@unb.aluno.br

Miguel Pimentel
FGA - Faculdade do Gama
University of Brasilia
Brasilia, Brazil
140156143@unb.aluno.br

***Abstract*—This article aims to present an empirical approach related on how to detect parking space vacancy (occupied or vacant). For this, CRISP-DM was used as methodology, PKLot Dataset as dataset and a model based on CNN. As a result, it was possible to obtain a 99.8 per cent of accuracy. Using the results found, for future work, a case study can be made applying this model associated with iOT concepts in a parking lot, with the objective of creating a low-cost parking detection system.**

***Index Terms*—CNN, PKLot dataset, parking lot detection, AI,**

I. Introduction

The Internet of Things(iOT) provided a new technology paradigm envisioned as a global network of machines and devices capable of interacting with each other [1]. In that context, the term Smart cities, also associated with iOT, makes reference to cities that use technology in a extreme level in a way to help better management of resources and life quality for the population such as [2]:

- Data gathering;
- Water and electricity distribution systems;
- Crime detection;
- Social services;
- Sustainability;
- Transportation;

In this scenario, parking is one of the main problems in big cities urban mobility. Those problems could be cause by different reasons such as:

- Insufficient parking spaces;
- Non effective use of parking spaces;
- Looking for a parking available space for huge periods of time;
- Bad signalization for allowed or not parking spaces;
- Bad use of priority parking spaces;
- People using cars to go to over crowded places;

Thus, this article aims to propose a Artificial Intelligence (AI) system for detecting parking lot vacancies (occupied or vacant) using Convulational Neural Network (CNN) and the PKLot dataset. As result, we obtained 99,8 per cent of prediction accuracy. In future works, those results could be used as base to create a low cost parking lot detection system.

II. Methodology

The Cross Industry Standard Process for Data Mining (CRISP-DM) is a methodology developed for data mining. Since its conception in 2006, the methodology has become increasingly used by data mining specialists to solve problems of this nature [3].

The CRISP-DM defines the data mining process to handle with massive amount of data, the process cycle is fragmented into six stages. The figure 1 shows the CRISP-DM, the arrows represent stages and their interactions into CRISP-DM cycle [3].

As presented in figure 1, the CRISP-DM provides its structure based on sequential cycles. Based on these attributes, CRIPS-DM enables continuous improvement. Those assumption conforms with the ultimate research goal: Detect parking lot vacancy using CNN. This work approached, the following CRIPS-DM stages: Business Understanding; Data Understanding and Data Preparation [3].

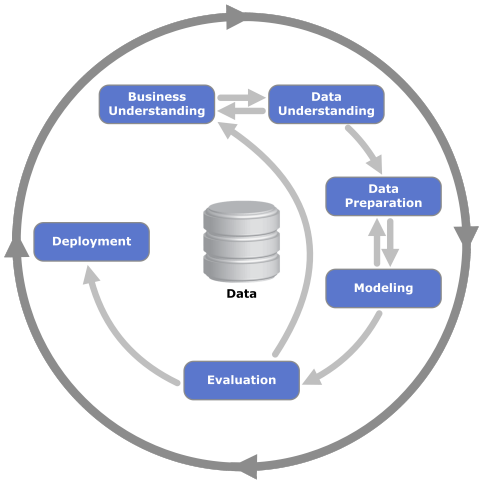

Fig. 1. CRISP-DM Methodology Cycle - Source: Colin Shearer

A. CRIPS-DM Phases

- **Business Understanding**: It can be understood as the initial phase of the data mining process, in this step the identification and understanding of the problem occurs.

Thus, it is necessary to search for information and details about the organization and problem [3].

- **Data Understanding**: Data Understanding is the stage to understand about the data content related to a specific problem. It is necessary to organize, inspect and describe the maximum data available, in order to identify which data is relevant to solve the problem formulated in the previous stage, Business Understanding [3].
- **Data Preparation**: At this stage, data begins to be manipulated. The data analysis format is defined and adjustments of technical issues are made. In practice, it means the data set's attributes are defined for study [3].
- **Modeling**: In this fourth stage, data mining techniques are defined in order to obtain the best result for the problem. In other words, it means that the best strategy for the problem is defined [3].
- **Evaluation**: In this stage the obtained results are studied and validated according to the initial problem [3].
- **Deployment**: Here, knowledge obtained in previous steps is applied to solve the initial question. In this step, the central purpose is to create the most applicable solution for the problem [3].

III. Dataset

The PKLot dataset was used to achieve the research goals of this work [4]. This dataset was created with the objective of being applied in researches, it contains more than 12,417 parking images and 659,899 images of parking spaces separated from each other [4].

Those images were captured at a 5-minute interval for a month using a low-cost high-definition camera. In this way, the images were captured in different whether conditions (overcast, sunny, and rainy periods).

The images were organised into three subsets named UFPR04, UFPR05 and PUCPR. UFPR04 and UFPR05 contain images of different views of the same parking lot captured from the 4th and 5th floors of the UFPR building. The last dataset contains images captured from the 10th floor of the administration building of the PUCPR [4].

Fig. 2. PKLot dataset Non-fragmented Images - Source: De Almeida, Paulo RL

The image below, present the image resources obtained:

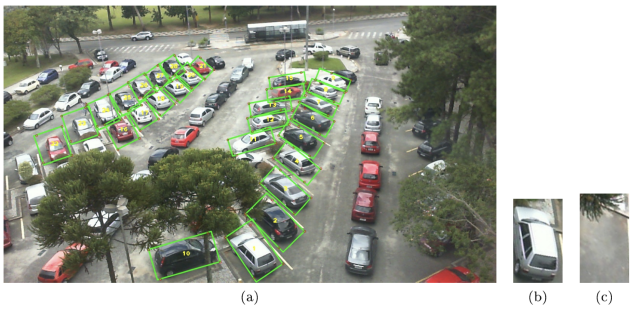

Fig. 3. PKLot dataset segmented Images - Source: Bargas, A. Pimentel, M.

In addition, to facilitate data manipulation, for each parking lot image was created an Extensible Markup Language (XML) file containing the position and situation (vacant or occupied) of each parking space [4]. The individual parking spaces were extracted from each parking lot image using the information available in the corresponding XML file.

IV. Data Preparation

The parking lot fragmented images have different dimensions, since then it requires perform manipulations to transform them to the same size. Resizing would cause quality losses, and consequently impacting the training phase.

Thus, using the highest height and width of the pklot dataset as a base, it was create a frame with 110x58 pixels dimension. Therefore, a black padding was added to the images that have a height and width less than the base value.

As a result, images were created as shown below.

Fig. 4. Example of manipulated Image - Source: De Almeida, Paulo RL

In addition, during the normalization of the images, a data frame was created, with all relevant information from

the PKLot dataset, that is, its relative path, name of the image (According to PKLot Dataset), the date of capture of the images, climatic condition and vacancy of the vacancy (occupied or vacant).

To perform the manipulation of images was the following tools:

- Open CV: Open Source Computer Vision Library is a multi-platform library, totally free to academic and commercial use, for the development of applications in computer vision area, just follow the BSD Intel license model [6]. In this work, it was used to create the padding as mentioned before.
- Pandas: One of the most used tools in Python, Pandas is known to be a fast, powerful, flexible and easy to use open source data analysis and manipulation tool [7]. In this paper, this tool was used to create the data frame and export it as csv to model phase.

V. Modeling

. The architecture developed for the solution is based on blocks composed of layers of convolution and max-pooling, followed by layers with dense networks.

Due to the fact that the data we are processing are images, the convolution and max pooling blocks are 2-dimensional. For the adaptation of this two-dimensional data to dense networks, a flatten layer is also used, with the objective of creating a linear representation for the data resulting from the processing. The model representation can be seen in the Figure 5.

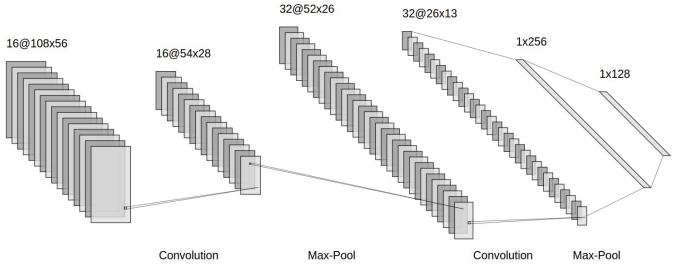

Fig. 5. Purposed Model - Source: Bargas, A. Pimentel, M.

To develop this model it was used the following tools:

- Keras: One of the most used tools in Machine Learning, Pandas is known as a consistent and simple APIs. In other words, provides interfaces to create machine learning models [8].
- Numpy: It is a package for the Python language, is known to supports multidimensional arrays and arrays, having a wide collection of mathematical functions to work with these structure [9].

The model implementation view can be in the Figure 6.

VI. Evaluation

For the evaluation phase, we use tools to obtain metrics and visualize the results. The tools can be described as the following :

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 108, 56, 16)	448
max_pooling2d (MaxPooling2D)	(None, 54, 28, 16)	0
conv2d_1 (Conv2D)	(None, 52, 26, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 26, 13, 32)	0
flatten (Flatten)	(None, 10816)	0
dense (Dense)	(None, 128)	1384576
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 1)	129
activation (Activation)	(None, 1)	0

Total params: 1,389,793
Trainable params: 1,389,793
Non-trainable params: 0

Fig. 6. Model Summary - Source: Bargas, A. Pimentel, M.

- SKLearn: The main purpose of this tool is to provide a simple light weight machine learning framework for predictive analysis and data classification. It also include utensils for model evaluation and metrics generation [11].
- Matplotlib: It is a package for the Python language, is known to supports multidimensional arrays and arrays, having a wide collection of mathematical functions to work with these structure [12].

As result, using the PKlot dataset as validation, it was obtained accuracy of 0.9985 and data loss of 0.0082, a very high rate. In conclusion sections we are going to discuss more about it.

In this perspective, a ROC curve (Receiver Operating Characteristic Curve) is a graphical representation that illustrates the performance of a binary classifier system as its threshold of discrimination varies. Applying the validation data set as base, it was obtained the following ROC curve for the purposed model. It is possible to check that the model performed almost without false positives. The figure bellow presents it:

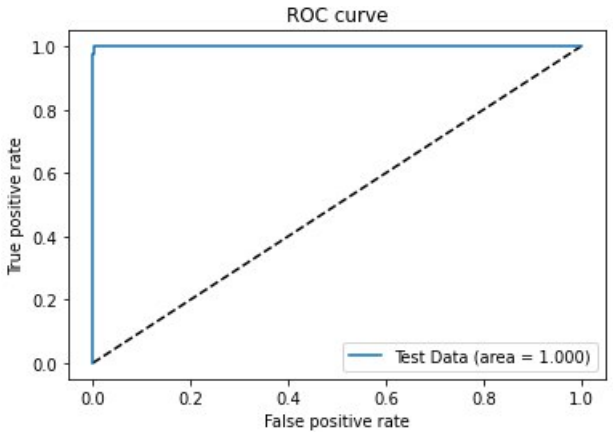

Fig. 7. ROC Curve - Source: Bargas, A. Pimentel, M.

VII. Conclusions

From the set of data preparation steps and development of the model, it was possible to obtain a very high accuracy. In

addition, a loss rate was also very low, which indicates that the model has a lot of effectiveness in classify parking slot vacancy.

However, the ROC curve is a fairly high ranking rate and, it presents a slightly different behavior compared to most machine learning models. Accuracy tests were performed using the PKLot data set, which can somehow indicate some addictive behavior of the data, and even a limitation of the results found.

Thus, for future work it is fundamental the purposed model as base the model and perform the same study using other data sets to ensure that this model can be applied in different contexts.

ACKNOWLEDGMENT

This article was developed along the discipline of special topics in Software Engineering: Autonomous Cars. Addressing some subjects covered during the 2020/1 period.

All the steps covered throughout this work, that is, the manipulation of the PKLot dataset and application of the deep learning model, are available in detail via jupyter notebook in the repository of this work [10].

REFERENCES

[1] Lee, In, and Kyoochun Lee. "The Internet of Things (IoT): Applications, investments, and challenges for enterprises." Business Horizons 58.4 (2015): 431-440.

[2] Chourabi, Hafedh, et al. "Understanding smart cities: An integrative framework." 2012 45th Hawaii international conference on system sciences. IEEE, 2012.

[3] Shearer, Colin. "The CRISP-DM Model: The New Blueprint for Data Mining." Journal of Data Warehousing 5 , no. 4 (2000).

[4] De Almeida, Paulo RL, et al. "PKLot–A robust dataset for parking lot classification." Expert Systems with Applications 42.11 (2015): 4937-4949.

[5] Murray, Naila, and Florent Perronnin. "Generalized max pooling." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[6] Bradsky, G. R.; Pisarevski, V.; Bouguet, J. Learning OpenCV: Computer Vision with the OpenCV Library. Springer, 2006.

[7] Murray, N. Florent P. "Generalized max pooling." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[8] Hollet, F. and others, 2015. Keras. Available at: https://github.com/fchollet/keras.

[9] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 0.1038/s41586-020-2649-2.

[10] Bargas, A. Pimentel, M. Keras. Available at: https://github.com/PKLOT-UNB/notebooks

[11] Cournapeau, David. Scikit Learn : Machine Learning in Python. Available at: https://scikit-learn.org/

[12] Hunter, D. John. Matplotlib : Visualization with Python. Available at: https://matplotlib.org/