Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Automatic recognition of Colombian car license plates using convolutional neural networks and Chars74k database

Automatic recognition of Colombian car license plates using convolutional neural networks and Chars74k database

D E Arroyo-Pérez[1], O I Álvarez-Canchila[1], A Patiño-Saucedo[2], H Rostro González[2], and A Patiño-Vanegas[1]

[1] Facultad de Ciencias Básicas, Universidad Tecnológica de Bolívar, Cartagena de Indias, Colombia

[2] Departamento de Ingeniería Electrónica, Universidad de Guanajuato, Salamanca, México

E-mail: didierarroyo2507@gmail.com, apatino@utb.edu.co

Abstract. A methodology for the automatic recognition of Colombian car license plates using convolutional neural networks is proposed. One of the biggest challenges when using onvolutional neural network is the demand for large amounts of samples for training. In this work, we show that if we do not have enough images of vehicle license plates to carry out the training, we can complement it with databases of letters and numbers that are not extracted from cars. The network was trained with the Chars74k database and images of characters extracted from plates of Colombian automobiles. The Chars74k contains approximately 74000 images of all the letters of the Spanish alphabet and all digits from 0 to 9. From chars74k database we have chosen 33849, because the Colombian plates have only uppercase letters and digits. Only 3549 (about 10% of the total) images of characters extracted manually from plates of Colombian automobiles were added. At the input of the convolutional neural network, 70% of the images were used for training, 20% for validation and 10% for testing and the resulting validation accuracy was above 99%. By making a preliminary test on Colombian plates never before used in training, a percentage of correctly recognized plates above 98% was achieved.

1. Introduction

The ability to automatically detect and recognize plates is a key tool used in many tasks. For example, in the identification of stolen vehicles or those that have committed a traffic offense, in the control of traffic or access to private places. The detection and recognition of plates is a challenge, since not all images captured with a camera will be of good quality due mainly to lighting conditions, deterioration of the plates and environmental conditions. Despite the large number of plate recognition systems available online, many do not work effectively everywhere, because each country has its own plate format [1–3]. Many automatic plate recognition systems exist with different degrees of precision and speed [4]. Currently, the systems that use the deep learning technique stand out, since the extraction of features is not done manually, but they are automatically extracted. The strongest deep learning methods involve convolutional neural network (CNN). CNN is a special type of multilayer perceptron trained in supervised mode using a gradient descent backpropagation learning algorithm that allows automated feature extraction. CNN has proven to achieve cutting-edge results in tasks such as optical character recognition, generic object recognition, real-time face detection, voice recognition, plate recognition, etc [5]. The biggest challenges of CNN are its high computational cost and the demand for large numbers of samples for training. However, the high computational cost does not currently represent a major problem since there are cloud computing services, where the user has the availability on demand of computer system resources, especially data storage and computing power, without direct active management by the user.

On the other hand, despite the large number of databases currently available, finding one that can be used for our training is a real challenge. In our case, the database must show many of the possible conditions (lighting, deterioration, environmental, etc) in which the plate that we want to recognize can be found. The motivation of this work is the recognition of Colombian car license plates, to control access to parking lots and generate automatic payment tickets. Colombian plates recognition has been carried out by Calderon, *et al.* [6]. They built their own database, perform a segmentation of each character on the plate and each binary image is encoded into a vector; which is used as input to a conventional neural network. In our recognition system, each character on the plate is also segmented, but its image goes directly without any treatment, into a convolutional neural network for recognition. Our network was trained with the Chars74k database [7] and images of characters extracted from plates of Colombian automobiles. Initially, we have trained the network with 33849 images chosen from the Chars74k database. Later, we did the training adding only 3549 (about 10% of the total) images of characters extracted manually from plates of Colombian automobiles. The network was trained in Keras, a Python package for deep learning. The CNN training was conducted in Google Colab, a free Jupyter Notebook environment that runs completely in the cloud. The results show an improvement in recognition when we add images of characters extracted from Colombian plates.

2. Convolutional neural network

Convolutional neural networks are the state of the art for different task in image processing [8], and are loosely inspired by observations of biological vision [9], where different aspects of the image are represented in a hierarchical way from low to high level representations. A typical convolutional neural network is a cascade of sequential operations, as follows.

2.1. Convolution operator

The basic operation of a CNN is a two-dimensional (2D) convolution between the input image and a set of filters or kernels. The weights of the filter are updated in every iteration using backpropagation [10], a generalization of the gradient descent algorithm for multiple layers. The resulting images are called feature maps.

2.2. Non-linear activation function

A nonlinear activation function is used to improve the performance of the training algorithm, by rectifying the values of the feature maps so they remain positive. This activation function is called Relu. Other typical activation functions are the sigmoid and hyperbolic tangent [9].

2.3. Pooling operator

The most widely used pooling operation is max-pooling [9], which consists of selecting the most active pixel value in a region. Its purpose is to reduce the dimensions of the matrices and achieve spatial invariance.

One conv-maxpooling-relu sequence constitutes a single convolutional layer and multiple convolutional layers are stacked to represent increasingly more high-level information: the first layers typically learn general information such as shapes or colors, and the last layers learn detailed information. Finally, a classifier is used on the feature maps of the last convolutional layer. Typically, a fully connected neural network with the softmax activation function is used.

The model used in this work is trained from the construction of a convolutional neural network with the characteristics mentioned previously. In each iteration, the values of the convolutional filters that extract the characteristics are adjusted and in this way the error decreases until a desired classification percentage is achieved.

3. Training data

In the experiment we restrict the problem to recognizing only typical Colombian license plates such as taxis and private cars, since they are the ones that normally enter the parking lots of shopping centers. A valid Colombian license plate of a car consists of three letters and three numbers separated by a space. We consider in this work, only the recognition of Colombian car plates that have been captured by a static camera, where the front part of the vehicle without movement, must be within the depth of field of the optical system of the camera. Furthermore, we consider only the plates whose characters can be recognized by means of a normal human eye. The Figure 1 shows some examples of images of cars with Colombian plates that serve as an entrance to our recognition system, with different degrees of difficulty to perform a good recognition.

Figure 1. Sample of Colombian car license plates.

It is well known that convolutional neural networks perform better when there is a large amount of data. But in addition, such data must be varied enough so that they show many of the possible conditions in which our object to be recognized can be found. For this reason, when there is not enough data to train a CNN with the aim of recognizing car license plates, we propose to train with public databases with many characters and add characters extracted from plates similar to the ones we want to recognize. In order to test our hypothesis, we have trained our network with the public database Chars74k and with images of characters extracted from plates of Colombian automobiles. The chosen characters of Chars74 provide us variability in the shape. On the other hand, the segmented characters of Colombian plates provide us variability in lighting, damage and environmental conditions.

Chars74k contains approximately 74000 characters of all types. Because the Colombian plates have only capital letters and digits, a data cleaning was made choosing 33849 images of capital letters of the Spanish alphabet and digits from 0 to 9. Each character has approximately 950 different images. On the other hand, from 600 Colombian license plates, 3549 characters were manually segmented to finally obtain a database with a total of 37398 images. All images were resized to 80x40 pixels with 3 RGB channels. Figure 2 shows the letter G as an example of the characters used for training. In Figure 2 shows images of letters extracted from Colombian plates (im 34 to im 63) and letters that belong to the Chars74k database (img017-00001 to img017-00014).

 1547 (2020) 012024 doi:10.1088/1742-6596/1547/1/012024

Figure 2. A sample of the letter "G" extracted from Colombian plates (im 34 to im 63) and Chars74k database (img017-00001 to img017-00014).

Groups of images were selected randomly so that 70% of the data was used for training, 20% for validation and 10% for the test. That is, 26179; 7479 and 3740 images respectively. The network worked with 200 epochs, a learning rate of 0.001 and a batch size of 16.

4. Methodology

This work was carried out in two stages: First a character segmentation is performed and then each character is recognized using a previously trained CNN.

4.1. Segmentation

This stage is performed to segment the characters from the license plate. The segmentation can be developed in different ways [11]. Figure 3 shows an example the results during a segmentation process that has given us good results: (a) grayscale conversion, (b) binarization, (c) Gaussian blur, (d) edge detection, (e) identification of closed contours, (f) segmentation of full plate.

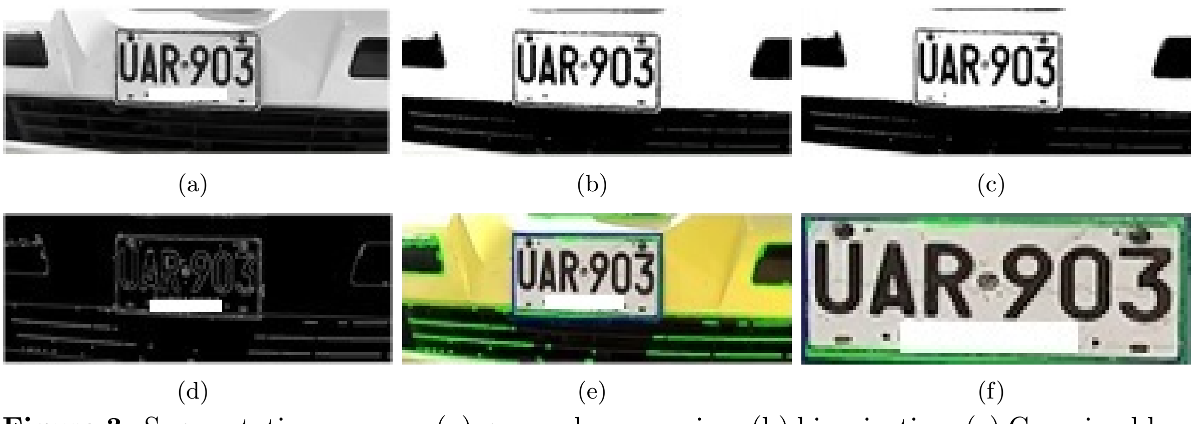


Figure 3. Segmentation process: (a) grayscale conversion, (b) binarization, (c) Gaussian blur, (d) edge detection, (e) identification of closed contours, (f) segmentation of the entire plate.

The final stage of segmentation example is shown in Figure 4, where the 6 characters are extracted from the entire segmented plate taking into account the fixed dimensions of the Colombian plates. Segmented characters serve as input to the convolutional neural network, previously trained, for their respective recognition.

Figure 4. Final result of the segmentation stage.

4.2. Structure of the convolutional neural network

The neural network designed for character recognition of Colombian car license plates consists of 3 convolution layers followed by a Relu activation function and a maxpooling layer [12]. It is followed by flatten, dense, dropout layers ending with a softmax activation layer with 36 outputs (26 letters and 10 numbers), which will indicate the classification percentages obtained for each class. The structure of this network is shown in the Table 1.

When we have an input image of 80x40 pixels a convolution is applied whose filter is 7x7 to generate 64 feature maps of 74x34. The 2x2 max pooling filter is applied to these maps and 37x17 feature maps are obtained. In the same way we do with the following convolutions whose filters are 3x3, to obtain 128 maps of 35x15. After applying the 2x2 max pooling filter, we obtain maps of 128 of 17x7. In the last convolution we obtain 256 maps of characteristics of 15x5, which after the filter Max pooling of 2x2 remain of 7x2. After the stage of extracting features for learning the network, the classification is made from the flatten, dropout and fully connected layer. It ends with the softmax function, which assigns probabilities to each of the characters to perform the respective classification.

Table 1. Structure of the designed convolutional neural network.

Layer (type)	Filters	Size/stride	Output shape
Conv2d_1 (conv2d)	64	7x7/1	(74 ; 34 ; 64)
Max_pooling2d_1 (maxpooling2)		2x2/1	(37 ; 17 ; 64)
Conv2d_2 (conv2d)	128	3x3/1	(35 ; 15 ; 128)
Max_pooling2d_2 (maxpooling2)		2x2/1	(17 ; 7 ; 128)
Conv2d_3 (conv2d)	256	3x3/1	(15 ; 5 ; 256)
Max_pooling2d_3 (maxpooling2)		2x2/1	(7 ; 2 ; 256)
Flatten_2(flatten)			(3584)
Dense_1 (dense)			(1024)
Dropout_1 (dropout)			(1024)
Activation_4 (activation)			(1024)
Dropout_2 (dropout)			(1024)
Dense_2 (dense)			(36)
Activation_5 (softmax)			(36)

The trained convolutional neural network receives each of the segmented characters and performs the classification process according to the training done previously. A result is obtained as shown in Figure 5 where in this case, each of the 6 characters is correctly classified. Figure 5 shows an example of the result of the recognition of the characters of a plate, by means of the previously trained convolutional neural network.

 IOP Publishing
1547 (2020) 012024 doi:10.1088/1742-6596/1547/1/012024

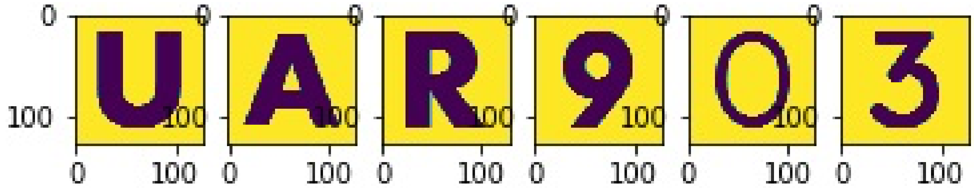
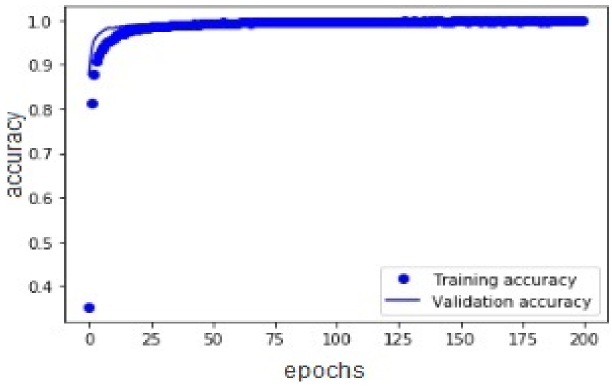


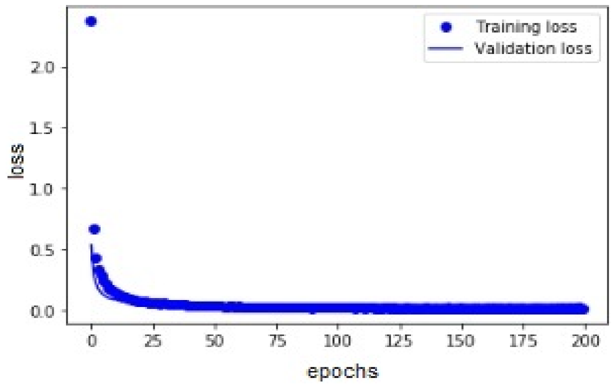
Figure 5. Result obtained by the convolutional neural network in the recognition of the characters.

5. Results

In the first stage, it has been possible to segment in each of the plates, its 6 characters with an effectiveness of 98%. In some cases the character is not properly segmented, and it can make its correct classification difficult. The CNN training was conducted in collaborator [13], a free Jupyter Notebook environment that runs completely in the cloud. Training the CNN only with characters of charsk74 obtained a percentage of classification of individual characters that did not exceed 90%. But, if the images of Colombian car license plates are included in the training, a percentage of 99.49% was achieved in the evaluation of the model. Of the 3740 images, he successfully identified 3721, presenting difficulty in 19 of them. Figure 6(a) and Figure 6(b) shows the progress in training in terms of accuracy and loss respectively during 200 epocs.



(a)



(b)

Figure 6. Training progress during the 200 epocs. (a) Accuracy. (b) Loss.

Finally, the total evaluation of the system for the total recognition of the 6 characters of a license plate, from the image of the license plate in the car, was carried out with 60 Colombian license plates. A classification percentage of 98% was achieved. This result is similar and sometimes improved compared to other works carried out by [6, 14, 15]. In some cases where the segmentation of each of the characters on the plates is done correctly, the fault is due to the CNN training process. In cases where none of the 6 characters are recognized, the fault is due to the segmentation process of the entire board. In other cases where only a few characters are recognized, the failure may be due to the training of the network, the bad segmentation of the character on the board, or its deterioration.

6. Conclusions

A Colombian car license plate recognition system has been proposed using the Chars74k character image database and a database created from character images extracted from Colombian license plates. The method consists in initially segmenting the 6 characters of the plate that the user wants to identify and then they are fed to be recognized by a CNN.

An accuracy percentage above 98% was obtained. In this work we have shown that if we do not have enough images of vehicle license plates to train a CNN, we can complement it with databases of images of letters and numbers that are not extracted from cars. A better recognition percentage can be obtained either by improving the segmentation process and/or by using more sophisticated CNN architecture. This work constitutes an advance in the technology of automatic license plate recognition systems, given that there is not much published research for the case of Colombian license plates using CNN.

References

[1] Shivakumara P, Tang D, Asadzadehkaljahi M, Lu T, Pal U and Hossein Anisi M 2018 *CAAI Transactions on Intelligence Technology* **3(3)** 169
[2] Halim S, Zulkifli M and Zulkipli M 2019 *Journal of Physics: Conference Series* **1358(012084)** 1
[3] Astawa I, Bawa I, Asri S and Kariati N 2019 *International Journal of Scientific and Technology Research* **8(11)** 481
[4] Du S, Ibrahim M, Shehata M and Badawy W 2013 *IEEE Transactions on Circuits and Systems for Video Technology* **23(2)** 311
[5] Lecun Y, Bengio Y and Hinton G 2015 *Nature* **521(7553)** 436
[6] Calderon J, Vargas J and Perez-Ruiz A 2017 License plate recognition for colombian private vehicles based on an embedded system using the zedboard *IEEE Colombian Conference on Robotics and Automation (CCRA)* (Bogotá: IEEE)
[7] De Campos T E, Babu B R, Varma M *et al.* 2009 Character recognition in natural images *Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP)* vol 2 (Lisbon: INSTICC Press)
[8] Mallat S 2016 *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **374(2065)** 1
[9] Schmidhuber J 2015 *Neural Networks* **61** 85
[10] Rumelhart D E, Durbin R, Golden R and Chauvin Y 1995 Backpropagation: The basic theory *Developments in Connectionist Theory. Backpropagation: Theory, Architectures, and Applications* (New Jersey: Lawrence Erlbaum Associates Inc.) pp 1–34
[11] Gonzalez R C and Woods R E 2017 *Digital Image Processing* (New York: Pearson)
[12] Patino-Saucedo A, Rostro-Gonzalez H and Conradt J 2018 Tropical fruits classification using an alexnet-type convolutional neural network and image augmentation *International Conference on Neural Information Processing* (Cambodia: Springer) pp 371–379
[13] Bisong E 2019 Google colaboratory *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners* (Berkeley: Apress) pp 59–64
[14] Wu S, Zhai W and Cao Y 2019 *IET Image Processing* **13(14)** 2744
[15] Kessentini Y, Besbes M, Ammar S and Chabbouh A 2019 *Expert Systems with Applications* **136** 159