



HAL
open science

Using Synthetic Images for Deep Learning Recognition Process on Automatic License Plate Recognition

Saulo Cardoso Barreto, Jorge Albuquerque Lambert, Flávio de Barros Vidal

► **To cite this version:**

Saulo Cardoso Barreto, Jorge Albuquerque Lambert, Flávio de Barros Vidal. Using Synthetic Images for Deep Learning Recognition Process on Automatic License Plate Recognition. 11th Mexican Conference on Pattern Recognition (MCPR 2019), Jun 2019, Queretaro, Mexico. pp.115-126, 10.1007/978-3-030-21077-9_11 . hal-04058300

HAL Id: hal-04058300

<https://hal.science/hal-04058300>

Submitted on 4 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Using synthetic images for deep learning recognition process on Automatic License Plate Recognition

Saulo Cardoso Barreto^[0000-0002-7793-8414], Jorge Albuquerque Lambert^{*[0000-0002-5750-5689]} and Flávio de Barros Vidal^[0000-0002-6317-218X]

^{*}Department of Mechanical Engineering, Department of Computer Science
University of Brasília, Brasília 70910-900, Brazil
saulo_cardoso@outlook.com, lambert.jal@dpf.gov.br, fbvidal@unb.br

Abstract. The Automatic License Plate Recognition has been the subject of several studies, given its applicability in real world situations (e.g. toll collection, identification of vehicles in parking lots or even for safety issues in vehicle control that cross borders between countries). In this work, we propose an analysis of the influence to retraining a plate recognition model and a deep neural network for object detection, using synthetic plates image databases from the Brazilian licence plates. The proposed data set uses variations of rotation, size and noise to evaluate the robustness. Thus, the influence of the use of synthetic plates images on the accuracy of systems responsible for locating real plates, segmenting the characters and recognizing them was evaluated and in the tests performed there was an increase in accuracy (considering a system trained with real plates) of three stages: character segmentation, letter recognition and number recognition (2.54%, 1.09% and 2, 49% respectively). It stands out the accuracy of 62.47% (in the number recognition step) obtained by a neural network trained exclusively with synthetic data and tested on real plates.

Keywords: Deep Neural networks · License plate recognition · Synthetic images.

1 Introduction

The task of locating and identifying car cards in real time or in images that have already been captured has received increasing attention in Computer Vision research due to its importance and wide variety of applications such as: traffic accident investigation, image analysis in crime scenes, automated verification of parking tickets and surveillance of speed limits [10]. Automatic License Plate Recognition (ALPR) is defined as a system whose objective is to perform the tasks described above in order to obtain a satisfactory result in the final application: locate a vehicle license plate and recognize each character[14]. In according to Li et al. [14], an input image of an ALPR system is captured under the influence of several factors: camera resolution, vehicle and camera orientation, presence of light in the environment at the time of capture, camera shutter speed, and climatic conditions among others. The process for recognizing a license plate can be divided into three steps, as illustrated in the Figure 1: i) find the region of an image corresponding to the vehicle's plate ii) from the plate image locate and segment the characters and finally iii) recognize each letter and number.

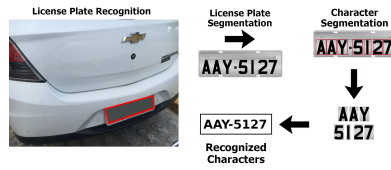


Fig. 1. ALPR system steps (the license plate was blurred due to privacy constraints).

Machine Learning techniques have been used for the three steps of recognizing a license plate. The greatest challenge for the ALPR and which prevents it from achieving results close to those of object recognition. For example, is the unavailability of a large-scale annotated database, given the difficulty in collecting and categorizing a data set with real license plates. In a situation where it is not possible to obtain a large scale database, the Data Augmentation method is used to create a synthetic database. Knowing that the license plates of the vehicles have many different standards, depending on the country in analysis, the project described in this paper is limited to the Brazilian standard adopted until 2018, which consists of three letters followed by four numbers (See a license plate sample in Figure 1).

The general objective of this work is to verify the ability to recognize real license plates from a trained neural network with a synthetic database, as well as to verify the accuracy of a previously trained network with real images, retrained with artificial plates (*transfer learning* and *fine-tuning*). Considering the difficulties of obtaining real databases with annotated vehicle license plates, the possibility of training a network and obtaining a high accuracy in the recognition of the characters, using only artificial plates, would be shown as a promising result in the area of Computer Vision and verify the influence on the accuracy of the system when performing a new training, starting from the initial weights of the previous training, but using artificial data as input.

2 Related Works

ALPR systems are not new in the industry and academy wherein many techniques were developed and improved along the last decades. However, even with a wide variety of techniques available, it is still a relevant research topic. Early works basically used image processing techniques to locate the vehicle's nameplate area and then recognize the data contained (characters and numbers) in the previously bounded area [4].

In recently years, many approaches were developed and used on ALPR systems. In current technical literature, many works can be found using any variation of the AI approach as describe in [19]. All these works (and many others available) using only real images of private license plate data sets, and most of them specific for a country, in the work developed in [6], the authors focused on Brazilian license plates and evaluated data augmentation techniques. Mainly because of the difficulty in achieving a large data set of real labeled images to perform the a good training of these models. Because of this issue, the analysis of the direct impact on the recognition process accuracy of license plates from synthetic images may contribute to the improvement of the entire recognition process, from training and validation to testing of the proposed model.

3 Artificial Neural Networks Models

3.1 Convolutional Neural Networks

LeCun [13] proposed in 1998 the use of convolutional neural networks (CNN) for image, voice, and time series recognition. What differentiates it from other types of neural networks is the use of the convolution operation instead of the multiplication of matrices in at least one of its layers. Two main features of a CNN stand out from other types of neural networks: local receptive fields and shared weights. **Local Receptive Fields:** Each pixel of an image is used as input of a neuron in a convolutional neural network [3]. Considering a fully connected neural network, neurons in one layer are connected to all neurons in the next layer. **Shared Weights:** In a neural network, each connection between neurons has an associate weight while in a convolutional neural network, all weights are shared and used throughout the whole image [12]. Thus, the same filter is applied for each $n \times n$ pixels, and a feature can be located in any region of the input image, thus conferring CNN's the properties of **translation and rotation invariance**. Considering that the weights matrix has parameters shared by all the neurons, not one for each, the number of parameters used by a neural network decreases substantially when using the convolutional architecture.

3.2 You Only Look Once (YOLO)

For applications that have the real-time requirement, the execution time of the various stages of the frameworks based on selection of regions of interest becomes a bottleneck of the strategy as stated by Zhao et al. [22] which also observed that systems that have only one step based on global regression directly map the pixels to bounding boxes, reducing the time taken for detection. The first object-recognition architecture to succeed in detection using only one step was proposed by Redmon et al. [16] in 2015, in which was not necessary to propose regions of interest, resulting in a detection up to 6-7 times faster than a Faster R-CNN network, with a detection time of 22 ms [9]. Therefore, such architecture is capable of processing videos in real time, but the disadvantage is a loss in accuracy.

4 Proposed Methodology

The flowcharts represented in Figures 2(a) and 2(b) illustrate the methodology used to verify the proposed approach and more detailed described in the next subsections.

4.1 Data sets

Artificial Dataset without variations: This data set (represented in Figure 3(a)) consists of license plates created using the OpenCV library [8]. The city and state name were omitted to avoid overfitting by preventing the neural network from extracting a feature from that part of the license which would not contribute to the use with license plate from different locations. The text font used was the mandatory as specified by the Brazilian traffic department. The colors and a gray gradient were also set so as to approach actual license plates. 902 artificial license plate have been created in this database, such amount is justified by the number of unique character sequences of the Diversified Real Data set used in the test step.

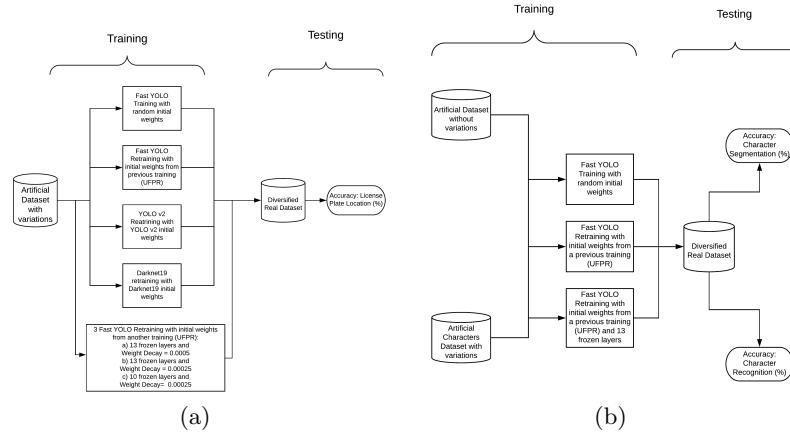


Fig. 2. (a) Training and tests stages of plate location architectures. (b) Training and tests stages of character segmentation and recognition architectures.

Artificial Data set with variations: In order to have a more complex data set, the SUN397 database [21], provided by the Massachusetts Institute of Technology, was used as the background image in which a plate was added with random and limited variations of rotation, scaling, Gaussian noise, brightness, perspective and sharpening using the Data Augmentation for Object Detection (YOLO) tool [18].

Diversified Real Data set: In order to have a database that was not used in any training cited in this work, the images available in [1] were used as reference for all the calculated accuracy. This choice is justified because it is a database with real license plates captured in different environments under varying circumstances, as illustrated in Figure 3(c). It is formed by 1126 images of which 902 are unique sequence of character. The annotation of this database had only the information of which characters were present in each plate, in this way, it was necessary to manually define the bounding box for each license plate.

Artificial Character Data set: To perform the retraining of the letter and number recognition networks, artificial databases were created for each group using the Data Augmentation for Object Detection (YOLO) tool [18]. In the training stage, 50 variations of each digit were used (totaling 500 images), and 20 variations of each letter (totaling 520 images). The figure 3(d) shows sample images of these databases.

4.2 Deep Neural Network Architectures

All the architectures described in this Subsection have been implemented using the neural network framework darknet [15], written in C language which presents optimization for execution in GPUs. The choice of the tool is justified by the fact that the Brazilian work used as reference [11] for the license plate recognition task was developed in this platform. Therefore, compatibility was maintained with the previously

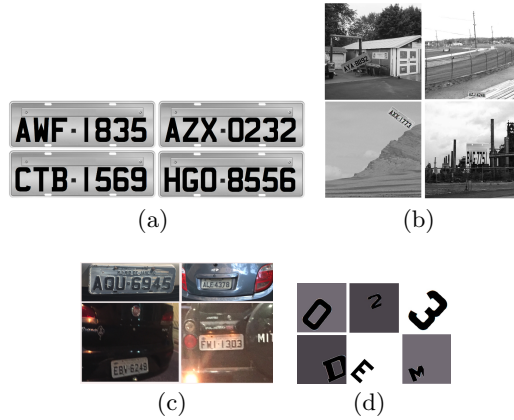


Fig. 3. (a) Artificial Dataset without variations. (b) Artificial Dataset with variations. (c) Diversified Real Database. (d) Artificial Character Data set.

generated training files. In addition, the YOLO architecture and its variations used in this work were also created in this framework. The fixed parameters for the training were: *momentum* = 0.9 and learning rate equal to 0.001, 0.0001 and 0.00001 up to the iterations 100, 25000 and 35000 respectively. The ratio used for the training and test data set was 75:25 % as done in [7].

Fast YOLO for Plate Location: Considering that the analysis of the training of a neural network with synthetic database involves several training with variations of parameters, it was decided to use the Fast YOLO because it presents a smaller number of layers, and consequently, a shorter convergence time (for training) and detection (for the test). Thus, two parameters were analyzed in order to observe their influence on the network accuracy: i) Freezing of 13 or 10 layers and ii) Decay of Weights equal to 0.0005 or 0.00025 as described in Table 1. Such an architecture reduces accuracy, but it is able to recognize objects in sampled images at a rate of 155 frames per second [9]. The YOLOv2 (with its 32 layers) and Darknet19 (25 layers) architectures were omitted due to space limitation but can be obtained in [2].

Table 1. Fast YOLO for Plate Location.

| Layer | Filters | Size | Input | Output |
|-------|-----------|------|------------------------|----------------|
| 0 | conv | 16 | 3 x 3/1 416 x 416 x 3 | 416 x 416 x 16 |
| 1 | max | 2 | 2 x 2/1 416 x 416 x 16 | 208 x 208 x 16 |
| 2 | conv | 32 | 3 x 3/1 208 x 208 x 16 | 208 x 208 x 32 |
| 3 | max | 2 | 2 x 2/2 208 x 208 x 32 | 104 x 104 x 32 |
| 4 | conv | 64 | 3 x 3/1 104 x 104 x 32 | 104 x 104 x 64 |
| 5 | max | 2 | 2 x 2/2 104 x 104 x 64 | 52 x 52 x 64 |
| 6 | conv | 128 | 3 x 3/1 52 x 52 x 64 | 52 x 52 x 128 |
| 7 | max | 2 | 2 x 2/2 52 x 52 x 128 | 26 x 26 x 128 |
| 8 | conv | 256 | 3 x 3/1 26 x 26 x 128 | 26 x 26 x 256 |
| 9 | max | 2 | 2 x 2/1 26 x 26 x 256 | 13 x 13 x 256 |
| 10 | conv | 512 | 3 x 3/1 13 x 13 x 256 | 13 x 13 x 512 |
| 11 | max | 2 | 2 x 2/1 13 x 13 x 512 | 13 x 13 x 512 |
| 12 | conv | 1024 | 3 x 3/1 13 x 13 x 512 | 13 x 13 x 1024 |
| 13 | conv | 1024 | 3 x 3/1 13 x 13 x 1024 | 13 x 13 x 1024 |
| 14 | conv | 30 | 1 x 1/1 13 x 13 x 1024 | 13 x 13 x 30 |
| 15 | detection | | | |

Table 2. Modified YOLO-VOC for Character Segmentation and Recognition.

| Layer | Filters | Size | Input | Output |
|-------|-----------|------|-----------------------|---------------|
| 1 | conv | 32 | 3 x 3/1 240 x 80 x 3 | 240 x 80 x 32 |
| 2 | max | | 2 x 2/1 240 x 80 x 32 | 120 x 40 x 32 |
| 3 | conv | 64 | 3 x 3/1 120 x 40 x 32 | 120 x 40 x 64 |
| 4 | max | | 2 x 2/2 120 x 40 x 64 | 60 x 20 x 64 |
| 5 | conv | 128 | 3 x 3/1 60 x 20 x 64 | 60 x 20 x 128 |
| 6 | conv | 64 | 1 x 1/1 60 x 20 x 128 | 60 x 20 x 64 |
| 7 | conv | 128 | 3 x 3/1 60 x 20 x 64 | 60 x 20 x 128 |
| 8 | max | | 2 x 2/2 60 x 20 x 128 | 30 x 10 x 128 |
| 9 | conv | 256 | 3 x 3/1 30 x 10 x 128 | 30 x 10 x 256 |
| 10 | conv | 128 | 1 x 1/1 30 x 10 x 256 | 30 x 10 x 128 |
| 11 | conv | 256 | 3 x 3/1 30 x 10 x 128 | 30 x 10 x 256 |
| 12 | conv | 512 | 3 x 3/1 30 x 10 x 256 | 30 x 10 x 512 |
| 13 | conv | 256 | 1 x 1/1 30 x 10 x 512 | 30 x 10 x 256 |
| 14 | conv | 512 | 3 x 3/1 30 x 10 x 256 | 30 x 10 x 512 |
| 15 | conv | 30 | 1 x 1/1 30 x 10 x 512 | 30 x 10 x 30 |
| 16 | detection | | | |

Modified YOLO-VOC for Character Segmentation and Recognition: As verified by Gonçalves et al. [5], the separation of the neural networks responsible for identifying the numbers and letters reduces the incorrect classifications. The size of the image of the network responsible for segmenting the characters (240 x 80) is justified by the ratio of the dimensions of the Brazilian car plates (3: 1). It is important to note that the segmented characters are sorted in descending order according to their coordinates on the horizontal axis, thus separating the first 3 that are letters, of the last 4 that are the numbers. The Table 2 represents the architecture used for character segmentation. For letter recognition the same architecture was used, adjusting only the size of the layers and the amount of filters. However for number recognition the first four layers were removed since the network performance would not be affected [11].

4.3 Accuracy Criteria

Considering the network responsible for locating the license plate in a image, if the Intersection over Union of the predicted bounding box is equal or greater to 70% ($IoU \geq 0.7$) when compared to the ground truth bounding box, the license plate defined by the neural network is considered accurate. The choice of this value is justified by the protocol developed by Li et al. in 2017 (using 0.7 for the anchor box definition) [14], who stated that this value would better evaluate the detection performance. For the character segmentation network, it is considered as a hit the segmentation of exact 7 characters, and for the character recognition the matching between the predicted and expected character.

5 Results and Discussion

In this Section, the results for the proposed methodology in Figures 2(a) and 2(b) will be presented. The relation between the terms are: **Epoch:** When a neural network received all the elements of the training set as input and updated the weights, it is said that an epoch was finalized; **Batch:** The amount of images used in a training step is defined by the batch size; **Iteration:** Number of steps performed with the defined batch.

5.1 Trained Networks to find a license plate in an image

Random Initial Weights vs. Initialized Weights: The Figure 4(a) illustrates the accuracy of neural networks in intermediate stages of their respective training, in which

it is verified that the final results are similar after 60 thousand iterations (a difference of 0.62% exists between the final values). It is also worth mentioning the decreasing behavior of the transfer learning carried out with initial non-random weights, because during the training its accuracy was reduced by 44.76%. It is also possible to verify the low values obtained by the training indicated in green in Figure 4(a) in which the highest value obtained was after 14 thousand interactions and was equal to 15.63% which represents 176 correctly located plates, given the 1126 images of database.

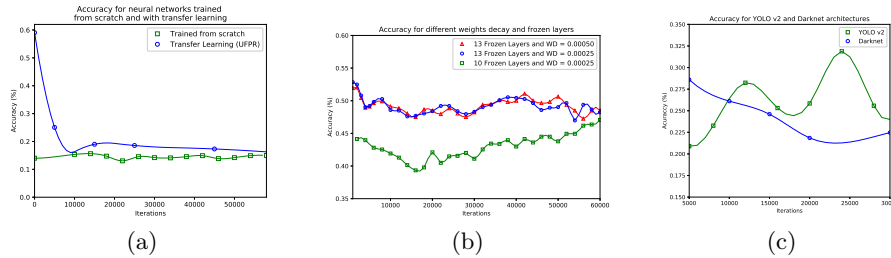


Fig. 4. (a) Comparison between two trainings: without transfer learning and with initialized weights from another training. (b) Influence of the freezing layers and the variation of the weight decay. (c) Comparison between two trainings: YOLO v2 and Darknet.

Different Weights Decay and Frozen Layers: Considering the three trainings performed in this stage (Figure 4(b)), the accuracy values for the two trainings with the same number of frozen layers were similar, since the average accuracy of the intermediate stages for $WeightDecay = 0.0005$ and $WeightDecay = 0.00025$ were equal to 49.51 % and 49.32 %, respectively, evidencing that for this case, the change in weight decay did not significantly influence the accuracy of the models. There is also a difference between the obtained values for the same value of Weight Decay (with a change in the layer freeze strategy), since the 13-layer frozen network located 557 of the 1126 license plates, while the 10-layer frozen network correctly located 474 vehicle license plates (difference of 7.37 %). The highest standard deviation was for the architecture with 13 frozen layers being equal to 1.46 % for the training with the decay of weights equal to 0,00025. When the network was analyzed with decay of 0.0005, the standard deviation was 1.25 %. Finally, the only network with 10 frozen layers had a standard deviation of 1.40 %.

Yolo v2 vs. Darknet: Comparing the results of both trainings performed with initial weights of neural networks trained to recognize the most varied objects, it is noted that the transfer learning performed with the YOLO v2 had an average accuracy of 2,23% greater than the Darknet architecture (26.26% and 24.03%, respectively) for the 30 thousand iterations. The accuracy of the analyzed networks begin to distance themselves from the 20,000th iteration, presenting a difference of approximately 10% when the accuracy of *YOLO* v2 reaches its maximum value and locates 359 of 1126

plates correctly (31.88%) as described in Figure 4(c), which is justified by the fact that Yolo v2 has the first 23 layers identical to Darknet-19 but has 7 additional layers, as described in [15].

5.2 Trained Networks to segment characters

Considering the neural networks trained to segment the characters, it is observed that only one of them exceeded the result obtained previously with the UFPR Training: the transfer learning performed with the UFPR initial weights made the accuracy increase by 2.58%, causing the number of correctly-located license plates to increase by 30 (going from 854 to 884). Both the YOLO training with 13 frozen layers (since the architecture of the network responsible for segmenting the characters is identical to the one responsible for locating a license plate) and training with random initial weights showed inferior performance: respectively 760 and 283 plates recognized, out of 1126 (67.50 and 25.13%).

Table 3. Accuracy for neural networks trained to segment characters.

| Training | Accuracy (%) |
|---|--------------|
| Fast YOLO (UFPR initial weights) | 78,42 |
| UFPR Training | 75,84 |
| Fast YOLO (UFPR initial weights and 13 frozen layers) | 67,50 |
| Fast YOLO (Random Initial Weights) | 25,13 |

Table 4. Accuracy for neural networks trained to recognize letters.

| Training | Accuracy (%) |
|---|--------------|
| Fast YOLO (UFPR initial weights) | 84,53 |
| Fast YOLO (UFPR initial weights and 13 frozen layers) | 83,71 |
| UFPR Training | 83,44 |
| Fast YOLO (Random initial weights) | 11,57 |

Table 5. Accuracy for neural networks trained to recognize digits.

| Training | Accuracy (%) |
|--|--------------|
| Fast YOLO (UFPR initial weights and 9 frozen layers) | 92,76% |
| Fast YOLO (UFPR initial weights) | 92,44% |
| UFPR Training | 90,27% |
| Fast YOLO (Random initial weights) | 62,47% |

5.3 Trained Networks to recognize characters

Letter recognition: This was the network that presented the worst performance for the scenario with random initial weights: only 307 letters out of 2652 were correctly recognized (11.57%). As for the improvements in initial UFPR training, both transfer learning and fine-tuning made the accuracy increase by 1.09 and 0.27%, respectively (Table 4).

Number recognition: In the stage of recognizing the numbers, the fine-tuning obtained a higher accuracy than the network that was submitted to *transfer learning*: the first recognized 10 more numbers than the second one 3278 in a set of 3536 numbers

- 92.76%. The freezing of the first 9 layers of the net (which has 12 layers) caused the accuracy to increase by 2.49 %. Thus, the network was able to recognize 88 additional characters in relation to the amount previously recognized by UFPR training (without any training with artificial databases). Considering all the trained networks with random initial weights, the person responsible for recognizing the numbers presented the best result, since 2209 numbers of actual plaques among the 3536 evaluated (62.47 %) were correctly classified by a neural network that did not receive numbers captured in real-world situations at any stage of your training, as described in Table 5.

6 Conclusions

Initially, by analyzing the performance of all trained neural networks with random initial weights and artificial database, our proposed approach was partially proved since 62 out of 100 numbers were recognized by a neural network trained exclusively with a synthetic database, proving that there is no need to use large annotated real databases for this task of a ALPR system. The character segmentation performed by the model initialized with random weights was able to correctly segment 25.13% characters of the real license plates. Considering that the training of this networks was done with a database without variations (Figure 2(b)), it did not extract features from license plates in different perspectives or angles, thus restricting its ability to correctly extract the expected 7 characters, nonetheless, a reasonable learning ability of the network with this result is noted. The second approach was fully confirmed by the results obtained for all the neural networks, but with different aspects since the influence in the segmentation (Table 3), letter and number recognition networks (Tables 4 and 5) is positive, that for all these steps there was an increase in accuracy caused by transfer learning. It should be noted the effect of applying the freezing layers strategy, since it is possible to verify the largest accuracy variation in this work: the transfer learning with UFPR initial weights and all free layers achieved an accuracy of 14,20% at the end of the iterations (Figure 4(a)), while the same architecture with 13 frozen layers and weight decay equal to 0.0005 achieved an average accuracy of 49.51% (Figure 4(b)) in the step of finding a license plate. Evidencing that the performed fine-tuning resulted in an increase of 35.31% in the model accuracy.

Finally, the results in the Figure 4(c) show that deep neural networks previously trained to recognize different objects can be used as a starting point for a specific training, knowing that the strategy has already been used to recognize plants [17], medical images [20] and Norwegian car plates [9]. The strategy made the maximum accuracy of the YOLO v2 architecture equal to 31.88% (Figure 4(c)) with only 24000 iterations, implying that features extracted from other training (edge information, color, position in the image, etc.) can be useful in developing a model to recognize a smaller number of classes. Analyzing the obtained results, it is verified that the use of synthetic databases for the license plate recognition task can improve the performance of some stages of the process. All presented results achieved improvements for the training step (less epochs and improved the accuracy) when synthetics labeled image are used.

References

1. Plates br - git hub. <https://bit.ly/2TEG2e5>, access on 10-11-2018

2. Yolo: Real-time object detection. <https://bit.ly/2fXFbXS>, acessado em: 06-08-2018
3. Bojarski, M., Choromanska, A., Choromanski, K., Firner, B., Jackel, L.D., Muller, U., Zieba, K.: Visualbackprop: visualizing cnns for autonomous driving. CoRR **abs/1611.05418** (2016), <http://arxiv.org/abs/1611.05418>
4. Du, S., Ibrahim, M., Shehata, M., Badawy, W.: Automatic license plate recognition (alpr): A state-of-the-art review. IEEE Transactions on Circuits and Systems for Video Technology **23**(2), 311–325 (Feb 2013). <https://doi.org/10.1109/TCSVT.2012.2203741>
5. Gonçalves, G.R., Menotti, D., Schwartz, W.R.: License plate recognition based on temporal redundancy. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). pp. 2577–2582 (Nov 2016). <https://doi.org/10.1109/ITSC.2016.7795970>
6. Gonçalves, G.R., Diniz, M.A., Laroca, R., Menotti, D., Schwartz, W.R.: Real-time automatic license plate recognition through deep multi-task networks. 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI) pp. 110–117 (2018)
7. Hamori, S., Kawai, M., Kume, T., Murakami, Y., Watanabe, C.: Ensemble Learning or Deep Learning? Application to Default Risk Analysis. Discussion Papers 1802, Graduate School of Economics, Kobe University (Jan 2018)
8. Itseez: Open source computer vision library. <https://github.com/itseez/opencv> (2015)
9. Jorgensen, H.: Automatic license plate recognition using deep learning techniques. <https://brage.bibsys.no/xmlui/handle/11250/2467209> (2017)
10. Kim, J., Kim, S., Lee, S., Lee, T., Lim, J.: License plate detection and recognition algorithm for vehicle black box. In: 2017 International Automatic Control Conference (CACS). pp. 1–6 (Nov 2017). <https://doi.org/10.1109/CACS.2017.8284273>
11. Laroca, R., Severo, E., Zanlorensi, L.A., Oliveira, L.S., Gonçalves, G.R., Schwartz, W.R., Menotti, D.: A robust real-time automatic license plate recognition based on the yolo detector. In: 2018 International Joint Conference on Neural Networks (IJCNN). pp. 1–10 (July 2018). <https://doi.org/10.1109/IJCNN.2018.8489629>
12. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (Dec 1989). <https://doi.org/10.1162/neco.1989.1.4.541>
13. LeCun, Y., Bengio, Y.: The handbook of brain theory and neural networks. chap. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258. MIT Press, Cambridge, MA, USA (1998), <http://dl.acm.org/citation.cfm?id=303568.303704>
14. Li, H., Wangy, P., Shen, C.: Towards end-to-end car license plates detection and recognition with deep neural networks. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS (Set 2017)
15. Redmon, J.: Darknet: Open source neural networks in c. <https://pjreddie.com/darknet/>, acessado em: 23-11-2018
16. Redmon, J., Divvala, S.K., Girshick, R.B., Farhadi, A.: You only look once: Unified, real-time object detection. CoRR **abs/1506.02640** (2015), <http://arxiv.org/abs/1506.02640>
17. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning deep convolutional networks for plant recognition. In: CLEF (2015)
18. shreeraman: Data augmentation for object detection(yolo). <https://github.com/srp-31/Data-Augmentation-for-Object-Detection-YOLO->
19. Silva, S.M., Jung, C.R.: Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). pp. 55–62 (Oct 2017)
20. Tajbakhsh, N., Shin, J.Y., Gurudu, S.R., Hurst, R.T., Kendall, C.B., Gotway, M.B., Liang, J.: Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Transactions on Medical Imaging **35**(5), 1299–1312 (May 2016)
21. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: Sun database: Large-scale scene recognition from abbey to zoo. IEEE Conference on Computer Vision and Pattern Recognition **abs/1703.06870** (2010)
22. Zhao, Z.Q., Zheng, P., tao Xu, S., Wu, X.: Object detection with deep learning: A review. CoRR **abs/1807.05511** (2018)