Methodology to Evaluate Artificial Intelligence Models to Recognize Peruvian License plates

Luis Angel Bastidas Cerazo[1], Carlos Alberto Sotelo López[2], Paola Chávez Ruiz[2], Miguel Angel Ontiveros Dueñas[3]

[1]*External Researcher, Electronic and Electric Engineering Faculty, National University of San Marcos, Lima, Peru.*
[2]*Professor, Electronic and Electric Engineering Faculty, National University of San Marcos, Lima, Peru.*
[3]*Professor, Faculty of Mechanical, Electrical and Electronic Engineering, National University San Luis Gonzaga, Ica, Peru.*
Email: luisbastidas.1993@gmail.com

Vehicles control is an important task in countries like Peru, where the number of vehicles in roads keep increasing and there are issues regarding car theft, Traffic regulations control, monitoring and enforcing, and others scenarios where it is important for officers to identify a vehicle. To assess this problem, using artificial intelligence models is an alternative. In this research, a methodology is presented to determine which model performs better for the task of recognizing license plates in image files obtained from uncontrolled environments. For this, five open-source models for object recognition are fine-tuned on a dataset of labelled images of Peruvian License plates. After this training, all models are evaluated following the proposed methodology, so metrics of performance are obtained with which to decide what model is better.

Keywords: Artificial Vision, Characters Recognition, Deep Learning, Object Detection, Peruvian License plates.

1. Introduction

The development of AI-based models has experienced rapid growth in recent years, enabling their application in multiple tasks, being one of them the task of computer vision.

In this paper, we discuss the results of the application of a methodology to evaluate artificial intelligence-based models fine-tuned to perform license plates detection in Peru.

Firstly, we cover the related work regarding license plate detection. Then, the Peruvian case is addressed, with the corresponding related word.

Afterwards, the methodology is described in a step-by-step basis, to be applied to assess the performance of five models fine-tuned using the Tensorflow Object Detection API to detect Peruvian Licenses Plate, using a self-implemented dataset of multiple photos of Peruvian cars with their license plates exposed.

The results are then presented and discussed. Lastly, the conclusions of the paper are presented, including the proposition of future works that could be carried on related to this matter.

2. Related work:

We covered the most recent related work regarding license plate detection in the World and Peru in specific. This review is done to identify if there is a methodology that can be chosen to assess the performance of the models trained for detecting and recognizing Peruvian license plates.

Firstly, we have the use of standard computer vision techniques applied over the pictures to locate the license plate. For his, edge detection, colour analysis, or special filters have been applied [1] [2].

The better accuracy for the detection of License plates were obtained by using Artificial Neural Networks and Deep Learning. For instance, the fine-tuning of models like AlexNet [3] and YOLO [4] [5] [6], gave better results, obtaining final models with accuracy values above 89%. These approaches, however, didn’t propose a methodology of evaluation. They fine-tuned a single model and measure the indicators of accuracy only.

Regarding the specific task of detecting and recognizing Peruvian License, the main approach to this was using traditional computer vision techniques combined with machine learning systems for the recognition phase of the task [7] [8] [9].

After the review of these works, we identify that there is not a common methodology that can allow a researcher o developer to select a model over others.

3. Methods and Methodology:

The methodology that was used is based on the evaluation of the following parameters per process:

For the training:

- Training time: it is the time taken from the beginning of the process of training until it finishes.
- Classification error at the end of the training process: it is the error produced when the model classifies wrongly an object.
- Localization error at the end of the training process: it is the deviation in the location of the detected element from the real location of the object to be detected.
- Regularization error: it refers to the error incurred in the model during the training process after applying regularization functions to avoid overfitting or underfitting.

For the license plate detection:

- Average precision (IoU=0.50:0.95): it indicates the proportion of the correct detections over all the objects that has been detected in an image (all True Positives and False Positives). It is calculated over an IoU (Intersection over Union) which ranges from 0.50 to 0.95. Ideally, in an image there should not be false positives, so the AP is 1, and it decreases if there are more false positives detected.

- Recall (IoU=0.50:0.95): it indicates the proportion of correct detections over all the real objects that should be detected (it means, all True positives and False Negatives). It is calculated over an IoU (Intersection over Union) which ranges from 0.50 to 0.95. Ideally, in an image there should not be false negatives, so the RE is 1, and it decreases if there are more false negatives.

For the license detection inference:

- False positives: it is the number of times that the model indicates an object to be a license plate when it is not.

- Detection rate: it is the number of license plates correctly detected over the total number of license numbers processed. Ideally, this rate should tend to one (100%).

- Minimum, maximum and average inference time: this indicates the minimum, maximum and average time consumed by the model to process the pictures and infer a valid license plate location.

All of these parameters must be estimated for each model using the same dataset and the same proportion of training data, validation data and test data.

After estimating the values for each of these parameters, a punctuation process based on rankings is conducted.

Considering there are N models under evaluation, for each parameter, all the models should be assigned a ranking position from 1 to N. This position will then be employed to assign points to the model based on the following equation:

$$P_{n,i} = N - R_{n,i}$$

Where:

Pn,i is the points assigned to the model "n" for the parameter "i".

N is the number of models under evaluation.

Rn,i is the ranking position of the model "n" according the estimated value obtained for the parameter "i".

For example, let's consider we are comparing five models (model 1, model 2, model 3, model 4, model 5) for the parameter training time, obtaining the parameters indicated in Table 1.

Table 1. Example of punctuation for the parameter training time

Model	Training Time	Ranking	Points
Model 1	20 seconds	3	3
Model 2	19 seconds	2	4
Model 3	40 seconds	5	1

Model 4	30 seconds	4	2
Model 5	10 seconds	1	5

Following this procedure, points must be assigned to all the models for each parameter as shown in Table 1. Then, the total points are summed up for each model, obtaining this way the final score for each model.

The model that gets more points following this methodology can be chosen as the model who gives the better general performance according to parameters evaluated.

With this, all models will get a score that can be useful to determine which performs better than the others reviewed. The methodology proposes a quick way to evaluate a set of trained or fined tuned models, but its results are only applicable to the models it has compared.

Also, some parameters can have a ponderation different for the others, which would depend on the use case for the resultant model.

4. Implementation

For the implementation of models to access the proposed methodology, five models were fine tuned to perform the object detection and recognition task of Peruvian license plates.

This task can be divided in two phases: The detection of the license plate(s) in the frame (location); and the recognition of the characters that indicate the number of the license plate.

These five models are the following, obtained from the Tensor Flow Detection Model Zoo [10]:

- EfficientDet D0 512x512
- EfficientDet D1 640x640
- SSD MobileNet V1 FPN 640x640
- SSD MobileNet V2 FPNLite 320x320
- SSD MobileNet V2 FPNLite 640x640

These models were fine-tuned using a script in python, employing the Tensorflow Object Detection API. This process was carried on in Google Colab[1].

The dataset employed for the fine-tuning is a dataset created by the Autor. It is made of a collection of 392 pictures containing a total of 406 license plates[2].

For the recognition process of the license numbers, the PaddleOCR [11] library was used. This is an open-source library to handle OCR tasks. With this library, we performed the task of recognizing each character in the plates located in the previous phase.

This phase of the process was carried on locally, in a PC with no dedicated graphic card, to simulate an environment with few computational resources.

[1] The script in Google Colab can be accessed in this link: https://colab.research.google.com/drive/178hMJ4hOm9QHkOi7hubeZ2R_uvBA7XpV?usp=sharing

[2] This dataset can be shared for research purposes contacting to the First Author Email

5. Results and discussion

In this section, we present the results obtained from the implementation described previously.

For the training (fine-tuning) process, the results are shown in Table II.

Table 2. Results of the training phase for the five models considered

Parameter	Based model				
	EfficientDet D0 512x512	EfficientDet D1 640x640	SSD MobileNet V1 FPN 640x640	SSD MobileNet V2 FPNLite 320x320	SSD MobileNet V2 FPNLite 640x640
Training time (seconds)	2 181.80	4 189.04	3 059.06	750.42	1 592.29
Classification error	0.089	0.077	0.037	0.024	0.027
Localization error	0.0012	0.0012	0.0132	0.0046	0.0047
Regularization error	0.0327	0.0345	0.8946	0.1365	0.1354

For the licence plate detection process, the results are shown in Table 3.

Table 3. Results of the detection phase for the five models considered

Parameter	Based model				
	EfficientDet D0 512x512	EfficientDet D1 640x640	SSD MobileNet V1 FPN 640x640	SSD MobileNet V2 FPNLite 320x320	SSD MobileNet V2 FPNLite 640x640
Average precision IoU=0.50:0.95	0.752	0.766	0.746	0.764	0.823
Recall IoU=0.50:0.95	0.786	0.796	0.800	0.802	0.853

Lastly, for the inference phase, the results for the five models considered are shown in Table 4.

Table 4. Results of the recognition phase for the five models considered

Parameter	Based model				
	EfficientDet D0 512x512	EfficientDet D1 640x640	SSD MobileNet V1 FPN 640x640	SSD MobileNet V2 FPNLite 320x320	SSD MobileNet V2 FPNLite 640x640
False positives	1	2	4	7	4
Detection ratio	0.9557	0.9778	0.9827	0.8744	0.9901
Minimum inference time	0.52 s	1.04 s	1.29 s	0.18 s	0.39 s
Maximum inference time	35.31 s	40.08 s	10.84 s	13.62 s	14.02 s
Average inference time	2.07 s	2.60 s	2.45 s	0.69 s	1.12 s

With these results, it is possible to apply the methodology described in this paper, obtaining the following punctuation for each model, shown in Table 5.

Table 5. Total points for the five models considered

Parameter	Based model				
	EfficientDet D0 512x512	EfficientDet D1 640x640	SSD MobileNet V1 FPN 640x640	SSD MobileNet V2 FPNLite 320x320	SSD MobileNet V2 FPNLite 640x640
Training time (seconds)	3	1	2	5	4
Classification error	1	2	3	5	4
Localization error	4	5	1	3	2
Regularization error	5	4	1	2	3
Average precision IoU=0.50:0.95	2	4	1	3	5
Recall IoU=0.50:0.95	1	2	3	4	5
False positives	5	4	3	1	3
Detection ratio	4	3	2	1	5
Minimum inference time	3	2	1	5	4
Maximum inference time	2	1	5	4	3
Average inference time	3	1	2	5	4
Total points	33	29	24	38	42

After applying the methodology, we found out that the model based on SSD MobileNet V2 FPNLite 640x640, which got 42 points. Therefore, we can consider that, taking into account the five models listed, for the same dataset, the model based on SSD MobileNet V2 FPNLite 640x640 performed better in general.

In order to complete the process of recognition, we used this fine-tuned model to detect licenses, and applied to the results the PaddleOCR library to recognise de number of the license plates detected. With this, we obtained a precision of 94.09% (from the 406 license plates contained in the dataset employed, 382 were correctly recognized and 14 were not).

6. Conclusion

A methodology to evaluate artificial intelligence-based models for the detection and recognition of Peruvian license plates was described. This methodology is proposed as practical approach to compare, from a set of models, which performs better according to given parameters.

There parameters were considered based on the application expected for the models. These covers the phase of training (fine-tuning), detection and inference. No ponderation was applied on these parameters, to avoid getting points biased.

Future work based on this methodology can include: considering other parameters for the assessment, pondering some parameters over others, applying the methodology to other tasks, etc.

Funding Statement:

The authors did not receive financing for the development of this research.

Data Availability:

No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflict of interest:

The authors declare that there is no conflict of interest.

References

[1]. J. Shashirangana, H. Padmasiri, D. Meedeniya and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," IEEE Access, vol. 9, pp. 11203-11225, 2021.

[2]. K. S. Raghunandan, P. Shivakumara, H. A. Jalab, R. W. Ibrahim, G. H. Kumar, U. Pal and T. Lu, "Riesz Fractional Based Model for Enhancing License Plate Detection and Recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, no. 9, pp. 2276-2288, 2018.

[3]. L. Zou, M. Zhao, Z. Gao, M. Cao, H. Jia and M. Pei, "License Plate Detection with Shallow and Deep CNNs in Complex Environments," Complexity, pp. 1-6, 2918. L. Zou, M. Zhao, Z. Gao, M. Cao, H. Jia and M. Pei, "License Plate Detection with Shallow and Deep CNNs in Complex Environments," Complexity, pp. 1-6, 2918.

[4]. S. Montazzolli and C. Rosito, "License Plate Detection and Recognition in Unconstrained Scenarios," in 15th European Conference on Computer Vision, Munich, 2018.

[5]. Y. Kessentini, M. D. Besbes, S. Ammar and A. Chabbouh, "A two-stage deep neural network for multi-norm license plate detection and recognition," Expert Systems with Applications, vol. 136, pp. 159-170, 2019

[6]. B.-G. Han, J. T. Lee, K.-T. Lim and D.-H. Choi, "License Plate Image Generation using Generative Adversarial Networks for End-To-End License Plate Character Recognition from a Small Set of Real Images," Applied Sciences, vol. 10, no. 8, p. 2780, 2020.

[7]. V. Chu, Sistema de Reconocimiento de Placas Vehiculares para Mejorar el Registro de Vehículos en el Hospedaje Suites Recreo - 2019, Lima: Universidad César Vallejo, 2019.

[8]. E. G., Sistema de reconocimiento de patrones en placas vehiculares para el acceso automático de visitas a un edificio, Lima: PUCP, 2014.

[9]. B. Ramirez and M. Tito, Reconocimiento automático de placas de rodaje utilizando una red neuronal convolucional para el ingreso de vehículos en la universidad Ricardo Palma, Lima: Ricardo Palma University, 2020.

[10]. Y. Du, C. Li, R. Guo, X. Yin, W. Liu, J. Zhou, Y. Bai, Z. Yu, Y. Yang, Q. Dang y H. y Wang, "PP-OCR: A Practical Ultra Lightweight OCR System". Available: http://arxiv.org/abs/2009.09941. [Last accessed: 24 May 2024].

[11]. Tensorflow, "Tensorflow Detection Model Zoo" [Online]. Available: https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md. [Last accessed: 24 May 2024].