Computer Vision
Car license plate recognition

Emanuele Corongiu <xcoron02@stud.fit.vutbr.cz>

December 23, 2023

1 Introduction and objectives

In this work, an attempt was made to train a YOLO (You Only Look Once) model for the purpose of recognizing and tracking car license plates. The model was trained under different conditions in order to appreciate some differences in its results. The most interesting part is to observe the training method and how this can then be reflected in the results obtained in terms of accuracy and loss metrics. Once the training was completed, a small experiment was also performed with a demo video in which Yolo's integrated tracker was tested.

2 Brief summary of methods in the literature

Yolo is not the only way to get locate objects of this type. An example of this is the method described in [1], which implements an ALPR algorithm based on image processing, i.e., frame decomposition and edges analysis. It was designed to be low computational cost and was mainly implemented through the use of the open CV library. Another example may be the method described in [2] which is based on a weight model.

In general, we can observe two main methods [2] in the literature, one more classical based on image processing, and one based on machine learning techniques, such as precisely Yolo.

3 Yolo V8

Yolo (You Only Lool Once) [3] is a well-known model used for object detection and image segmentation that uses, among other techniques, convolutional neural networks [4]. It represents an excellent tool that is well suited to the purpose of this project, which is given input images, or at least frames of a video, to detect and track license plates of vehicles. Yolo is a powerful and comprehensive tool; the model comes in various sizes and is currently up to version 8 (used here). This tool requires receiving the input dataset by splitting the images into training, validation, and test, respectively, where for each the information within the image should be given in text format. The nano model (3.2M parameters [3]) of YOLO V8 was used in this work.

4 Dataset

Two different datasets were used to train the YOLO model in order to observe the difference in the attenuated performance in accordance with the volume of data used. The first step was to train the model with a small dataset [5] containing a total of 301 images. The authors of the dataset obtained, with a yolov8s (11.2M parameters [3]) model, the following results, mAP: 99.5%, Precision: 97.5% and Recall: 98.9%. In the figure [1], is possible to see more details.

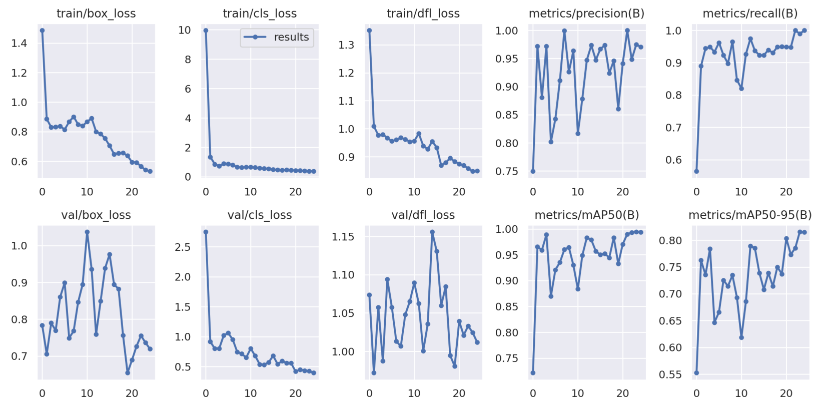

Figure 1. Results from [5]

The second dataset [6] used contained 8683 images of which 6083 were for training, 1733 for validation and 867 for testing.The authors reported the following results with the yolov5 model, mAP: 84.7%, Precision: 88.8% and Recall: 80.8%. In the figure [2], is possible to see more details. A more detailed description of the meaning of these graphs will be give in the next section.

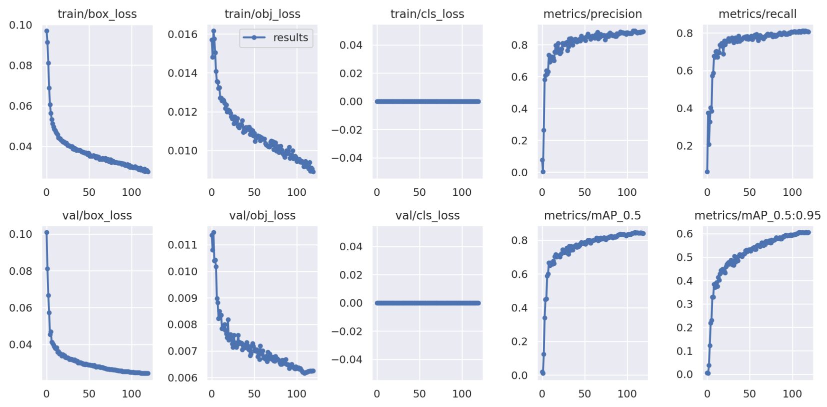

Figure 2. Results from [6]

5 Training and results

Training was performed using the yolov8n model, in a first step, using the dataset from about 300 images. Then the second dataset was used so as to analyze, also qualitatively the impact of the starting dataset on the achievable performance. Both tests were performed on 100 epochs. In the images [3] and [4] the results generated by the model can be observed.

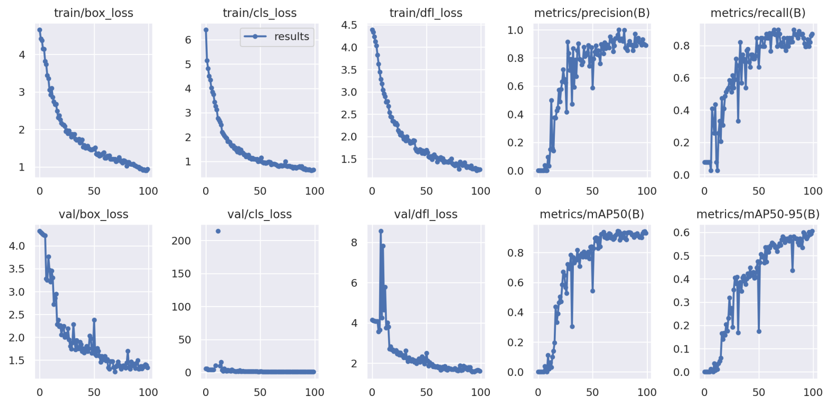

Figure 3. Result after training with the small dataset [5]

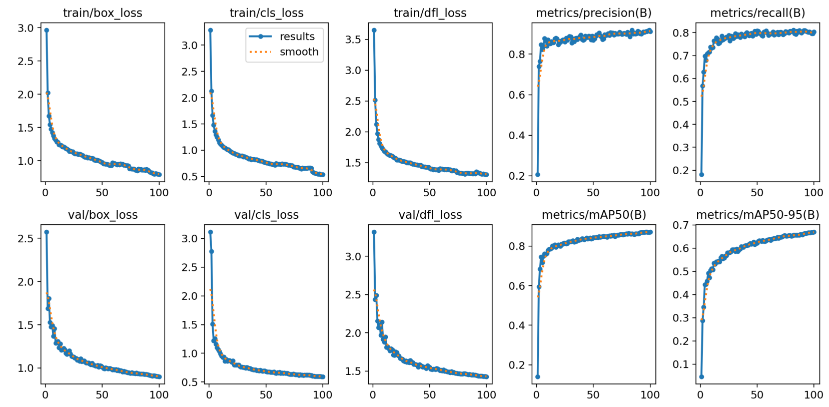

Figure 4. Result after training with the big dataset [6]

Within these graphs, we can observe on the X-axis the number of epochs correspondent for each point on the Y-axis. The number of epochs represents "the number of times all training data passes through the model architecture" [7].

The Y-axis can take on different meanings, in particular here we have five different metrics:

- Box loss: is a loss metric, so the lower the better. It represents the accuracy with which the box containing the plate is defined compared to the box given as input;

- class loss: another loss metric, but one that refers to how well the model can classify objects;
- mPA: this figure represents a summary of the previous two metrics by evaluating the average accuracy over all the data. This datum is usually referred to the Intersection over Union (IoU), defined as a value proportional to the area given by the intersection of the true bounded box and the predicted bounded box.Compared to this usually the mPA 50 and mPa 50:95 are given, where 50 and 95 represent the percentage of overlap between the bounded dox;
- Precision: represents the accuracy of the model, defined simply as the ratio of the number of bounded boxes revealed to their actual number;
- Recall: unlike Precision, this metric relates the number of true bounded boxes to the total number of predicted bounded boxes among which there may be false positives.

These metrics, as can be seen in the figures, were evaluated for both training and validation. The main differences can be observed in the loss metrics that did not reach values the same values. As can be seen from a comparison of the figures [1] and [2], with [3] and [4], respectively, the results obtained do not differ significantly except for the loss metrics. In particular, this may be due to the use by the authors of the two datasets of pre-trained models and larger models in general. Recall that the results obtained from this project are from the smaller Yolo V8 model. Also note the irregularity in the curves for the 300-image dataset due to the small number of data used. In general however, it can be observed, as expected, that the model trained with more than 8000 images obtained better results in loss metrics than what was obtained using the small dataset, a difference that is less pronounced in terms of accuracy and mPA.

6 Conclusion and possible developments for the project

In conclusion in this project training was performed with two different datasets of the Yolo V8 nano model. The model also generated data to evaluate the performance of the model and were compared with the results obtained by the authors of the two datasets. Next, a tracking test was performed although it is only mentioned in this report. An excerpt of the output video can be observed in figure [5]. The demo video was taken from [8].

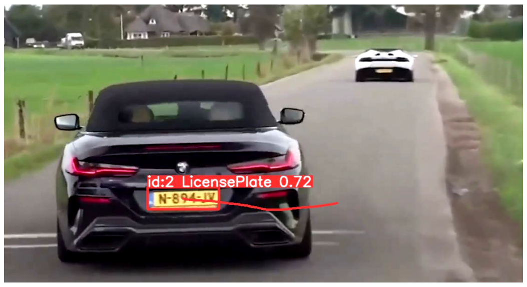

Figure 5. Result of the trained model used to perform tracking

One possible development for this project that has not been pursued here is the ability to read license plates. Through Optical Character Recognition (OCR) it is indeed possible to implement a license plate reader that allows, once identified with Yolo, the text inside to be read and extracted. One such project can be observed in [9].

References

[1] N. Kharina and S. Chernyadyev, "Software for car license plates recognition with minimal computing resources," in *2022 24th International Conference on Digital Signal Processing and its Applications (DSPA)*, pp. 1–4, 2022.

[2] Z. M. Gizatullin, M. M. Lyasheva, M. P. Shleymovich, and S. A. Lyasheva, "Automatic car license plate detection based on the image weight model," in *2022 Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pp. 1346–1349, 2022.

[3] F. Jacob Solawetz, "What is yolov8? the ultimate guide.." `https://blog.roboflow.com/whats-new-in-yolov8/#:~:text=YOLOv8%20is%20the%20newest%20state,and%20industry%2Ddefining%20YOLOv5%20model.`, 2023.

[4] R. J. B. U. of Technology, "Cv course's slides: Object detection and tracking," 2022.

[5] project 8vcch, "carplate dataset." `https://universe.roboflow.com/project-8vcch/carplate-xuk6s`, aug 2023. visited on 2023-12-23.

[6] projekt, "Tablice dataset." `https://universe.roboflow.com/projekt/tablice-73he1`, nov 2023. visited on 2023-12-23.

[7] M. Traore, "Roboflow train: Understanding training graphs." `https://help.roboflow.com/faqs/roboflow-train-understanding-training-graphs`, 2022.

[8] Arijit1080, "Licence-plate-detection-using-yolo-v8." `https://github.com/Arijit1080/Licence-Plate-Detection-using-YOLO-V8`, 2023.

[9] computervisioneng, "automatic-number-plate-recognition-python-yolov8." `https://github.com/computervisioneng/automatic-number-plate-recognition-python-yolov8`, 2023.