



MoroccoAI Data Challenge (Edition #001)

Automatic Number Plate Recognition (ANPR) in Morocco Licensed Vehicles

Who we are ? TTM : The Three Musketeers



Hamza RAOUZI

Second year master degree (Intelligent processing systems) student at Faculty of Science in Rabat.

Deep learning engineer intern at Deepecho.

[in /in/hamza-raouzi/](#)

[/hamzaraouzi](#)

hamza.raouzi@um5r.ac.ma



Safoine EL KHABICH

Second year master degree (Intelligent processing systems) student at Faculty of Science in Rabat.

Machine learning engineer (MLOps) intern at Beewant.

[in /in/safoinme/](#)

[/safoinme](#)

safoine.elkhabich@um5r.ac.ma



Younes OUASSINE

Data Science student at second year master "Intelligent Processing Systems", Faculty of Science Rabat.

[in /in/younesouassine/](#)

[/SmarT-Ouassine](#)

younes.ouassine@um5r.ac.ma



The challenge

Automatic Number plate Recognition (ANPR):

Is crucial for law enforcement and authorities in order to assure the safety of the roads and to check the registration and the licence of the vehicles.
Therefore the aim to automate this task is very beneficial.

Dataset

The dataset is 654 jpg pictures of the front or back of vehicles showing the license plate. They are of different sizes and are mostly cars.

The documentation refers to the original dataset provided by Modeling Simulation and Data Analysis (MSDA-laboratory) of Mohammed 6 polytechnic university.





Our approche

Our approach was to use Object Detection to detect plate characters from images. We have chosen to build two models separately instead of using libraries directly like easyOCR or Tesseract due to its weaknesses in handling the variance in the shapes of Moroccan License plates.

The first model was trained to detect the licence plate to be then cropped from the original image, which will be then passed into the second model that was trained to detect the characters.

Like any ML/DL Problem and in order to build our system we had to follow the next cycle :

- 1- Label and Annotate The images and Prepare the data for modeling
- 2- Select a Library and Model Architecture
- 3- Fine-Tune Hyperparameter and Train the model
- 4- Post-Process and Evaluate the result
- 5- Repeat the cycle until reaching a good result

Data Labeling and Annotation

First we start by annotating the dataset on our own using a tool called Labellmg.

Then we found that the dataset provided by MSDA Lab was publicly available and fits our approach, as they have prepared the annotation in the following form :



Figure 1: MSDA Lab dataset structure .

	Class	x min	y min	x max - x min	y max - y min	Image name	Width	Height
0	LP	322	162	139	40	20200614_162449b.jpg	720	960
1	LP	262	178	168	32	20200614_162450b.jpg	720	960
2	LP	299	128	85	30	20200614_162501b.jpg	720	960

Figure 2: Plate bounding boxes .

4936	7	105	19	36	73	20201106_224622.jpg	503	138
4937	9	142	21	36	74	20201106_224622.jpg	503	138
4938	3	181	25	37	72	20201106_224622.jpg	503	138
4939	h	311	47	50	46	20201106_224622.jpg	503	138
4940	6	433	41	39	72	20201106_224622.jpg	503	138

Figure 3: Characters bounding boxes .

Select library and model architecture

we have choose faster-rcnn model for both Object detection tasks, using library called detectron2 developed by FaceBook AI Research Laboratory (FAIR) based on Pytorch

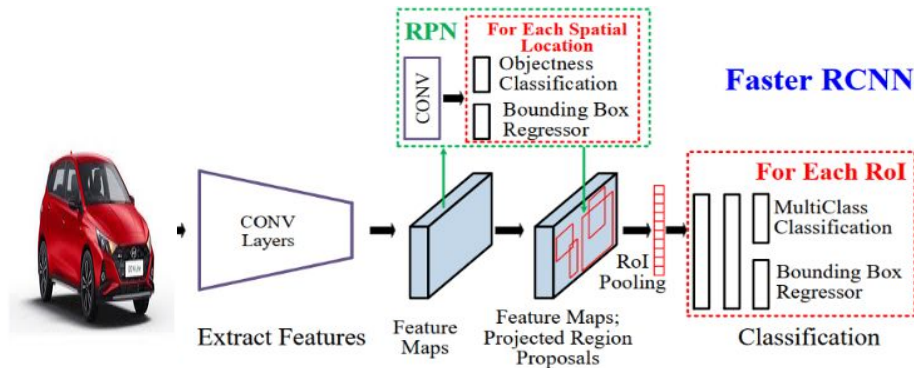


Figure 1: Faster RCNN for Plate detection.

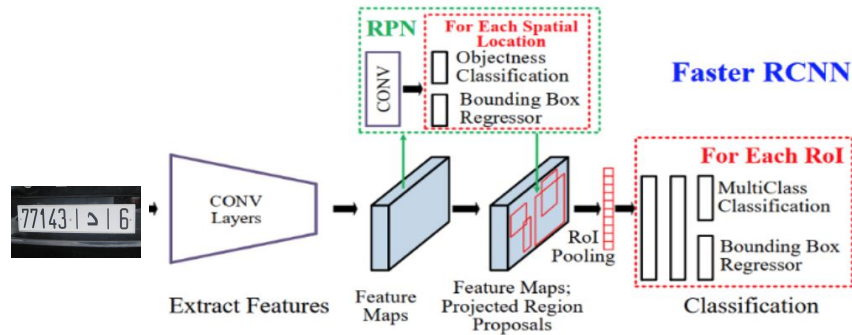


Figure 2: Faster RCNN for characters detection.

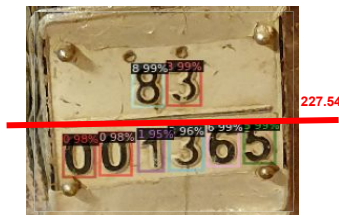


Modeling

The both faster-rcnn models were pretrained on the COCO dataset, because we didn't have enough data, so it was straightforward to take advantage of transfer learning of models that were trained on such a rich dataset.

Post-Processing

In the post-processing the main problem we countered after building a good model that could detect almost all characters is that the order of the plate_string was wrong for this reason we did algorithms that do 2 things :



1. Split characters based on median of Y_Min of all detected letters boxes, by taking characters where their Y_Max is smaller than Median_Y_Mins into a string called top_characters, and those who have Y_Max greater than Median_Y_Mins will be in bottom_characters.
2. Order characters in top and bottom list from left to right based on the X_Min of the detected Box of each character



Improvement

To improve our result we have created 50 copies of The 3 images that contains the “ج”, “M”, “المغرب” in the training set while doing some transformations since we had no other source for more plate images with this specific characters.

0.21568 0.83333 \longrightarrow 0.18627 0.79411

Then after we couldn't improve the model anymore with the type and number of data we had to decrease the threshold from 0.7 to 0.5 because in this problem, detecting some letters with lower confidence would lead to better results than not detecting it at all. especially if it's the letter “و” since it transfers into “waw”.

0.18627 0.79411 \longrightarrow 0.11764 0.74509



Thanks for your attention