AI Tech Assessment: Artificial Intelligence Intern
Deep learning analytical thinking **Ramon Andreu**

What do you think it occurred during this model development (trainning & evaluation)?

There are three possibilities that may occur:

1. The system may not have been done having in mind that the validation of the model should have been oriented to avoid false positives. Probably the validation metric is the accuracy, when it should have been the F1-Score or the precision. That's why errors occur when license plate numbers are assigned that are slightly different from the actual number.

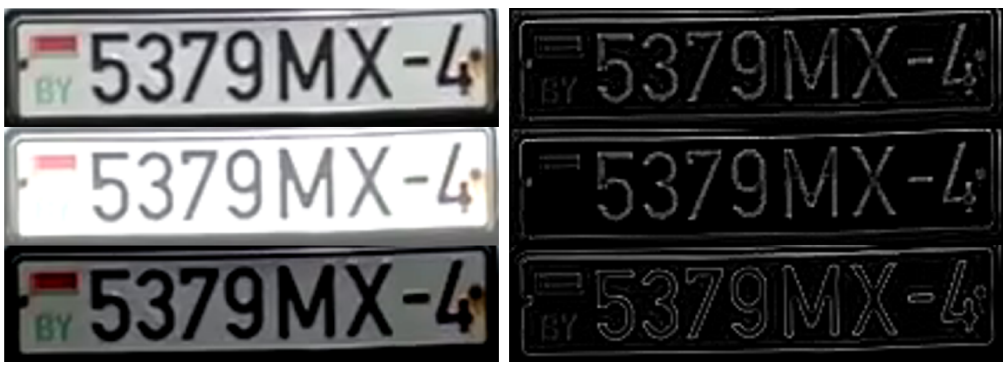
2. The database could have a lack of quality, maybe, it does not cover the diversity of lighting situations that exist throughout the day, and so, the model could have problems reading the plates.

3. As we know from the statement of the exercise, there are cases when a person was fined for a vehicle of a similar make and color but with a slightly different registration number. This could be an indication that the model may rely on other information besides the plate numbers.

How would you fix this behavior? Please provide at least 2 options explaining their pros and drawbacks

- **First Solution**
  To solve all three problems, the architecture must be redesigned. To begin, the image must be preprocessed as follows:

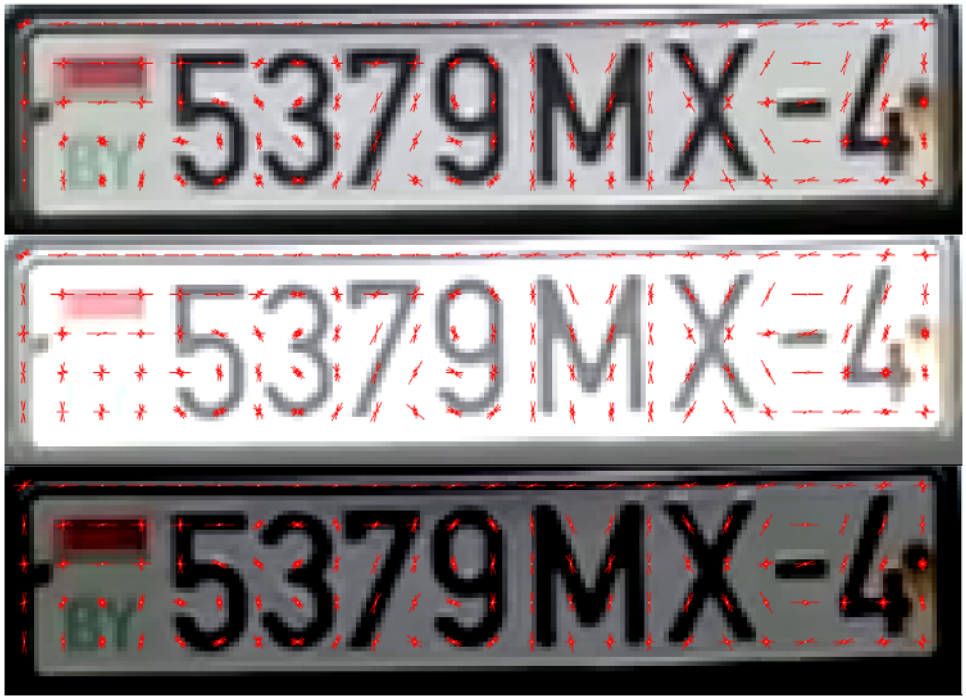
  1. To avoid dependencies on information other than the license plate, we can use a masking algorithm (like YOLO) to identify the license plate and take it apart from the original image.
  2. Then, to achieve invariance to changes in lighting throughout the day, we will use the digital gradient of the license plate image to train the Deep-Learning model:



3. When, training the Deep-Learning model, use the F1-Score as a validation metric. Also, we treat each character of the plate as an individual (to achieve better results, this way we can tune the F1-Score metric to count each character as a TP, FP, TN or FN).
4. **Pros and drawbacks:** Although this system tries to solve all the problems described (False Positives, low quality of the dataset and not relying only on the information of the car plate), using a simple Deep-Learning model could not be enough, and also, the robustness given by the digital gradient of the image could also not be enough to solve the problem.

- **Second Solution**
  We need, again, to solve the problems described above:

1. As we did with the first solution, to avoid dependencies on information other than the license plate, we can use a masking algorithm (like YOLO) to identify the license plate and take it apart from the original image.
2. Another way to achieve invariance to changes in lighting throughout the day, is using the Histogram of Oriented Gradients of the car plate image to train the model:



We can see that the direction of the gradients from the images is almost identical with each other, although the images are clearly different.

3. Another way to avoid false positives is by using the machine learning algorithm called AdaBoost (often used to recognize human faces), with its cascading architecture, which ensures that the analyzed element is part of the target class.
4. **Pros and drawbacks:** Although this system tries to solve all the problems described (False Positives, low quality of the dataset and not relying only on the information of the car plate), we are not using the potencial of a Deep-Learning model, and also, the robustness given by the Histogram of Oriented Gradients of the image could also not be enough to solve the problem.

What do you think it will occur when running this AI in a different country with different plates formats? How would you ensure system accuracy?

The length, font, layout, and even color of license plates differ depending on the country they originate from. So, probably the original system will be useless.

To ensure a good performance, the system would need to be fine-tuned with a database with all car plate formats that exists, although it may hinder convergence.

Do you know any OCR (Optical Character Recognition) algorithms (Deep learning based) that could be used here?

Yes, there are a lot of options:

- Tesseract
- GPT-4 (with vision version), in fact any LLM with a vision module could be usefull.
- CRNN (Convolutional Recurrent Neural Network)
- TrOCR (Transformer-based OCR)
- PaddleOCR

Explain a Computer Vision / Artificial Intelligence project in which you have participated (goals, your role, difficulties you found, how they were solved, ...)

A companion and I made a Deep-Learning classifier model to identify the type of artworks using Marenostrum 5, where we were also challenged to make a preprocessing to a large database of images, the MAMe Dataset (with 205 GB of images).

First we had to make a script to manage all the information, first the images, which we treated to obtain the best results (Gaussian filter to eliminate noise, a correction of

the illumination of the images (as there were differences), etc.). And then, we treated the rest of the data: Non-balanced classes, outliers...

And the last thing we had to do is create the architecture, which had to be feed-forward. We took inspiration from the AlexNet architecture, but we had to do a lot of "try and error" to find the best architecture, and then, the best parameters.