Fluendo recruitment questions

Deep learning and analytical thinking

Adrià Cebrián Ruiz

Contents

Question 1: What do you think occurred during this model development (training & evaluation)?

After reading the articles provided, I found out some key information for the diagnosis of the issue. The first report said that the license plate was misread by just one letter or number, specifically they mention the confusion between a 'G' and a 'C' character, which are very similar. The other report states that the images taken of their license plate were "blurred".

Taking into account these two facts, my theory is that the deep learning model was trained using a data set with clear pictures, without any blur or noise. And when trying to read plates in a real environment, where the images might have noise, it failed to perform correctly.

Question 2: How would you fix this behavior? Provide at least 2 options with pros and drawbacks

Given my conclusion from my previous answer. Two options come to mind:

Option 1: Train the model with a noisy dataset

Pros:

- Would enhance the model to cover a wide range of real world inputs.
- Doesn't require changes in the infrastructure.

Cons:

- Gathering and/or generating noise that is realistic is very challenging. Making it difficult to build a proper dataset for training.
- It could take some time to re-train the deep learning model.

Option 2: Upgrading image quality

This option could be done by upgrading the already installed cameras, or by performing some image processing technique to reduce noise (like *opening*, using a Gaussian filter, or taking the median using a kernel). But the most promising would be to upgrade the cameras, since the most important constraint is the blur of the images.
Pros:

- Wouldn't have to change the already trained model!

Cons:

- Would be excruciatingly costly.

Question 3: What do you think will occur when running this AI in a different country with different plate formats? How would you ensure system accuracy?

When deployed in a different country, the model may fail to classify due to differences in license plate formats.

A straightforward approach to this problem would be to train the model with the original data as well as the data from that other country, to make sure the model doesn't overfit to a single license type.

This would require some tuning, but would be doable!

Cool possible alternative implementation

In Machine Learning,, there exists a type of learning called *destillation learning*, where a small model learns from a large one.

My idea would be to do quite the contrary, training a model for each specific country first, and then make the large model learn from the set of smallest models. Searching for information about this subject, I found that this concept is called *ensemble distillation*.

Question 4: OCR Algorithms for License Plate Recognition

Two widely used deep learning approaches in Automatic Number Plate Recognition are CRNN and LPRNet.

- **CRNN** combines convolutional layers for feature extraction with recurrent layers for sequence modeling. It performs well on standard license plates, which makes it suitable for our purposes.
- **LPRNet (by Nvidia)** is a faster, fully convolutional aproach optimized for real-time use. By removing recurrent layers and relying solely on 1D convolutional decoding, it achieves comparable accuracy with faster inference.

Question 5: Explain a Computer Vision / Artificial Intelligence project in which you have participated

Cartoon Frame Classification

The goal was to develop a model that classifies individual cartoon frames based on the cartoon they belong to.

This project helped me learn about data processing and feature extraction. The main language used was Matlab, since it is very user-friendly and makes it very easy to train all kinds of models (SVM, Classification Trees, Neural Networks...).

The main challenges were finding the discriminant features for the model. Also we had very few images, so we couldn't go with deep learning approaches and large models.
We ended up identifying some features like RGB normalized color extraction, Oriented Gradient Histograms and Local Binary Patterns that ended up working fine (92%).

The project gave me some practical experience with image classification and dataset preprocessing.