Technical test – FLUENDO

Artificial Intelligence Internship

Elena Blanco López

eb4431@outlook.es

elena.blanco.lopez@estudiantat.upc.edu

01/06/2025

First question

The three primary errors reported by the victims about the Deep Learning–based ANPR system used at the Dartford Crossing are:

1. Not detecting license plates from new vehicles not registered in the ANPR systems.
2. Generating the same license plate for different vehicles with similar apperance.
3. Misreading license plates that differ by only few characters, leading to incorrect matches.

Additionally, some victims reported that the images the ANPR cameras captured were clearly blurry. Based on these declarations, the following potential issues during model training can be inferred:

Dataset quality and model generalisation: The training data may lack sufficient diversity, which limits the model's generalization ability, as it was reported in the first issue. Diversity comprises images from different resolutions, scales, climate scenarios, vehicle models with different sizes and colors, etc.

Model misfocus: The model appears to focus on parts of the image it shouldn't, such as the vehicle's appearance rather than the license plate (second issue). This could be detected using explainable methods, for example Grad-CAM which produces "visual explanations" (R. R. Selvaraju et al. [1]).

Model architecture: The model may fail to capture broader contextual information. For instance, if the model doesn't implement attention mechanisms and relies on standard convolutional layers with small kernel sizes, the receptive field may be too limited. This contributes to issues as the last one in the list.

Regarding model evaluation, it's possible that the test set didn't reflect the full range of real-world scenarios and challenges. As a result, the model may have presented a not representative high test accuracy leading to its deployment, without the engineering team having the full knowledge of its real-world limitations.

Second question

On one hand, if inference speed is not the major concern, which I believe is the case here, I would recommend a two-stage approach. This involves training two separate models: one for object detection (Faster R-CNN, YOLO) to localize the license plate, and another for character recognition (OCR model), which is fed with the cropped plates. This separation makes it less likely for the system to confuse vehicles with a similar apperance, as the recognition model operates only on the cropped region containing the plate.

On the other hand, given that license plates follow a standardized format and are specifically designed for machine readability (high contrast between characters and background), deploying a complex vision transformer model may be costly and unnecessary.

Based on these considerations, I recommend using **Faster R-CNN** for the license plate detection task, as it ensures very high accuracy, and a **lighter OCR model** for the character recognition task. For the OCR component, my suggestions would be:

RCNN + CTC loss approach: A set of fully convolutional blocks which encode visual features at different levels, followed by a RNN sequential decoding and CTC loss (especially used when the alignment between the input and output is unknown).

Transformer-based approach: Incorporating attention mechanisms to the encoder and decoder architectures for better context capturing.

The first approach is computationally cheaper than the second but it has slower inference time because it follows a sequential decoding rather than a parallel decoding (as in the transformer-based approach). However, it is also worth noting that transformers can be too complex for this task and typically require larger datasets to achieve good generalization.

Additionally, collecting data that captures a wide range of real-world scenarios and making thoughtful dataset splits is essential. The dataset should include license plates with a wide variety of character combinations and font variations, a diverse range of vehicle models and colors, as well as the inclusion of images with varying resolutions and different climate conditions, such as rain, poor lighting on cloudy days, or nighttime scenarios. For detection, applying data augmentation relevant to the task would also improve a lot the model's capability to generalize and, in specific, recognize plates from models not yet registered by the system.

In conclusion, constructing a diverse dataset, applying data augmentation techniques consciously, designing an architecture suitable for the task and evaluating the model on a representative set of data, are the key improvements I would implement.

Third question

Since the model sometimes fails to make correct predictions on new vehicle models not yet registered by the system, it's likely to also struggle recognizing license plates from different countries. As I previously mentioned, the poor generalization ability of the detection model could be improved by applying data augmentation techniques relevant to the problem. For example, applying rotation might not be useful here, since the camera will always capture images in the same orientation.

Fourth question

In an Object Detection course I have took this semester we got introduced to the image-to-text task in computer vision, and I remember studying the RCNN combined with CTC loss approach. However, we didn't go into much depth on OCR algorithms, so I'm not completely up to date with the current state-of-the-art.

Fifth question

I have coursed Computer Vision and Object Recognition in my master's degree and worked developing deep learning-based vision models for several different tasks: object detection and recognition, multi-label classification, semantic segmentation and depth estimation.

The last one consisted specifically in predicting body and cloth depth from a synthetic dataset. The work started with a baseline UNet-2D model and consisted in and progressively incorporating several improvements: hyperparameter tuning, applying data augmentation relevant to the problem, substituting the UNet-2D backbone by a pretrained vision transformer encoder, developing a perceptual loss, and finally, create a color-coded SMPL pose image and concatenate it with the RGB image as input. The goal was to implement and evaluate each of these enhancements to determine the best-performing model. I took care of the preprocessing part, the hyperparameter tuning, which consisted in a grid search over learning rates, batch sizes and convolution filter sizes, and lastly, developing the code for integrating transformer-based encoders with the UNet decoder. Specifically, we experimented with two architectures: a DeiT-base pretrained encoder paired with a UNet decoder (including a Center block) (A. Varma et al. [2]), and the TransDepth encoder combined with the same UNet decoder structure (G. Yang et al. [3]).

The main difficulty here was the limited computational resources available. We had to alternate between using Kaggle and Google Colab GPUs to train our models, which often led to interruptions. Sometimes errors would occur midway or even at the end of long runs, such as ROM-related issues. To address this, we implemented checkpoint callbacks to be able to resume training even when the kernel crashed, as well as saving the metrics and loss histories to enable later analysis or plotting.

Personally, I also found it a bit challenging to implement some of the models described in the research papers from scratch. It is also worth mentioning that the DeiT-UNet model at first wasn't learning anything, the learning curves were almost completely constant. It was quite challenging trying to discover what was the problem and after an exhaustive exploration I found out that changing the model's output activation function from sigmoid to linear enabled the model to effectively learn. I believe this was due to the ”squashing” performed by the sigmoid function, the gradient can then become very small (sigmoid saturation), and the model fails in learning.

References

---

[1] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, “Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization,” CoRR, vol. abs/1610.02391, 2016. arXiv:1610.02391. [Online]. Available: http://arxiv.org/abs/1610.02391.

[2] A. Varma, H. Chawla, B. Zonooz, and E. Arani, “Transformers in self-supervised monocular depth estimation with unknown camera intrinsics,” CoRR, vol. abs/2202.03131, 2022. arXiv: 2202.03131. [Online]. Available: https://arxiv.org/abs/2202.03131.

[3] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, “Transformers solve the limited receptive field for monocular depth prediction,” CoRR, vol. abs/2103.12091, 2021. arXiv: 2103.12091. [Online]. Available: https://arxiv.org/abs/2103.12091.