I have created a sample project to explore what would affect the model's ability to recognize the plates with a simplest deep learning model and synthetic dataset just to grasp the idea of the question. Instructions on how to run the script are written in the README file in side the script folder.

In ANPR algorithm these are several steps it take to recognize plate but to make this case simple and only relate to model the step where deep learning OCR are developed,

The model consists of a 2 CNNs layer with simple hyper parameters and trained with a cropped character from synthetic license plate, created from 5 different font styles. The model is able to classify character by character correctly from the synthetic test license plate.

However, with a real image there's nowhere near applicable. The model failed miserably:(Mainly because different data and images are wrongly resized, padded and cropped. Despite this, getting hand-on the problem gives me a room to critically figure out a sense of possible problems within the question contexts. The answers below assume that these problems occur because of model development only.

Here are my thought and answer on the question

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

To answer this question, several things can be consider a problem which occur in the model development process so I will group them into each part:

- Dataset problem - For the first problem, the training image could be different from the real world images, this could be the case of the quality of training image are better than the actual images, for example the brightness on the training plate, or the bended testing plates. Thus the model only recognize a pattern from the training image and unable to calculate the same weight for the real image. Take my project for a vague example. A model only been train on 5 selected fonts , and the test image were construct from totally different fonts, the curve and edges which model look for in the real image represent a totally different character.

  For the second problem, this mean the model take into account of the car appearance as well, which can be beneficial if a feature of car appearance available with their license plate data. However in my opinion, this is not appropriate and shouldn’t be happening when recognizing a license plate. The model should focus only on the character part and only classify based on that. Other features like a pixel which contain car’s color or brand should be removed before feed into model as this could make out model prioritize a wrong pattern thus effect model losing its ability to classify character.

- Model Architecture problem - The model design are inadequate or not properly tune for the problem. Lets assume the original model use 2 layers of CNNs with a filter size of 32 and 64 both with a stride of 4. This could contribute to detecting a wrong character because of the filter or layers are not sufficient enough to capture a

complex pattern of each character / the number of stride are too huge that some pixel got skip and lead to less information in the model. The CNN architecture only might not be enough to represent the semantics of each pixel.

- Training Process problem: Number of epochs, batch size , learning rate, are also crucial. An example would be the effect of choosing too few epochs and the model underfitted or too high that the model memorize training data and overfitted. The learning rate directly influence how the weight is change from the gradient, too small and the model learn super slow and stuck in the local minima, too large the weight shift back and forward and diverge.

- Evaluation problem - Evaluating the training dataset can be tricky, accuracy/loss metrics can be bias toward training dataset thus the model are overfitting. In this case using validation accuracy/loss instead can be better for this problem or include a reinforcement learning system to adjust the model weight to the right direction.

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

The problem can occur from many area between training and evaluation thus I propose a couple of option that need to be done to make sure the problem are detected and control.

First option, I would increase the variety and amount of data use in training, some data augmentation like rotating, adjusting brightness or more should be apply to the training set for model to learn all conditions.

Pros: This would improve generalization of the model and resilient toward noises environmental change.

Cons: Training will take longer time and it could introduce more complexity than needed from unrealistic transformation.

Second option, reconfigure better model architecture which can capture more complex pattern including adding one more layer of CNNs or increasing filters size. In this part other deep learning model and technique could be added. I would suggest adding residual connection which help in training deeper networks by allowing gradients to flow more easily during back propagation. Also adding gradient clipping to set a threshold for each gradient in each timestep could be beneficial.

Pros: Increase complexity to capture more pattern in the training image, residual connection could help reduce the vanishing gradient problem, gradient clipping prevent model from the problem of exploding gradient. Overall make the model learn richer features.

Cons: Increase complexity and computational cost and introduce higher risk of overfitting. Moreover, the model need to be proper design and tuned the architecture and hyper parameters carefully to balance performance and efficiency

Third option, Applying a transfoer model instead by using CNN as a encoder as it were more meant for classification task not generation and use a regualr transformer decoder to decode information from the CNNs. This could be more suitable and fix the problem as it compute attention weight between encoder and decoder which will contain more contextual information about each pixel than the original model. The encoder- decoder architecture allow parallelize training and many technique/ trick for example, teacher forcing. This could train way faster than original model but hyper parameters need to be carefully selected to make the model learn properly.

Pros: Allow the model to focus on specific regions of the image relevant to each character (helps with distorted, misaligned, or not clear plates).

Cons: Need a large dataset to generalize well and hyper parameter like attention head are very sensitive to data and changes.

3.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?

We need to first consider what are the different between license plate in EU and other country. As far as I know I can group to two main differences and this is how I would ensure accuracy on these diferences:

1. Format and structure, while eu has a consistent format of 7-8 character which are either latin alphabet or number, and it contain a blue section on the left to indicate country code. while other country like USA format can be varies from states to states, country like Japan or Thailand has a different alphabet thus required a modification in order to train the same model that could classify all kind of plates. In this case I'd choose to use an adapter idea to frozen first model weight and train another weight using different language image, then it is able to turn on and off for different language character.
2. Different spacing and segmentation, the image of a plate will have totally different layout and spacing between character. To tackle this problem, we should build another model to classify license plate type/country in order to select the right preprocess and ensure the data use to predict are in the same size for the main model to classify character.

Combining the two solution, the differences is mitigated and our model will be able to classify any country license plates. :)

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

From researching, I found TrOCR, which is a transformer model that combine Visual transformer as an encoder and normal transformer as a encoder. I personally have no experience on this technology so far but I know it is a big leap forward and a current state of art. The model contain a Visual Transformer which feed the input of each image as a patch pieces and positional embedding this could enrich image contextual information. Then use a encoder-decoder attention to generate an output.

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

An Artificial Intelligence project I have participate in are on the topic of building a model for a classification task “Drug-Drug Interaction”, where we classify 4 different type of interaction between each drugs using different architecture and collect information on the advantages and disadvantages of each architecture. My role is to research, design and fine tune the architecture of the model. One of the difficulty is that I have waste time when first starting the project, by applying the wrong approach. With the aim of achieving high accuracy for the task, the approach of action first and try out all model all hyper parameter exhaustively was what i went for and with only basic knowledge of how to use each architecture, the results were really bad and training time take forever to get the good result. This effect the project’s timeline, make the time management and communication unstable. To solve this, I start learning and planing more of which approach to try and what would be the expected result and why. For each architecture in deep learning required a thorough understanding in order to select the most suitable ones for which the task. I review information about CNNs, LSTM , BiLSTM, RNN and Transformer and try each model with different hyper parameter according to the way its modify the algorithm. I assign my thought of how each model would perform and which one should be better in which area. I end up with a simple CNN+BiLSTM model which able to reach a great result with the most generalization and i am able to update and communicate directly my timeline and plan along the project with my colleague which make the project more carefully progress into the right direction.

References

https://medium.com/@tejpal.abhyuday/trocr-transformer-based-optical-recognition-model-811f7b3217da