Assignment 1

Part 1:

Results:

```
Loading model ../checkpoints/exp1/best.ckpt

100%|██████████| 16/16 [00:09<00:00,  1.61it/s]

<Figure size 800x800 with 20 Axes>

Character Accuracy: 99.93
Word Accuracy: 1.00
```

Part 2:

The data has been split into test and train (80/20 ratio) and saved in the plates_1 folder.

Part 3:

The best-performing model was achieved by setting the number of epochs to 100 and the numberof batches to 5. The total character accuracy is 19,41%. There were other attempts to experiment with the parameters and the results of those experiments are mentioned in the notebook as well.

A Levanstein distance has also been added as a part of the model's performance evaluation.

Part 4:

The original code had word accuracy results, which in the case of using license plates as images, did not carry any significance as license plates normally do not have any words in them, therefore word accuracy has been excluded from the model analysis. It has also been decided to calculate symbol precision and symbol recall instead.

When it comes to the results of the new model, its character accuracy is 19,41%, which indicates that the model is able to recognize individual characters on the license plates. However, the low results in symbol recall indicate that the model might be too “sure” about its precision but is not able to recall the symbols that are present in the ground truth labels.

Low Recall for specific symbols might be an indication that a larger dataset or architectural adjustments are needed. Additionally, there is a chance that the OCR model makes contextual errors based on the scarcity of the data, which results in its poor results. For instance, it might confuse “O” with “0” or “I” with “1. Lastly, image quality might also be the reason for low results.

Improvement proposals:

1. Transfer learning – it is worth trying to use pre-trained models as a starting point for the OCR model.

2. Fine-Tuning - trying different modifications such as changing learning rate, batch_size, epochs, etc. could also facilitate the model’s results.

3. Data Augmentation – by applying transformations to the images in the dataset (rotations, translation, brightness, sharpness, etc.), it could be also possible to score higher results.

4. Bigger Datasets – the dataset that has been used for this assignment has only 133 pictures for training and 34 pictures for testing, which might not be enough for the model to learn and perform adequately.

5. Ensemble Models – it is also worth using predictions from several OCR models with different architectures to get a better accuracy score.

6. Weight Decay - this feature could be used to reduce overfitting of the model.

Part 5:

It has been decided to update the CRNN architecture to make it more flexible so that it can be used for more tasks and various images, therefore a new convolutional and recurrent layer has been added. The thinking behind this modification is the following:

1. Model Functionality: By increasing the depth of the neural network, one can increase its capacity to learn more challenging patterns.

2. Feature Extraction: Each layer is responsible for learning different part of data. Convolutional layers are used to work with small features like object's edges, while recurrent layers are able to capture high-level patterns. By adding one more layer of each, a chance to get better results arises.

3. Large Context: In CRNN, recurrent layers are used to handle sequential data. An additional layer can provide the model with more context and memory for learning more complex sequences of symbols, for instance license plates.

4. Underfitting: Due to the fact that the dataset used for training is rather small, it might be crucial to adjust the model by adding more layers so that it facilitates the model's learning ability.