Homework4: License Plate Localization

Description

For each image, there is one license plate. You are asked to localize the 4 corners of the license plate. That is, predict the (x, y) of each corner, 8 values in total. To reduce difficulties, you can fill in the blank of this reference code to run the code.

For example:


Ground-truth are drawn in orange. Predictions are drawn in red.

Dataset

Download the dataset from Kaggle. (You can download it by using Colab)

Dataset is organized as:

```
ccpd6000/
    train_images/
    test_images/
    train.csv
    sample.csv
```

There are 3000 images with annotation for training, 3000 images without label for testing. All images are taken from CCPD.

Each row in **train.csv** has following fields:

1. **name** specifies the name of the image, full path is ccpd6000/train_images/
2. **BR_x, BR_y** is the position of bottom-right corner

3. **BL_x, BL_y** is the position of bottom-left corner
4. **TL_x, TL_y** is the position of top-left corner
5. **TR_x, TR_y** is the position of top-right corner

The origin is at the top-left of the image

sample.csv serves as a sample submission. Your submission should have the same format as sample.csv. Note that **name** is sorted in alphabetical order

Evaluation

The metric is the root mean-square error between the predicted locations and the ground-truth locations of the 3000 testing images:

The metric is the root mean-square error between the predicted locations and the ground-truth locations of the 3000 testing images:

$$RMSE = \sqrt{\frac{1}{4N}\sum_{i=1}^{N}\sum_{j=1}^{4}\|\mathbf{p}_i^j - \hat{\mathbf{p}}_i^j\|^2}$$

where:

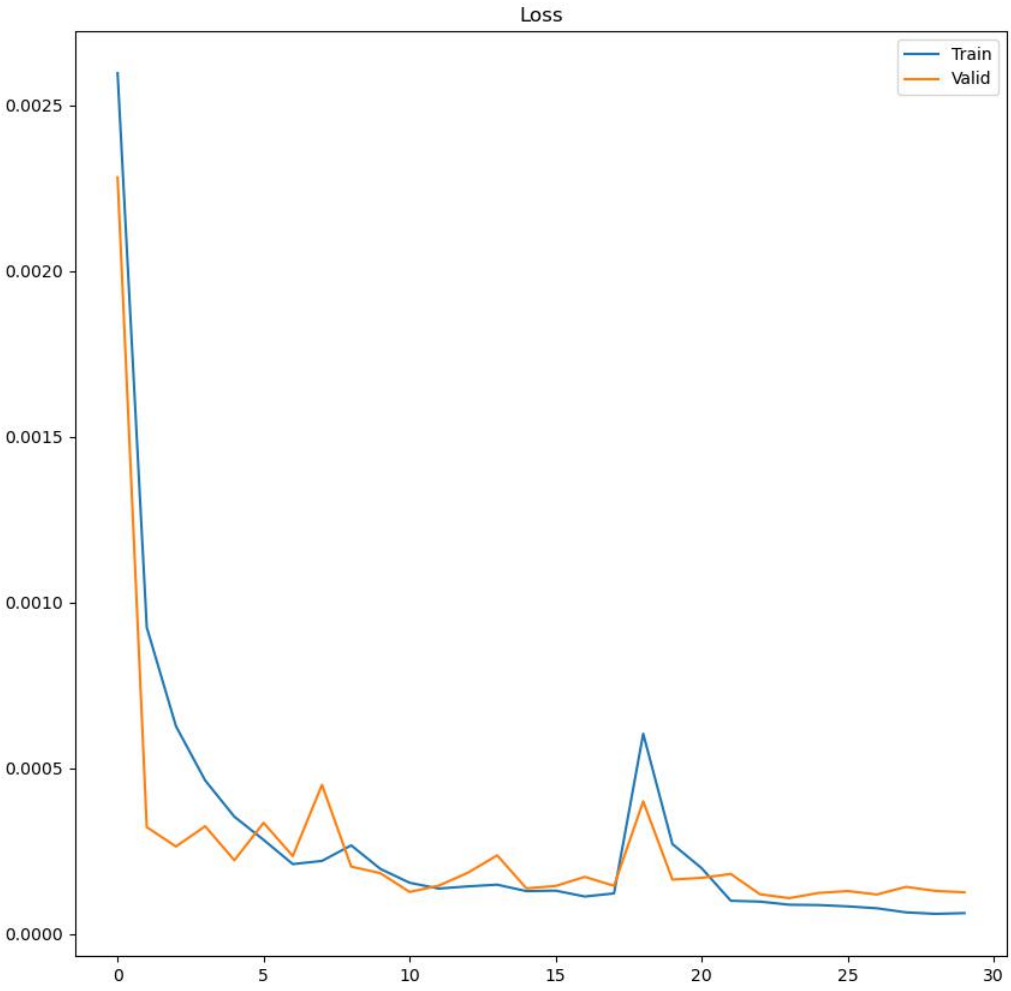
N is the number of images,
j is the index of the corner,
$\mathbf{p}_i^j$ is the predicted location (x, y) of the j-th corner of image i.
$\hat{\mathbf{p}}_i^j$ is the ground-truth location (x, y) of the j-th corner of image i.

Please submit your result to Kaggle to evaluate the performance. Your team name on kaggle should be studentID_name (ex: 111062552_黃偉哲)

Homework

1. Coding:
    - Please finish "TODO" part in provided reference code.
    - Pytorch guideline please reference pytorch document.
    - When using Colab, remember to change "Runtime Type" to "GPU" to accelerate training. For more information, please visit here.
2. Report: The report should be written in Jupyter Notebooks using Markdown cells. Please download your notebook and submit it to eeclass. Your report should include but not be limited to the following:

- Overlay training losses and validation losses in the same figure against each epoch. Ex:



- Visualization results of your final model on: (1)test_images/0000.jpg, (2)test_images/1000.jpg. Ex:

- The number of total parameters and trainable parameters. Ex: 11193192 (~11M) total parameters, and 11193192 (~11M) trainable parameters.
- The hyperparameters(epochs, learning rate, loss function, optimizer) of your final model.
- Describe and compare the methods you have tried in this project.

Score

You will be evaluated by comparing your Submission CSV to the ground truth Solution CSV by the RMSE Score.

Final score: coding(70 point) + report(30 point)

The coding score:

- Higher than baseline (70 point): 70 point
- Higher than baseline (65 point): 65 point
- Higher than baseline (60 point): 60 point
- Higher than baseline (50 point): 50 point
- Lower than baseline (50 point): 30 point
- Didn't submit and finish the code: Get 0 point

Please remember to write a report about your method, or your report score will be 0.

Hint

Possible ways to improve:

1. LR(learning rate) decay or smaller LR.
2. Train longer (typically until the validation loss is converged).
3. Use deeper model, like ResNet18, to extract features. (You can use pretrained weight.)
4. Different optimizer, loss, etc.
5. Data augmentation.