"""
Author: Hao Zheng

Exercise 5
"""

This files gives a short overview about my files, my model architecture, etc.

My best performing model is also uploaded as pt file, it achieved a RMSE of 17.894 on the challenge server. Due to my hardware(no GPU) and limited access to Google Colab I was not able to try out more CNN variations with different layers, but my relativly simple CNNs already achieved decent results on the server.

I tried out different variations with different number of training images, number of updates, training batch sizes; The last and best performing model has 6 instead of 5 hidden layers which improved the score a bit more.

Files:
Python:
- train_model.py
- test_model.py
- model_architecture.py
- datasets.py
- utils.py

Folders:
- results
- training

Others:
- model_logs
- working_config.json

1. train_model.py: File for training the model, mainly based on the main file from the example project and adjusted to our impainting challenge. Different model parameters can be adjusted in working_config.json.
To run train_model.py : python3 train_model.py working_config.json

2. test_model.py : File for testing the trained model on the provided test images in inputs.pkl. The model must be saved in the "results" folder.
To run test_model.py : python3 test_model.py Name_Of_Model
E.g. Name_of_Model = best_model.pt

3. model_architecture.py: Modified version of the architecture of the example project: For my model, I used the provided model architecture of the examples project and adjusted it to our challenge.

The input channels are 4(RGB values and the known array) and output channels are 3(only predicted RGB for whole images)
The activation function is the ReLU function.

The default CNN has : 5 hidden layers, 64 kernels, kernel size 3(can be adjusted in working_config.json, e.g. more hidden layers)

4. datasets.py: Classes used for our datasets/loader and the provided function from ex4 to create input array, known arrray and target arrays for training data
The number of images used for training/validation/test data can be adjusted here, since I do not have the resources to use the whole image set (~ 29k images),
I used only a smaller number of images for training my models.

5. utils.py: Contains different functions used in other scripts - e.g.plotting function, some functions were taken from the example project.

6. results: In this folder the results from model training are saved, in particular plots, tensorboard files and the model itself as pt file. I included some plots from the best performing model, the best model and its tensorboard files.

7. training: contains all training images, roughly 29 000 in total.

8. model_logs: Text file to keep track of differently trained models, their losses and their score on the challenge server.

9. working_config.json: JSON file to specify model parameters (n hidden layers, n kernels, kernel size, n updates) for training the model.

I used parts of code from:

- example project
- Unit 5 Data Loading