Data

Dataset

- Load data
- Apply normalization and transformations
- Return PyTorch Tensor

Examples:
- data/paired_image_dataset.py
- data/realesrgan_dataset.py

Utilities

We also provide some useful functions for data processing, such as color space conversion, degradation modeling, MATLAB functions, etc.

Examples:
- utils/color_util.py
- utils/matlab_functions.py
- data/degradations.py

Model

Model

- Define model-level behaviors, such as setup, feed data, training policy, validation, and recording.

Examples:
- models/sr_model.py
- models/edvr_model.py

Architecture

- Define the specific network architectures.

Examples:
- archs/srresnet_arch.py
- archs/rrdbnet_arch.py

Loss

- Define loss functions.

Examples:
- losses/losses.py

Option/Config

Training Options

- Define the training configurations.

In *options/train*

Examples:
- ESRGAN/train_ESRGAN_x4.yml
- EDVR/train_EDVR_L_x4_SR_REDS.yml
- BasicVSR/train_BasicVSR_REDS.yml

Testing Options

- Define the testing configurations.

In *options/test*

Examples:
- ESRGAN/test_ESRGAN_x4.yml
- EDVR/test_EDVR_L_x4_SR_Vid4.yml
- BasicVSR/test_BasicVSR_REDS.yml

Training

Training/Testing Loop

- Define the main training/testing loop.
- Defined in *train.py* and *test.py.*

Optimizers

- Define the optimizers.
- Defined in each model file.

Schedulers

- Define the learning rate schedulers.
- Defined in *models/base_model.py*.

Logger

- Tools to trace the training process.

Now support: (see *utils/logger.py*)
- Text logger
- Tensorboard logger
- Wandb logger