DeepHQ: Learned Hierarchical Quantizer for Progressive Deep Image Coding

This repository contains the source codes of the paper
"DeepHQ: Learned Hierarchical Quantizer for Progressive Deep Image Coding".

Environment Setup

Instead of manually installing required packages, you can simply build and run the provided Docker image.

```
cd docker
docker build -t deephq .
docker run --gpus all -it --rm -v $PWD:/workspace deephq /bin/bash
```

This will prepare all dependencies automatically.

Test (with pretrained checkpoint)

You can evaluate the model with the following command:

```
python eval_DeepHQ.py \
  --checkpoint (CHECKPOINT_PATH) \
  --data (TESTSET_PATH) \
  --cuda
```

- `(CHECKPOINT_PATH)`: Path to the pretrained checkpoint
- `(TESTSET_PATH)`: Directory that contains the test images (e.g., Kodak dataset PNGs)

👉 Pretrained checkpoint can be downloaded from here.

Training

DeepHQ training consists of **three phases**, where each phase uses the model trained from the previous phase as a pretrained model.

- **Phase 1**: Train the base model with MSE loss
  - Pretrained model for initialization:
    Download the TCM model (N=128, λ=0.05, MSE optimized) from the official repo:
    LIC_TCM GitHub

```
CUDA_VISIBLE_DEVICES='0' \
python -u (PHASE1_DIR)/train_phase1.py \
  -d (DATASET_DIR) \
  --cuda --N 128 --lambda 0.2 \
  --epochs 40 --lr_epoch 36 38 \
  --save_path (PHASE1_CHECKPOINT_DIR) \
  --batch-size 8 \
  --checkpoint (PRETRAINED_TCM_CHECKPOINT) \
  --save --num-workers 10
```

- **Phase 2**: Train the intermediate DeepHQ model

```
CUDA_VISIBLE_DEVICES='0,1' \
torchrun --nproc-per-node=2 --master-port=29501
(PHASE2_DIR)/train_phase2_ddp.py \
  -d (DATASET_DIR) \
  --cuda --N 128 --lambda 0.05 \
  --epochs 20 \
  --save_path (PHASE2_CHECKPOINT_DIR) \
  --batch-size 4 \
  --checkpoint (PHASE1_CHECKPOINT_DIR)/0.05checkpoint_latest.pth.tar \
  --save --num-workers 4
```

- **Phase 3**: Train the full DeepHQ model

```
CUDA_VISIBLE_DEVICES='2,3' \
torchrun --nproc-per-node=2 --master-port=29500
(PHASE3_DIR)/train_phase3_ddp.py \
  -d (DATASET_DIR) \
  --cuda --N 128 --lambda 0.05 \
  --epochs 20 \
  --save_path (PHASE3_CHECKPOINT_DIR) \
  --batch-size 4 \
  --checkpoint (PHASE2_CHECKPOINT_DIR)/checkpoint_latest.pth.tar \
  --save --num-workers 4
```

Notes

- `(DATASET_DIR)` should contain the training dataset (e.g., CLIC).
- `(PHASE1_DIR)`, `(PHASE2_DIR)`, `(PHASE3_DIR)` indicate the directories where the training scripts for each phase are located.
- `(PHASE1_CHECKPOINT_DIR)`, `(PHASE2_CHECKPOINT_DIR)`, `(PHASE3_CHECKPOINT_DIR)` are the output checkpoint directories for each training phase.
- `(PRETRAINED_TCM_CHECKPOINT)` is the pretrained TCM model checkpoint (N=128, λ=0.05, MSE optimized).

- Each phase continues from the checkpoint of the previous phase.