一、HAT: Hybrid Attention Transformer for Image Restoration

github 链接：https://github.com/xpixelgroup/hat

配置过程（主要参考 README）：

1. 从 github 下载该项目，可以直接下载，也可使用命令
   git clone https://github.com/XPixelGroup/HAT.git
   进入文件夹 cd HAT

2. 可以为该项目创建新的 conda 环境或直接在 base 环境中配置
   conda 创建环境自行百度
3. 安装相关包，注意 pytorch 安装容易出错，要使用对应的 cuda 版本，可进入 pytorch 官网使用对应命令安装

```
pip install -r requirements.txt
python setup.py develop
```

4. 参考 github 页面给出的链接准备数据集

   https://github.com/XPixelGroup/BasicSR/blob/master/docs/DatasetPreparation.md

5. 测试

（1）下载预训练模型：
https://pan.baidu.com/share/init?surl=u2r4Lc2_EEeQqra2-w85Xg 验证码：qyrl

（2）测试命令：

```
python hat/test.py -opt options/test/HAT_SRx4_ImageNet-pretrain.yml
```

（3）测试结果保存在./results 中

（4）自定义数据集测试替换为：`./options/test/HAT_SRx4_ImageNet-LR.yml`，修改 dataroot_lq 属性值为自定义数据集所在文件夹位置。

HAT / options / test / HAT_SRx4_ImageNet-LR.yml

chxy95 add tile mode

Code Blame 48 lines (42 loc) · 949 Bytes

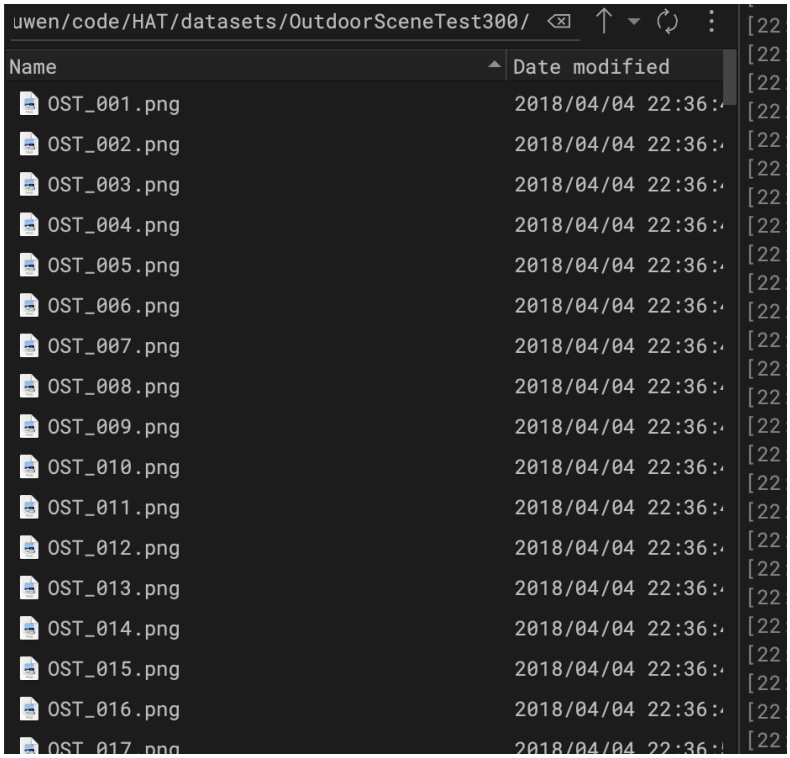
```yaml
name: HAT_SRx4_ImageNet-LR
model_type: HATModel
scale: 4
num_gpu: 1  # set num_gpu: 0 for cpu mode
manual_seed: 0

tile:
  tile_size: 512 # max patch size for the tile mode
  tile_pad: 32

datasets:
  test_1:  # the 1st test dataset
    name: custom
    type: SingleImageDataset
    dataroot_lq: input_dir
    io_backend:
      type: disk
```

例如

```yaml
dataroot_lq: ./datasets/OutdoorSceneTest300
```



6. 训练

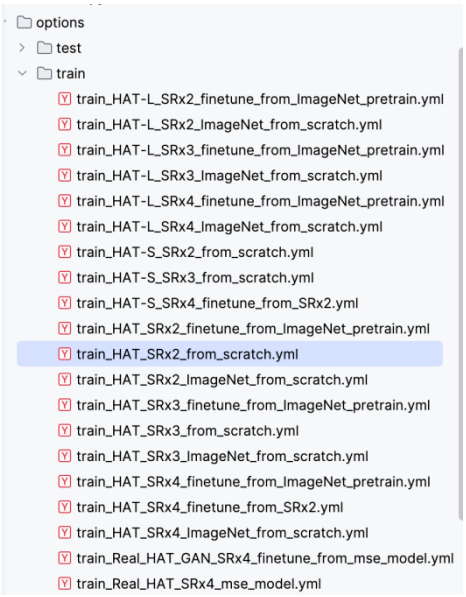
（1）训练命令：根据不同环境选择每个节点的可用数量以及 gpu 编号

```
CUDA_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 python -m torch.distributed.launch --nproc_per_node=8 --m
```

（2）训练日志和权重保存在./experiments 文件夹下

从头训练：

（1）配置参考该文件（包括预训练模型加载，数据集配置均在此处修改）



（2）使用 DIV2K 数据集进行训练，数据准备参考 https://github.com/XPixelGroup/BasicSR/blob/master/docs/DatasetPreparation.md

DIV2K

DIV2K is a widely-used dataset in image super-resolution. In many research works, a MATLAB bicubic downsampling kernel is assumed. It may not be practical because the MATLAB bicubic downsampling kernel is not a good approximation for the implicit degradation kernels in real-world scenarios. And there is another topic named *blind restoration* that deals with this gap.

Preparation Steps

1. Download the datasets from the official DIV2K website.
2. Crop to sub-images: DIV2K has 2K resolution (e.g., 2048 × 1080) images but the training patches are usually small (e.g., 128x128 or 192x192). So there is a waste if reading the whole image but only using a very small part of it. In order to accelerate the IO speed during training, we crop the 2K resolution images to sub-images (here, we crop to 480x480 sub-images).
   Note that the size of sub-images is different from the training patch size (`gt_size`) defined in the config file. Specifically, the cropped sub-images with 480x480 are stored. The dataloader will further randomly crop the sub-images to `GT_size x GT_size` patches for training.
   Run the script extract_subimages.py:

   ```
   python scripts/data_preparation/extract_subimages.py
   ```

   Remember to modify the paths and configurations if you have different settings.
3. [Optional] Create LMDB files. Please refer to LMDB Description. `python scripts/data_preparation/create_lmdb.py` . Use the `create_lmdb_for_div2k` function and remember to modify the paths and configurations accordingly.

Data access

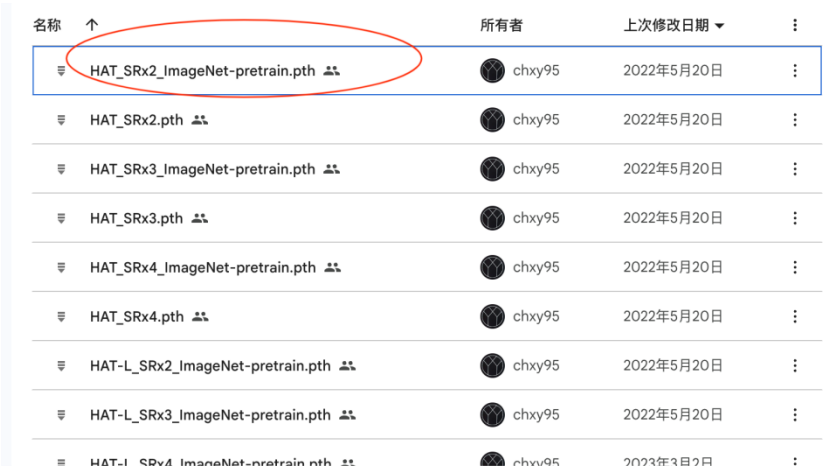
- (NTIRE 2017) Low Res Images:
  - Train Data Track 1 bicubic downscaling x2 (LR images)
  - Train Data Track 2 unknown downgrading operators x2 (LR images)
  - Validation Data Track 1 bicubic downscaling x2 (LR images)
  - Validation Data Track 2 unknown downgrading operators x2 (LR images)
  - Train Data Track 1 bicubic downscaling x3 (LR images)
  - Train Data Track 2 unknown downgrading operators x3 (LR images)
  - Validation Data Track 1 bicubic downscaling x3 (LR images)
  - Validation Data Track 2 unknown downgrading operators x3 (LR images)
  - Train Data Track 1 bicubic downscaling x4 (LR images)
  - Train Data Track 2 unknown downgrading operators x4 (LR images)
  - Validation Data Track 1 bicubic downscaling x4 (LR images)
  - Validation Data Track 2 unknown downgrading operators x4 (LR images)
- (NTIRE 2018) Low Res Images:
  - Train Data Track 1 bicubic x8 (LR images)
  - Train Data Track 2 realistic mild x4 (LR images)
  - Train Data Track 3 realistic difficult x4 (LR images)
  - Train Data Track 4 realistic wild x4 (LR images)
  - Validation Data Track 1 bicubic x8 (LR images)
  - Validation Data Track 2 realistic mild x4 (LR images)
  - Validation Data Track 3 realistic difficult x4 (LR images)
  - Validation Data Track 4 realistic wild x4 (LR images)
- High Resolution Images:
  - Train Data (HR images)
  - Validation Data (HR images)

在官方给的权重上进行细调：

（1）由于官方没有提供 imagenet 预训练的模型权重，直接使用官网在 imagenet 上细调后的权重文件，从这里下载 https://github.com/XPixelGroup/HAT

（见 how to test 下面给的百度网盘链接）

（2）选择这个文件下载：



（3）加载权重文件，初始化网络参数，可以参考这个博客：

https://blog.csdn.net/qq_42698422/article/details/100547225

（4）模型初始化后，就按照从头训练网络的步骤，在 DIV2K 数据集上训练网络