ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks

Neelu Verma
MP19AI002
Indian Institute of Technology
Jodhpur

1 Empirical Analysis

This paper proposed ESRGAN model that achieves consistently better perceptual quality than previous SR methods

1. Required dependencies: python==3.7, tensorflow-gpu==2.1.0, CUDA10.1, numpy, opencv-python, PyYAML, tqdm, Pillow

2. Dataset:
training data:DIV2k and Flickr2K where Low Resolution images for training : DIV2K_train_LR_
Ground-Truth Hight Resolution Images: DIV2K_train_HR
Test dataset: Set5, Set14, BSD100 Set14
Subimages newly created dataset for Interpolation:
DIV2K800_sub_bicLRx4 (LR), DIV2K800_sub (GT)

3.Hyperparameter setting:
batch_size: 16, input_size: 32, gt_size: 128, ch_size: 3, scale: 4,

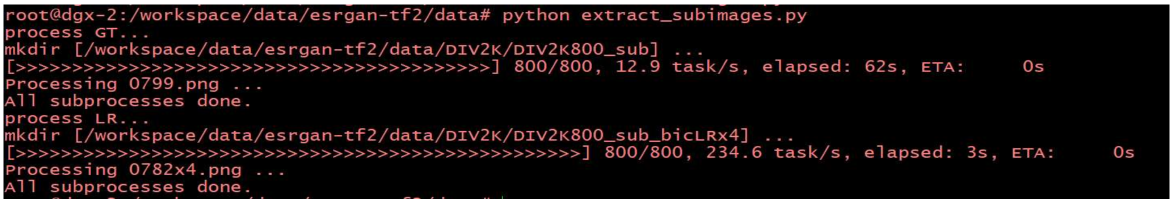
4.Models: Models are trained in RGB channels, and the training dataset is augmented with random horizontal flips and 90-degree rotations.

2 Qualitative Results

2.1 Running rename.py file to rename image file in LR folder DIV2K_train_LR_bicubic

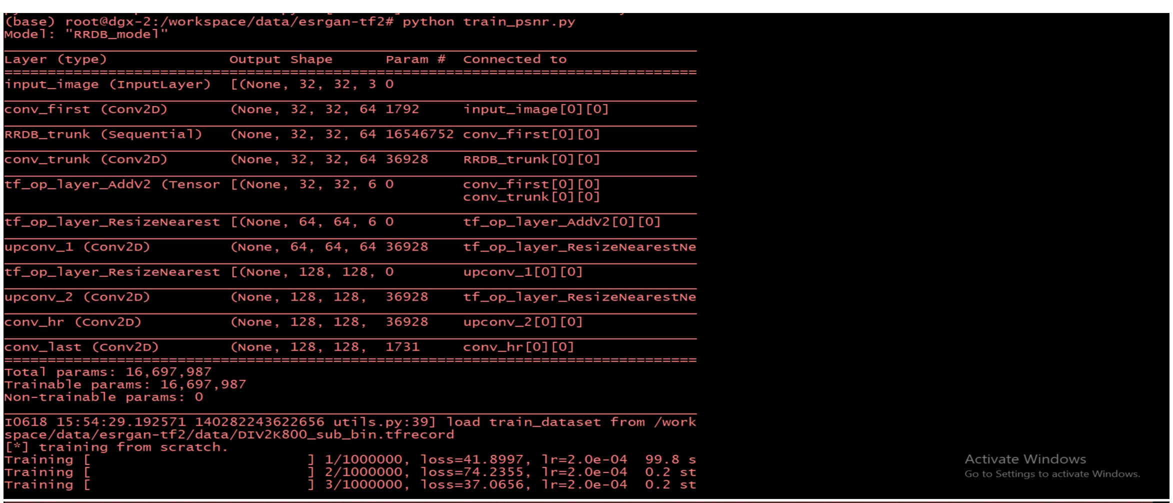
```
root@dgx-2:/workspace/data/esrgan-tf2/data# python rename.py
Finished.
```

2.2 Running extract_subimages.py file to extract sub-images from HR folder and LR folder.

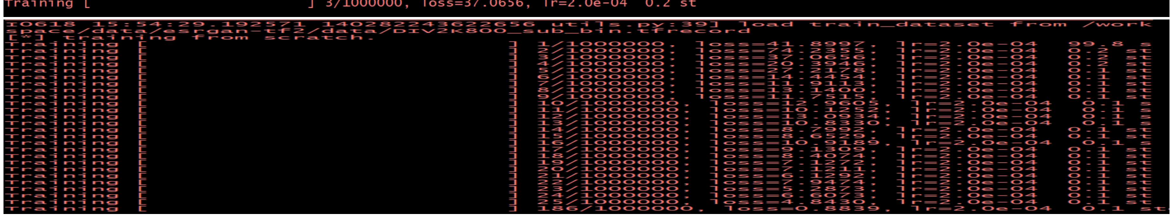


```
root@dgx-2:/workspace/data/esrgan-tf2/data# python extract_subimages.py
process GT...
mkdir [/workspace/data/esrgan-tf2/data/DIV2K/DIV2K800_sub] ...
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 800/800, 12.9 task/s, elapsed: 62s, ETA:     0s
Processing 0799.png ...
All subprocesses done.
process LR...
mkdir [/workspace/data/esrgan-tf2/data/DIV2K/DIV2K800_sub_bicLRx4] ...
[>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>] 800/800, 234.6 task/s, elapsed: 3s, ETA:     0s
Processing 0782x4.png ...
All subprocesses done.
```

2.3 Running Train_psnr.py to train the PSNR RDDB model by Scratch took 2 days to train on DGX-2.



```
(base) root@dgx-2:/workspace/data/esrgan-tf2# python train_psnr.py
Model: "RRDB_model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_image (InputLayer)        [(None, 32, 32, 3 0
__________________________________________________________________________________________________
conv_first (Conv2D)             (None, 32, 32, 64 1792        input_image[0][0]
__________________________________________________________________________________________________
RRDB_trunk (Sequential)         (None, 32, 32, 64 16546752    conv_first[0][0]
__________________________________________________________________________________________________
conv_trunk (Conv2D)             (None, 32, 32, 64 36928       RRDB_trunk[0][0]
__________________________________________________________________________________________________
tf_op_layer_AddV2 (Tensor       [(None, 32, 32, 6 0           conv_first[0][0]
                                                              conv_trunk[0][0]
__________________________________________________________________________________________________
tf_op_layer_ResizeNearest       [(None, 64, 64, 6 0           tf_op_layer_AddV2[0][0]
__________________________________________________________________________________________________
upconv_1 (Conv2D)               (None, 64, 64, 64 36928       tf_op_layer_ResizeNearestNe
__________________________________________________________________________________________________
tf_op_layer_ResizeNearest       [(None, 128, 128, 0           upconv_1[0][0]
__________________________________________________________________________________________________
upconv_2 (Conv2D)               (None, 128, 128,  36928       tf_op_layer_ResizeNearestNe
__________________________________________________________________________________________________
conv_hr (Conv2D)                (None, 128, 128,  36928       upconv_2[0][0]
__________________________________________________________________________________________________
conv_last (Conv2D)              (None, 128, 128,  1731        conv_hr[0][0]
==================================================================================================
Total params: 16,697,987
Trainable params: 16,697,987
Non-trainable params: 0
__________________________________________________________________________________________________
I0618 15:54:29.192571 140282243622656 utils.py:39] load train_dataset from /work
space/data/esrgan-tf2/data/DIV2K800_sub_bin.tfrecord
[*] training from scratch.
Training [                              ] 1/1000000, loss=41.8997, lr=2.0e-04  99.8 s
Training [                              ] 2/1000000, loss=74.2355, lr=2.0e-04  0.2 st
Training [                              ] 3/1000000, loss=37.0656, lr=2.0e-04  0.2 st
```

2.4 Running train_esrgan.py to train the ESRGAN model with the pretrain PSNR model 5 days on DGX-2

```
(base) root@dgx-2:/workspace/data/esrgan-tf2# python train_esrgan.py --cfg_path="./configs/esrgan.yaml" --gpu=0
Model: "RRDB_model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_image (InputLayer)     [(None, 32, 32, 3 0
_________________________________________________________________
conv_first (Conv2D)          (None, 32, 32, 64 1792      input_image[0][0]
_________________________________________________________________
RRDB_trunk (Sequential)      (None, 32, 32, 64 16546752  conv_first[0][0]
_________________________________________________________________
conv_trunk (Conv2D)          (None, 32, 32, 64 36928     RRDB_trunk[0][0]
_________________________________________________________________
tf_op_layer_AddV2 (Tensor    [(None, 32, 32, 6 0         conv_first[0][0]
                                                         conv_trunk[0][0]
_________________________________________________________________
tf_op_layer_ResizeNearest    [(None, 64, 64, 6 0         tf_op_layer_AddV2[0][0]
_________________________________________________________________
upconv_1 (Conv2D)            (None, 64, 64, 64 36928     tf_op_layer_ResizeNearestNe
_________________________________________________________________
tf_op_layer_ResizeNearest    [(None, 128, 128, 0         upconv_1[0][0]
_________________________________________________________________
upconv_2 (Conv2D)            (None, 128, 128,  36928     tf_op_layer_ResizeNearestNe
_________________________________________________________________
conv_hr (Conv2D)             (None, 128, 128,  36928     upconv_2[0][0]
_________________________________________________________________
conv_last (Conv2D)           (None, 128, 128,  1731      conv_hr[0][0]
=================================================================
Total params: 16,697,987
Trainable params: 16,697,987
Non-trainable params: 0
_________________________________________________________________
Model: "Discriminator_VGG_128"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 128, 128, 3)]     0
_________________________________________________________________
linear2 (Dense)              (None, 1)                 101
=================================================================
Total params: 14,505,161
Trainable params: 14,499,401
Non-trainable params: 5,760
_________________________________________________________________
I0619 07:44:24.506086 140022443017984 utils.py:39] load train_dataset from /workspace/data/esrgan-tf2/data/DIV2K800_sub_bin.tfrecord
[*] training from pretrain model ./checkpoints/psnr_pretrain.
Training [                       ] 151/400000, loss_G=1.0536, loss_D=0.0019, lr_G=1.0e-04, lr_D=1.0e-04  0.1 step/sec
Training [                       ] 152/400000, loss_G=1.3132, loss_D=0.1490, l
Training [                       ] 153/400000, loss_G=1.2330, loss_D=0.0431, lr_
Training [                       ] 154/400000, loss_G=1.1995, loss_D=0.1376, lr_G=1.0e
Training [                       ] 155/400000, loss_G=1.1946, loss_D=0.0075, lr_G=1.0e
Training [                       ] 156/400000, loss_G=1.0302, loss_D=0.0431, lr_G=1.0e
Training [                       ] 157/400000, loss_G=1.1923, loss_D=0.0943, lr_G=1.0e
Training [                       ] 158/400000, loss_G=1.1098, loss_D=0.0046, lr_G=1.0e-04, lr_D=1.0e-04  0.1 step/se
Training [                       ] 159/400000, loss_G=1.2056, loss_D=0.0298, lr_G=1.0e-04, lr_D=1.0e-04  0.1 step/se
```


2.5 Test and Result of PSNR model on Set5 and Set14.The corresponding images stored in Result folder.

```
(base) root@dgx-2:/workspace/data/esrgan-tf2# python test.py --cfg_path="./configs/psnr.yaml"
[*] load ckpt from ./checkpoints/psnr_pretrain/ckpt-1.
[*] Processing on Set5 and Set14, and write results
'set5_path' form /workspace/data/esrgan-tf2/data/Set5
  PSNR/SSIM
  [head.png            ] Bic=32.01db/0.76, SR=31.56db/0.78
  [bird.png            ] Bic=30.27db/0.87, SR=31.32db/0.89
  [woman.png           ] Bic=26.44db/0.83, SR=27.44db/0.87
  [baby.png            ] Bic=31.96db/0.85, SR=31.53db/0.88
  [butterfly.png       ] Bic=22.25db/0.72, SR=23.98db/0.79
'set14_path' form /workspace/data/esrgan-tf2/data/Set14
  PSNR/SSIM
  [barbara.png         ] Bic=25.19db/0.69, SR=25.40db/0.73
  [monarch.png         ] Bic=27.60db/0.88, SR=29.11db/0.90
  [face.png            ] Bic=31.98db/0.76, SR=31.47db/0.78
  [pepper.png          ] Bic=29.38db/0.84, SR=29.10db/0.84
  [man.png             ] Bic=25.74db/0.68, SR=26.17db/0.72
  [baboon.png          ] Bic=22.06db/0.45, SR=22.09db/0.49
  [foreman.png         ] Bic=27.65db/0.86, SR=27.24db/0.88
  [coastguard.png      ] Bic=25.33db/0.52, SR=25.15db/0.56
  [ppt3.png            ] Bic=21.76db/0.82, SR=23.10db/0.87
  [bridge.png          ] Bic=24.38db/0.56, SR=24.46db/0.62
  [lenna.png           ] Bic=29.67db/0.80, SR=30.17db/0.82
  [zebra.png           ] Bic=24.15db/0.68, SR=25.73db/0.74
  [flowers.png         ] Bic=25.85db/0.72, SR=26.53db/0.76
  [comic.png           ] Bic=21.69db/0.59, SR=22.14db/0.65
[*] write the visual results in ./results/psnr_pretrain/
(base) root@dgx-2:/workspace/data/esrgan-tf2#
```


Result of Set5 using PSNR model, Three images showing Low Resolution Bicubic(Bic) image, PSNR Super Resolution(SR) image, Ground Truth respectively

Result of Set14 using PSNR model, Three images showing Low Resolution Bicubic(Bic) image, PSNR Super Resolution(SR) image, Ground Truth respectively