

How to use the example inpainting code in test mode

By Mike Wang

Project GitHub Link:

<https://github.com/mikewang928/generative-inpainting-pytorch>

Pre-requisite:

- Python3
- PyTorch 1.0+
- torchvision 0.2.0+
- tensorboardX
- pyyaml (5.4.1) Important, version 5.6+ will result in additional Loader in the field of load() function. To check your pyyaml version in your terminal *pip show pyyaml*

Steps:

- ⇒ **Step 1:** Download the latest trained network model from (<https://drive.google.com/drive/folders/1qbfA5BP9yzdTFFmiOTvYARUYgW1zwBBK>) on your local directory as *generative-inpainting-pytorch/checkpoints/imagenet/hole_benchmark*
- ⇒ **Step 2:** check if you have a GPU, if you have a GPU (with cuda) skip to Step 3. Else, change the config.yaml file from *generative-inpainting-pytorch/configs/config.yaml* as

```
# data parameters
dataset_name: imagenet
data_with_subfolder: True
train_data_path: /media/ouc/4T_A/datasets/ImageNet/ILSVRC2012_img_train/
val_data_path:
resume:
batch_size: 48
image_shape: [256, 256, 3]
mask_shape: [128, 128]
mask_batch_size: True
max_delta_shape: [32, 32]
margin: [0, 0]
discounted_mask: True
spatial_discounting_gamma: 0.9
random_crop: True
mask_type: hole # hole | mosaic
mosaic_unit_size: 12

# training parameters
exname: benchmark
# cuda: True
cuda: False
gpu_ids: [0, 1, 2] # set the GPU ids to use, e.g. [0] or [1, 2]
num_workers: 4
lr: 0.0001
betal: 0.5
beta2: 0.9
n_critic: 5
niter: 50000
print_iter: 100
viz_iter: 1000
viz_max_out: 16
snapshot_save_iter: 5000

# loss weight
coarse_l1_alpha: 1.2
l1_loss_alpha: 1.2
ae_loss_alpha: 1.2
global_wgan_loss_alpha: 1.
gan_loss_alpha: 0.001
wgan_gp_lambda: 10

# network parameters
netG:
  input_dim: 3
  ngf: 32

netD:
  input_dim: 3
  ndf: 64
```

Also change the *test_single.py* as:

```
73 elif args.mask:
74     raise TypeError("{} is not an image file.".format(args.mask))
75 else:
76     # Test a single ground-truth image with a random mask
77     ground_truth = default_loader(args.image)
78     ground_truth = transforms.Resize(config['image_shape'][::-1])(ground_truth)
79     ground_truth = transforms.CenterCrop(config['image_shape'][::-1])(ground_truth)
80     ground_truth = transforms.ToTensor()(ground_truth)
81     ground_truth = normalize(ground_truth)
82     ground_truth = ground_truth.unsqueeze(dim=0)
83     bboxes = random_bbox(config, batch_size=ground_truth.size(0))
84     x, mask = mask_image(ground_truth, bboxes, config)
85
86 # Set checkpoint path
87 if not args.checkpoint_path:
88     checkpoint_path = os.path.join('checkpoints',
89                                     config['dataset_name'],
90                                     config['mask_type'] + '_' + config['expname'])
91 else:
92     checkpoint_path = args.checkpoint_path
93
94 # Define the trainer
95 netG = Generator(config['netG'], cuda, device_ids)
96 # Resume weight
97 last_model_name = get_model_list(checkpoint_path, "gen", iteration=args.iter)
98 print("-----")
99 print(last_model_name)
100 print("-----")
101 netG.load_state_dict(torch.load(last_model_name, map_location=torch.device('cpu')))
102 model_iteration = int(last_model_name[-11:-3])
103 print("Resume from {} at iteration {}".format(checkpoint_path, model_iteration))
104
105 if cuda:
106     netG = nn.parallel.DataParallel(netG, device_ids=device_ids)
107     x = x.cuda()
108     mask = mask.cuda()
109
110 # Inference
111 x1, x2, offset_flow = netG(x, mask)
112 inpainted_result = x2 * mask + x * (1. - mask)
113
114 utils.save_image(inpainted_result, args.output, padding=0, normalize=True)
115 print("Saved the inpainted result to {}".format(args.output))
116 if args.flow:
117     utils.save_image(offset_flow, args.flow, padding=0, normalize=True)
118     print("Saved offset flow to {}".format(args.flow))
119 else:
120     raise TypeError("{} is not an image file.".format)
121 # exit no grad context
122 except Exception as e: # for unexpected error logging
123     print("Error: {}".format(e))
124     raise e
125
126 if __name__ == '__main__':
127     main()
128
129
```

This is to make sure that your torch is working in cpu mode.

Step 4: In terminal cd to the *generative-inpainting-pytorch* directory and then run
python test_single.py \

```
--image examples/imagenet/imagenet_patches_ILSVRC2012_val_00008210_input.png \
--mask examples/center_mask_256.png \
--output examples/output.png
```