



Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

[Read the guide](#)

 [milesial](#) / [Pytorch-UNet](#)

[Code](#) [Issues 35](#) [Pull requests 1](#) [Discussions](#) [Actions](#) [Security](#) [Insights](#)

[master](#) [Pytorch-UNet / utils / dataset.py](#) [Jump to](#)

[Go to file](#)



 [milesial](#) Carvana dataset loader

Latest commit 4ad8323 on Jul 30, 2020 [History](#)

[2 contributors](#)  

71 lines (55 sloc) | 2.27 KB

[Raw](#) [Blame](#)   

```
1 from os.path import splitext
2 from os import listdir
3 import numpy as np
4 from glob import glob
5 import torch
6 from torch.utils.data import Dataset
7 import logging
8 from PIL import Image
9
10
11 class BasicDataset(Dataset):
12     def __init__(self, imgs_dir, masks_dir, scale=1, mask_suffix=''):
13         self.imgs_dir = imgs_dir
14         self.masks_dir = masks_dir
15         self.scale = scale
16         self.mask_suffix = mask_suffix
17         assert 0 < scale <= 1, 'Scale must be between 0 and 1'
18
19         self.ids = [splitext(file)[0] for file in listdir(imgs_dir)
20                     if not file.startswith('.')]
21         logging.info(f'Creating dataset with {len(self.ids)} examples')
22
23     def __len__(self):
24         return len(self.ids)
25
26     @classmethod
27     def preprocess(cls, pil_img, scale):
28         w, h = pil_img.size
29         newW, newH = int(scale * w), int(scale * h)
30         assert newW > 0 and newH > 0, 'Scale is too small'
31         pil_img = pil_img.resize((newW, newH))
32
33         img_nd = np.array(pil_img)
34
35         if len(img_nd.shape) == 2:
36             img_nd = np.expand_dims(img_nd, axis=2)
37
38         # HWC to CHW
39         img_trans = img_nd.transpose((2, 0, 1))
40         if img_trans.max() > 1:
41             img_trans = img_trans / 255
42
43         return img_trans
44
45     def __getitem__(self, i):
46         idx = self.ids[i]
47         mask_file = glob(self.masks_dir + idx + self.mask_suffix + '.*')
48         img_file = glob(self.imgs_dir + idx + '.*')
```

```
49
50     assert len(mask_file) == 1, \
51         f'Either no mask or multiple masks found for the ID {idx}: {mask_file}'
52     assert len(img_file) == 1, \
53         f'Either no image or multiple images found for the ID {idx}: {img_file}'
54     mask = Image.open(mask_file[0])
55     img = Image.open(img_file[0])
56
57     assert img.size == mask.size, \
58         f'Image and mask {idx} should be the same size, but are {img.size} and {mask.size}'
59
60     img = self.preprocess(img, self.scale)
61     mask = self.preprocess(mask, self.scale)
62
63     return {
64         'image': torch.from_numpy(img).type(torch.FloatTensor),
65         'mask': torch.from_numpy(mask).type(torch.FloatTensor)
66     }
67
68
69 class CarvanaDataset(BasicDataset):
70     def __init__(self, imgs_dir, masks_dir, scale=1):
71         super().__init__(imgs_dir, masks_dir, scale, mask_suffix='_mask')
```