```python
import os
import numpy as np
import PIL.Image as Image
import torch
from torch.utils import data
import pdb
import random


class ImageFiles(data.Dataset):
    def __init__(self, root, crop=None, flip=False,
                 source_transform=None,
                 mean=None, std=None, training=False):
        super(ImageFiles, self).__init__()
        self.training = training
        self.mean, self.std = mean, std
        self.flip = flip
        self.s_transform, self.crop = source_transform, crop
        self.root = root
        names = os.listdir(root)
        self.img_filenames = list(map(lambda x: os.path.join(root, x), names))
        names = list(map(lambda x: '.'.join(x.split('.')[:-1]), names))
        self.names = names

    def __len__(self):
        return len(self.names)

    def __getitem__(self, index):
        # load image
        img_file = self.img_filenames[index]
        img = Image.open(img_file)
        name = self.names[index]
        if self.crop is not None:
            # random crop size of crop
            w, h = img.size
            th, tw = int(self.crop*h), int(self.crop*w)
            if w == tw and h == th:
                return 0, 0, h, w
            i = random.randint(0, h - th)
            j = random.randint(0, w - tw)
            img = img.crop((j, i, j + tw, i + th))
        if self.s_transform is not None:
            img = self.s_transform(img)
        WW, HH = img.size
        img = np.array(img, dtype=np.uint8)
        if len(img.shape) < 3:
            img = np.stack((img, img, img), 2)
        if img.shape[2] > 3:
            img = img[:, :, :3]
        if self.flip and random.randint(0, 1):
            img = img[:, ::-1].copy()
        img = img.astype(np.float64) / 255
        if self.mean is not None:
            img -= self.mean
        if self.std is not None:
            img /= self.std
        img = img.transpose(2, 0, 1)
        img = torch.from_numpy(img).float()
        if self.training:
```

```
60              return img
61          else:
62              return img, name, WW, HH
63
64
65  if __name__ == "__main__":
66      sb = ImageFiles('../../data/datasets/ILSVRC14VOC/images')
67      pdb.set_trace()
68
```