MRI 2D Image PNG Generator

December 12, 2018

0.0.1 Import all dependencies

In [1]:
```
from collections import defaultdict
import time
import os


import ipywidgets as widgets
from ipywidgets import interact, interactive, IntSlider, Layout, interact_manual,fixed


import numpy as np
import pandas as pd

from tqdm import tqdm

from medicaltorch import datasets as mt_datasets
from medicaltorch import models as mt_models
from medicaltorch import transforms as mt_transforms
from medicaltorch import losses as mt_losses
from medicaltorch import metrics as mt_metrics
from medicaltorch import filters as mt_filters

from skimage.io import imsave # for saving png images


import torch
from torchvision import transforms
from torch.utils.data import DataLoader
from torch import autograd, optim
import torch.backends.cudnn as cudnn
import torch.nn as nn

import torchvision.utils as vutils

cudnn.benchmark = True
import matplotlib.pyplot as plt
```

```
import matplotlib
%matplotlib inline
```

0.0.2 Much of the following code is based on the medical torch library

We save our data in the following manner:

–root

—-data/

————-mri/

————————/OAS30001_MR_d2430

————————————/anat4

————————————————————-sub-OAS30001_ses-d2430_T1w.nii.gz (this is the raw MRI scan)

————————————/anat2

————————/OAS30002_MR_d2430

————————/OAS30003_MR_d2430

In [2]:
```
from __future__ import print_function
import ipywidgets as widgets

listOfMRIFolders = [folder for folder in os.listdir('./data/mri') if folder.find(".zip")

ROOT_DIR_GMCHALLENGE = "./data/mri/OAS30001_MR_d2430/anat4/"
mri_input_filename = os.path.join(ROOT_DIR_GMCHALLENGE,
                                                'sub-OAS30001_ses-d2430_T1w.nii.gz')

pair = mt_datasets.SegmentationPair2D(mri_input_filename, None)
```

In [3]:
```
os.listdir('./data/')
df = pd.read_csv('./data/Patient Data_Last.csv')

def getClassification(label):
    return df.loc[df['Label'] == label, 'cdr'].iloc[0]


def getFirstMRIFilePath(label):

    firstAnat = None
```

```
        # just get first anat folder
        for file in os.listdir('data/mri/'+label):
            if file.find('anat') != -1:
                firstAnat = file
                break



        filesInAnat = os.listdir('data/mri/'+label+'/'+firstAnat)
        firstMRIFound = None
        for file in filesInAnat:
            if file.find('.json') == -1 and file.find('nii.gz') != -1:
                firstMRIFound = 'data/mri/'+label+'/'+firstAnat+'/'+file
                break
        return firstMRIFound


    def getAllMRIFilePaths(label):

        annats = []
        for file in os.listdir('data/mri/'+label):
            if file.find('anat') != -1:
                annats.append(file)
        print(annats)


        MRIs = []
        for annat in annats:
            filesInAnat = os.listdir('data/mri/'+label+'/'+annat)
            for file in filesInAnat:
                if file.find('.json') == -1 and file.find('nii.gz') != -1:
                    MRIs.append('data/mri/'+label+'/'+annat+'/'+file)
        return MRIs
```

0.1 Test if you can obtain all file paths of MRIs

```
In [4]: getAllMRIFilePaths('OAS30007_MR_d0061')

['anat3', 'anat2']


Out[4]: ['data/mri/OAS30007_MR_d0061/anat3/sub-OAS30007_ses-d0061_run-02_T1w.nii.gz',
         'data/mri/OAS30007_MR_d0061/anat2/sub-OAS30007_ses-d0061_run-01_T1w.nii.gz']
```

0.1.1 View a particular slice of an MRI according to patient ID (e.g OAS30007_MR_d0061)

```
In [5]: def view_slice(sliceNumber,MRILabel):


            pair = mt_datasets.SegmentationPair2D(getFirstMRIFilePath(MRILabel), None)

            slice_pair = pair.get_pair_slice(sliceNumber)
            input_slice = slice_pair["input"]
            img = input_slice
            #print(img[100])
            plt.imshow(img)
            plt.show()
            matplotlib.image.imsave('name.png', img)
            print(img.shape)


        p = interactive(view_slice, sliceNumber=(1,224),MRILabel=listOfMRIFolders)
        display(p)
```

```
interactive(children=(IntSlider(value=112, description='sliceNumber', max=224, min=1), Dropdown(
```

0.1.2 Visualize Patient Data in Table Form

Note, you must have a CSV file with patient information from the OASIS-3 dataset in the data directory

The code will output the unique cognitive decline ratings in the CSV

```
In [6]: import os
        import pandas as pd
        df = pd.read_csv('./data/Patient Data_Last.csv')
        [str(i) for i in df.cdr.unique()]
```

```
Out[6]: ['0.0', '0.5', '1.0', '2.0', '3.0']
```

```
In [7]: from PIL import Image
        import imageio


        def convertFileToGreyscale(path):
            img = Image.open(path)
            img = img.convert('L')
            imageio.imwrite(path, np.array(img))


        def generateImagesWithSlice(slice):
            folderToSaveIn = 'slice'+str(slice)
```

```
if folderToSaveIn not in os.listdir('./data/'):
    os.mkdir('./data/'+folderToSaveIn)
    os.mkdir('./data/'+folderToSaveIn+'/train')
    os.mkdir('./data/'+folderToSaveIn+'/train/00')
    os.mkdir('./data/'+folderToSaveIn+'/train/0.5-1.0')
    os.mkdir('./data/'+folderToSaveIn+'/train/>1.0')

    os.mkdir('./data/'+folderToSaveIn+'/test')
    os.mkdir('./data/'+folderToSaveIn+'/test/00')
    os.mkdir('./data/'+folderToSaveIn+'/test/0.5-1.0')
    os.mkdir('./data/'+folderToSaveIn+'/test/>1.0')



root = './data/'+folderToSaveIn+'/'

# currently alternating 50-50 between training and testing. This logic can of course
alternate = True
for folder in listOfMRIFolders:
    diagnosis = getClassification(folder)

    if 0==diagnosis:
        label = '00'
    elif diagnosis <= 1.:
        label ='0.5-1.0'
    else:
        label = '>1.0'

    for imagePath in getAllMRIFilePaths(folder):
        pair = mt_datasets.SegmentationPair2D(imagePath, None)
        slice_pair = pair.get_pair_slice(slice)
        input_slice = slice_pair["input"]
        img = input_slice

        if img.shape == (176,256):
            pngName = folder + '_diagnonis='+str(int(diagnosis))



            if alternate:
                fullTrainPath = root+'train/'+str(label)+'/'+pngName+'.png'
                matplotlib.image.imsave(fullTrainPath, img)
                convertFileToGreyscale(fullTrainPath)
                alternate = not alternate
            else:
                fullTestPath = root+'test/'+str(label)+'/'+pngName+'.png'
                matplotlib.image.imsave(fullTestPath, img)
                convertFileToGreyscale(fullTestPath)
```

```
                alternate = not alternate
```

0.1.3 Use the following command to generate PNG images at a particular slice of the brain. See the code block above to identify the directory structure in which they will be saved in. Note that we have defaulted this directory for a slice at vertical level of 105. Use the scan visualization tool more above to check which slice you want.

Note that an 'annat' will be printed out for every MRI file converted

```
In [8]: generateImagesWithSlice(105)
```

```
['anat3', 'anat4', 'anat5']
['anat3']
['anat3']
['anat4']
['anat3']
['anat3']
['anat2']
['anat2']
['anat4']
['anat3', 'anat2']
['anat2']
['anat3']
['anat2']
['anat2']
['anat3']
['anat3', 'anat2']
['anat2']
['anat2']
['anat3', 'anat2']
['anat3', 'anat4']
['anat3']
['anat2']
['anat3', 'anat2']
['anat4']
['anat3', 'anat2']
['anat2']
['anat3', 'anat2']
['anat4']
['anat2']
['anat4']
['anat6', 'anat3', 'anat4', 'anat5']
['anat4']
['anat3', 'anat2']
['anat3', 'anat2']
['anat4']
```

```
['anat4']
['anat3', 'anat4']
['anat4']
['anat2']
['anat2']
['anat3', 'anat2']
['anat3', 'anat2']
['anat3', 'anat2']
['anat2']
['anat2']
['anat3', 'anat2']
['anat3', 'anat2']
['anat2']
['anat7']
['anat4']
['anat4']
['anat1']
['anat4']
['anat3']
['anat3', 'anat2']
['anat2']
['anat3', 'anat2']
['anat3', 'anat4', 'anat5']
['anat2']
['anat2']
['anat4']
['anat7', 'anat6', 'anat3', 'anat4', 'anat5']
['anat3', 'anat2']
['anat3']
['anat3', 'anat4']
['anat2']
['anat2']
['anat3', 'anat4']
['anat3', 'anat2']
['anat3', 'anat2']
['anat3']
['anat4']
['anat3']
['anat3']
['anat4']
['anat2']
['anat4', 'anat5']
['anat2']
['anat2']
['anat2']
['anat2']
['anat2']
['anat2']
```

```
['anat2']
['anat3', 'anat2']
['anat4']
['anat3', 'anat2']
['anat3', 'anat2']
['anat2']
['anat3']
['anat3']
['anat4', 'anat5']
['anat3']
['anat3', 'anat2']
['anat3', 'anat4', 'anat5']
['anat3', 'anat2']
['anat7']
['anat2']
['anat2']
['anat2']
['anat4']
['anat3', 'anat2']
['anat3']
['anat1']
['anat3', 'anat2']
['anat4']
```

0.1.4 Sample dataloaders for PyTorch

```
In [9]: import torch
        from torchvision import datasets, transforms

        data_root = 'data/slice105/'
        train_root = data_root + 'train'
        test_root = data_root + 'test'


        train_dataset = datasets.ImageFolder(root=train_root)
        test_dataset = datasets.ImageFolder(root=test_root)

        def get_data_loaders(batch_size):
            train_loader = torch.utils.data.DataLoader(
                    train_dataset, batch_size=batch_size, shuffle=True, num_workers=4)
            test_loader = torch.utils.data.DataLoader(
                    test_dataset, batch_size=batch_size, shuffle=False, num_workers=4)
            return (train_loader, test_loader)
```

0.1.5 Test if image is RGB or gresycale. Usefule if working with 1 channel or 3 channel CNNs

```
In [10]: from scipy.misc import imread, imsave, imresize
         image = imread('data/slice105/test/00/OAS30004_MR_d2229_diagnonis=0.png')
         if(len(image.shape)<3):
               print( 'gray')
         elif len(image.shape)==3:
               print( 'Color(RGB)')
         else:
               print( 'others')
```

```
gray


/usr/local/Cellar/ipython/6.5.0/libexec/vendor/lib/python3.7/site-packages/ipykernel_launcher.py
`imread` is deprecated in SciPy 1.0.0, and will be removed in 1.2.0.
Use ``imageio.imread`` instead.
```