Facial Recognition

Author

Name: Aqsa mahmood Email: aqsa.mahmood@city.ac.uk

Hardware Specification

This project was run on the following hardware specification: | Hardware | Description | | ------ | ------ | | System Model: | MS-7B93 | | System Type | x64-based PC | | Processor(s): | AMD64 Family 23 Model 113 Stepping 0 AuthenticAMD ~3800 Mhz | | Graphics | NVIDIA GeForce RTX 2070 SUPER | | BIOS Version: | American Megatrends Inc. 1.00, 14/06/2019 | |Total Physical Memory: | 16,332 MB|

Code Structure

```
GANKneeMRI2
|   README.md
|   environment.yml.yml
|   setup.bat
|   setup.sh
└───src
|   └───models
|       |   mrnet.py
|       |   resnet_G_and_C.py
|   └───utils
|       |   data_load.py
|       |   fid.py
|       |   knee_mri_viewer.ipynb
|       |   logger.py
|       |   utilities.py
|   └───training
|       |   baseline_model_training.py
|       |   wasserstein_GP_256.py
|   └───traininer
|       |   basenet_model_task.py
|       |   wgan_task.py
|   └───bash_scripts
|       |   base-mode-train-cloud.bat
|       |   base-mode-train-cloud.sh
|       |   wgan-model-train.bat
|       |   wgan-model-train.sh
|       |   mrnet-model-train.bat
|       |   mrnet-model-train.sh
|   └───model_weights
|       |   ....
|       |   ....
|   |   Dockerfile
|   |   Evaluating_classificationPerf .py
|   |   mrNet_main.py
|   |   setup.py
|   |   testing_GAN_model.py
|   |   wgan_main.py
```

Python File/Folder Description

- testing_GAN_model.- this file essentially allows test mri scans to be genrated from the trained WGAN models
- mrNet_main.py - this file essentially allows the training of the baseline model
- wgan_main.py - this file essentially allows the training of WGAN
- untilities.py - this file contains general utility functions used throught the files
- dataset_loader.py - contains functions to allow loading the MRI Dataset
- mrnet.py - contains the baseline model
- resnet_G_and_C.py - contains WGAN Models Discriminator and Generator
- setup.py - this file is for Google Cloud purposes and is not yet functinal. The contents of this can be ignored.
- Dockerfile - this file is for Google Cloud purposes and is not yet functinal
- /trainer - this directory is for Google Cloud purposes and is not yet functinal

  #### Model Downloads

  The 'model_weights' directory located in src folder contains weights of the trained WGAN models

Installation

First, install Python 3.7 and Anconda Then we need to setup the anaconda enviroment. On a linux environment, navigate to the FacialRecognition project, run the follwing on command line as super user:

```
$ ./setup.sh
```

This script shouuld create the conda enviroment, with all devlopment dependencies, and activate the environment 'project2020' | Depenedency | Version | | ------ | ------ | | python | 3.7.6 | | pytorch | 1.2.0 | | pillow | 6.1| | torchvision | 0.4.0 | | fastai | 1.0.57 | | ipykernel: | 4.6.1 | | pytest | 3.6.4 | | bqplot | | scikit-learn | 0.19.1 | | pip | 19.0.3 | | cython | 0.29.1 v | papermill | 1.2.0 | | black | 18.6b4 v | ipywebrtc | | lxml | 4.3.2 | | pre-commit | 1.14.4 | | pyyaml | 5.1.2 | | requests | 2.22.0 v | einops | 0.1.0 | | cytoolz | | seaborn | | mtcnn | | scikit-image | | decord | 0.3.5 | | nvidia-ml-py3 | | nteract-scrapbook | | azureml-sdk[notebooks,contrib] | 1.0.30 | | facenet-pytorch | | opencv-python | 3.4.2.16 | | opencv-contrib-python | 3.4.2.16 | | jupyter | 1.0.0 | | spyder | 4.1.3 |

On a windows environment, can run the following script via Anaconda Prompt :

```
$ setup.bat
```

Note that setuping up the environment does take some time. Once, the the script has finished, the environment 'project2020'. On the same Anaconda Prompt, spyder can be launched via the follwing command:

```
$ spyder
```

Running Application

The python files of interest is **wgan_main.py** and **mrNet_main.py**. These python files can be invoked via scripts located in the 'bash_scripts' directory. The wgan-model-train.bat will run the wgan_main.py file, and will start WGAN training. The batch file can be invoked via Anaconda Prompt with the follwing example command:

```
$ wgan-model-train.bat --base-results-dir="D:/results" --data-
dir="C:/Users/adds0/OneDrive - University of
Glasgow/Documnets/courses/Msc/Project/MRI/MRNet/resources" --plane=axial --
diagnoses=ACL --dataset-type=train
```

The above argument call would train a WGAN model for the dataset that is of axial plane and which have a diagnosis of ACL. Note that the first argument '--base-results-dir' is the resutls directory path, and the second argument, '--data-dir', is the location of the original dataset.

The wgan-model-train.sh can be run from Anaconda Prompt by invoking the follwing command:

```
$ wgan-model-train.sh --base-results-dir="D:/results" --data-
dir="C:/Users/adds0/OneDrive - University of
Glasgow/Documnets/courses/Msc/Project/MRI/MRNet/resources" --plane='axial' --
diagnoses='ACL' --dataset-type='train'
```

***Note before invoking the wgan-model-train.sh, it needs to be made executable with the follwing command**:

```
$ chmod u+x wgan-model-train.sh
```

If one wants to train a WGAN model for a differet plane, then one can specify 'sagittal' or 'coronal'. Alternatively, if WGAN need to be trained for a diferent diagnoses, the follwing options are avilable:

- 'Abnormal' - only selects scans that are only abnormal
- 'Meniscus'- only selects scans that are only Meniscus
- 'All' - selects scans that are all ACL,Meniscus and Abnormal
- 'None' - selects scans that are normal

The mrnet-model-train.bat will run the mrNet_main.py file, and will start base model training with or without the augmented dataset . The batch file can be invoked via Anaconda Prompt with the follwing example command:

```
$ mrnet-model-train.bat --base-results-dir="D:/results" --data-
dir="C:/Users/adds0/OneDrive - University of
Glasgow/Documnets/courses/Msc/Project/MRI/MRNet/resources" --plane=axial --dataset-
type=train --epochs=100
```

The above argument call would train a Base MRNet model for the dataset that is of axial plane without a augmented dataset. Note that the first argument '--base-results-dir' is the resutls directory path, and the second argument, '--data-dir', is the location of the original dataset. To include training with an augmented dataset,specify '--augmented-data-dir' with the directory which contains the augemnted dataset. Ensure the directory structure is similar to the contents within '--data-dir'. Again,If one wants to train the baseline model for a differet plane, then one can specify 'sagittal' or 'coronal'.