MRI Data Classification using CNN

Submitted By:

Divya Saxena (1001773376)

Table of Contents

1. Introduction

Neuro-imaging is a branch of medical imaging that focuses on the brain. In addition to diagnosing disease and assessing brain health, neuro-imaging also studies:

- How the brain works
- How various activities impact the brain

In deep learning a **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep neural networks most commonly applied to analyzing visual imagery

The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers.

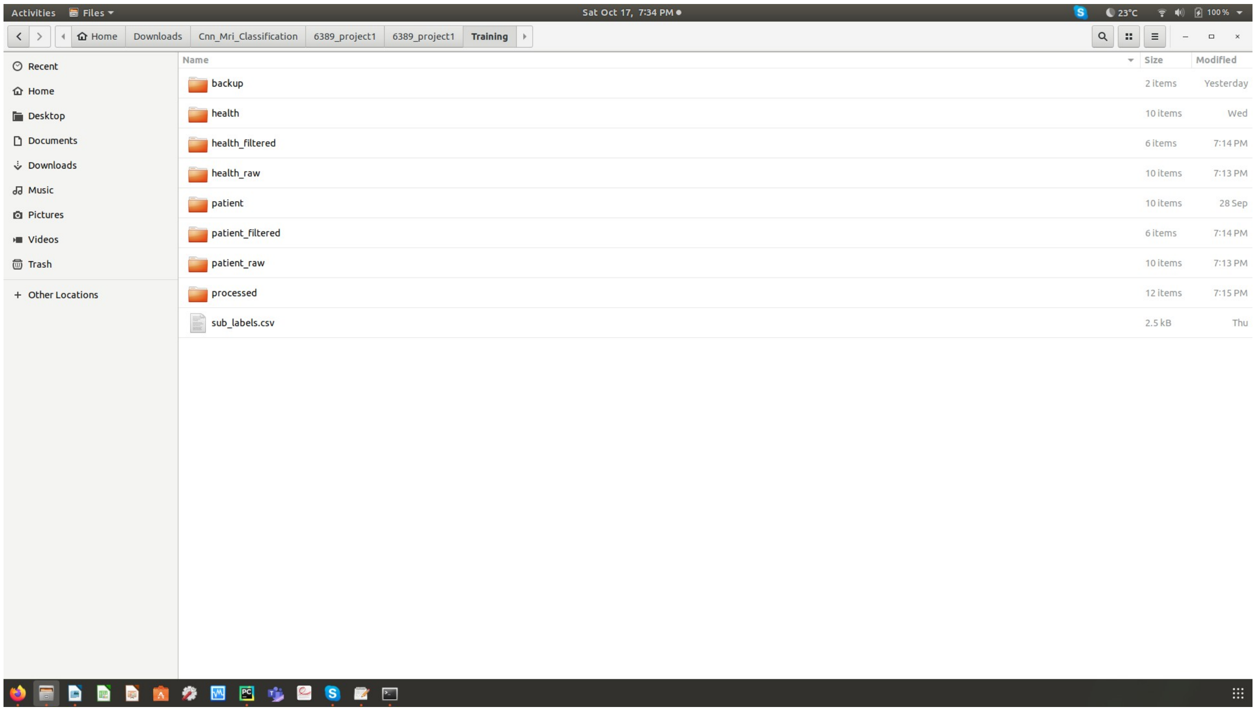
MRI Images have been given as Input to CNN for binary classification of images as Healthy and Patient.

Independent test dataset has been used for evaluating the model performance.

2. Code Description

Python Files
/Cnn_Mri_Classification/Data_preprocessing/preprocess_data.py
/Cnn_Mri_Classification/Data_preprocessing/preprocessing_testing.py
/Cnn_Mri_Classification/Data_preprocessing/show_image.py
/Cnn_Mri_Classification/Model_training_testing/cnn_model.py
/Cnn_Mri_Classification/Model_training_testing/training.py

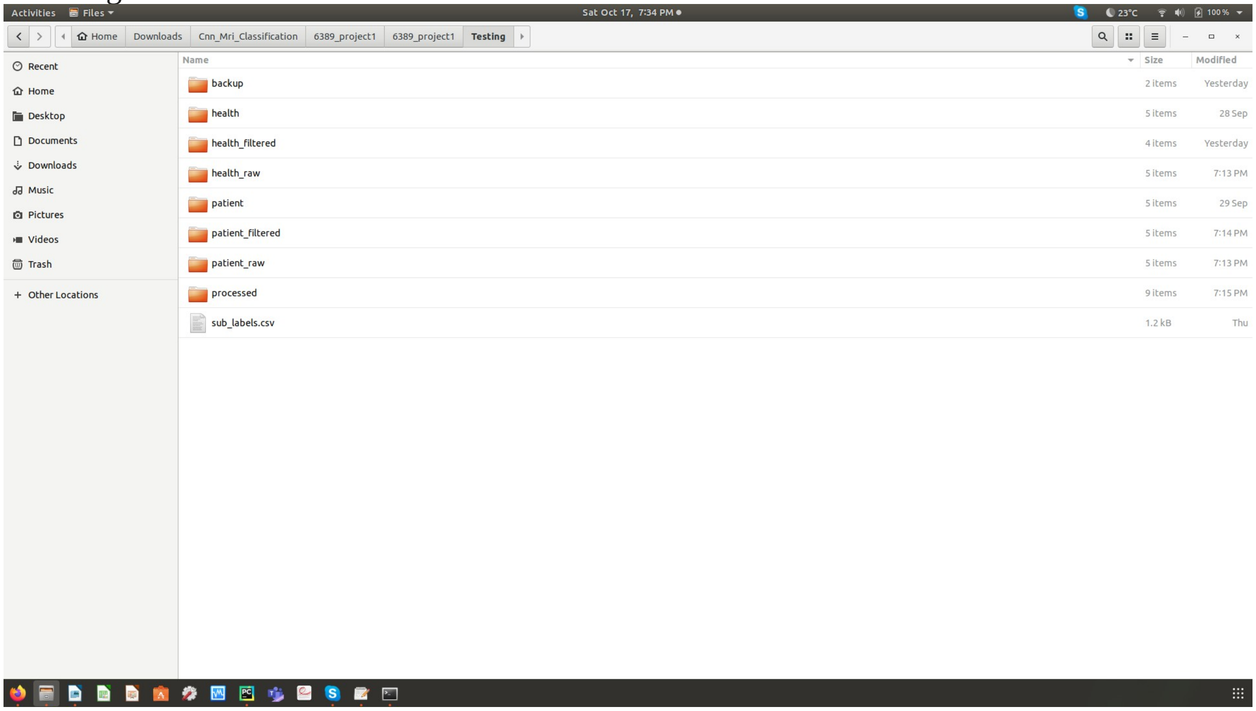
Folder Structure for dataset:
/Cnn_Mri_Classification/6389_project1/6389_project1/Training
We have created some folders after each stage during pre-processing as attached below, images during each stage will get saved in new folder, so that we can easily identify the changes during each stage.



Name	Size	Modified
backup	2 items	Yesterday
health	10 items	Wed
health_filtered	6 items	7:14 PM
health_raw	10 items	7:13 PM
patient	10 items	28 Sep
patient_filtered	6 items	7:14 PM
patient_raw	10 items	7:13 PM
processed	12 items	7:15 PM
sub_labels.csv	2.5 kB	Thu

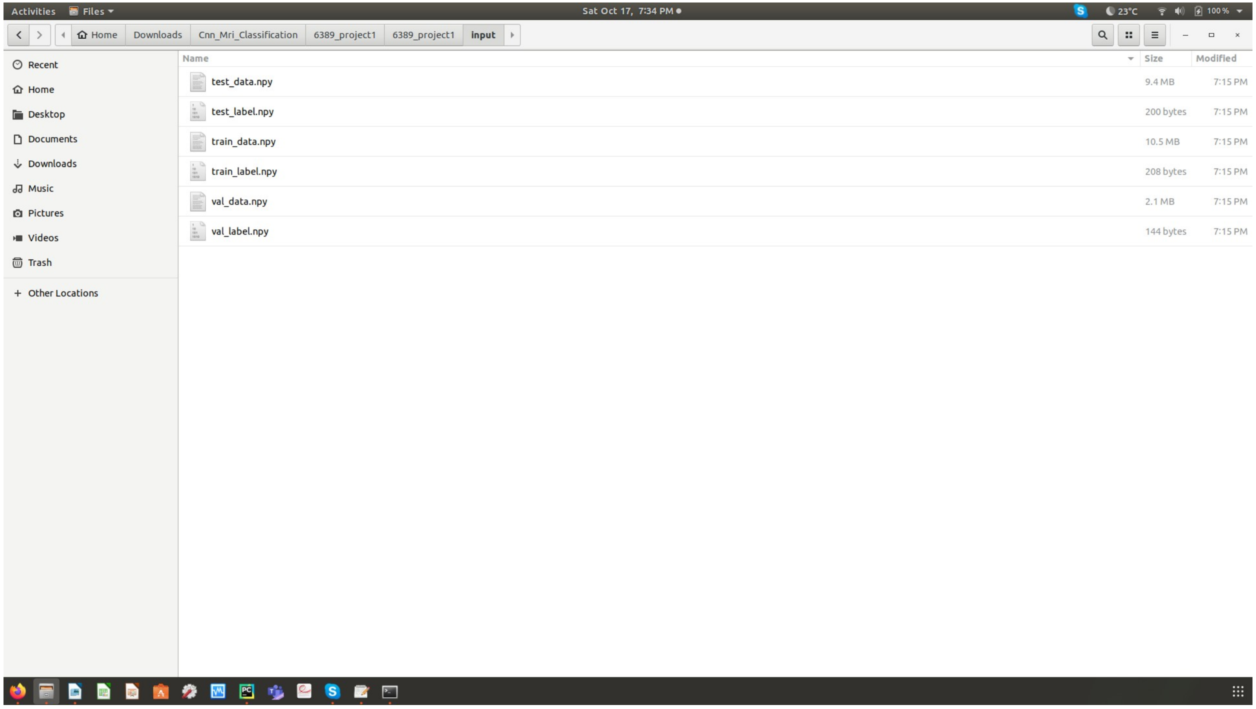
/Cnn_Mri_Classification/6389_project1/6389_project1/Testing

We have created some folders after each stage during pre-processing as attached below, images during each stage will get saved in new folder, so that we can easily identify the changes during each stage.



Name	Size	Modified
backup	2 items	Yesterday
health	5 items	28 Sep
health_filtered	4 items	Yesterday
health_raw	5 items	7:13 PM
patient	5 items	29 Sep
patient_filtered	5 items	7:14 PM
patient_raw	5 items	7:13 PM
processed	9 items	7:15 PM
sub_labels.csv	1.2 kB	Thu

After pre-processing, some files have been generated for the CNN model as attached below. These files will be used by CNN model for traing/testing/validation /Cnn_Mri_Classification/6389_project1/6389_project1/input



Name	Size	Modified
test_data.npy	9.4 MB	7:15 PM
test_label.npy	200 bytes	7:15 PM
train_data.npy	10.5 MB	7:15 PM
train_label.npy	208 bytes	7:15 PM
val_data.npy	2.1 MB	7:15 PM
val_label.npy	144 bytes	7:15 PM

3. How to run code

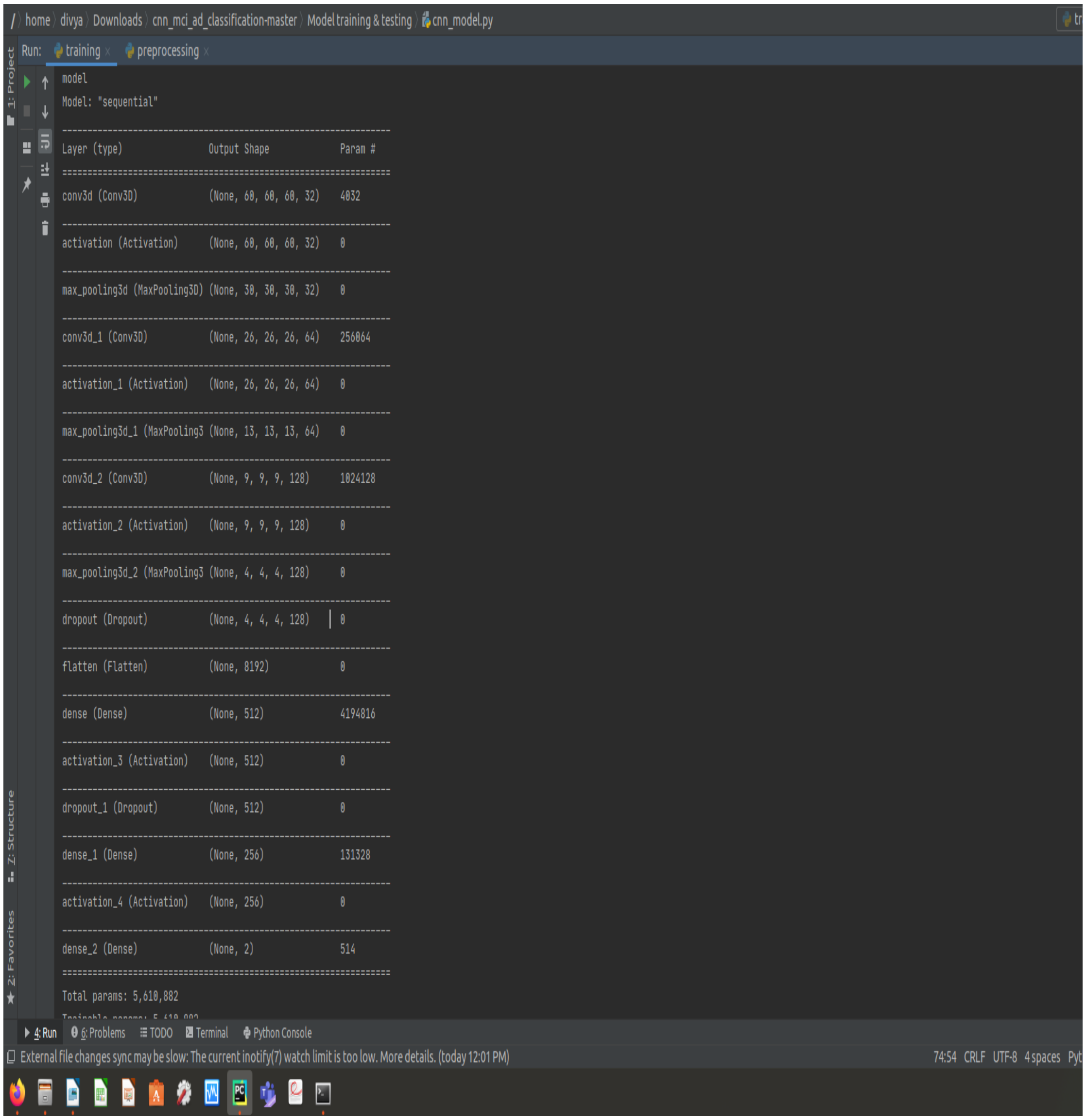
To Show MRI Image
python /Cnn_Mri_Classification/Data_preprocessing/show_image.py

For Pre-processing Data
python /Cnn_Mri_Classification/Data_preprocessing/preprocessing_testing.py

To Train Model
python /Cnn_Mri_Classification/Model_training_testing/training.py

4. Convolutional Neural Network Model Description

”sequential” model has been used to create CNN. 17 layers have used to create the model. Attaching screenshot for the same. MRI Images given are very big in size approximately 128*128*128 as we can see here .MRI Image has been resized to 64*64*64 due to limited GPU and CPU.
Adam optimizer has been used as it is an adaptive learning rate optimization algorithm that’s been designed specifically for training deep neural networks.


/ home divya Downloads cnn_mci_ad_classification-master Model training & testing cnn_model.py
Run: training preprocessing
1: Project
Z: Structure
2: Favorites
4: Run
6: Problems
TODO
Terminal
Python Console
74:54 CRLF UTF-8 4 spaces

```
model
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv3d (Conv3D)              (None, 60, 60, 60, 32)    4032
_________________________________________________________________
activation (Activation)      (None, 60, 60, 60, 32)    0
_________________________________________________________________
max_pooling3d (MaxPooling3D) (None, 30, 30, 30, 32)    0
_________________________________________________________________
conv3d_1 (Conv3D)            (None, 26, 26, 26, 64)    256064
_________________________________________________________________
activation_1 (Activation)    (None, 26, 26, 26, 64)    0
_________________________________________________________________
max_pooling3d_1 (MaxPooling3 (None, 13, 13, 13, 64)    0
_________________________________________________________________
conv3d_2 (Conv3D)            (None, 9, 9, 9, 128)      1024128
_________________________________________________________________
activation_2 (Activation)    (None, 9, 9, 9, 128)      0
_________________________________________________________________
max_pooling3d_2 (MaxPooling3 (None, 4, 4, 4, 128)      0
_________________________________________________________________
dropout (Dropout)            (None, 4, 4, 4, 128)      0
_________________________________________________________________
flatten (Flatten)            (None, 8192)              0
_________________________________________________________________
dense (Dense)                (None, 512)               4194816
_________________________________________________________________
activation_3 (Activation)    (None, 512)               0
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
activation_4 (Activation)    (None, 256)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 514
=================================================================
Total params: 5,610,882
Trainable params: 5,610,882
```

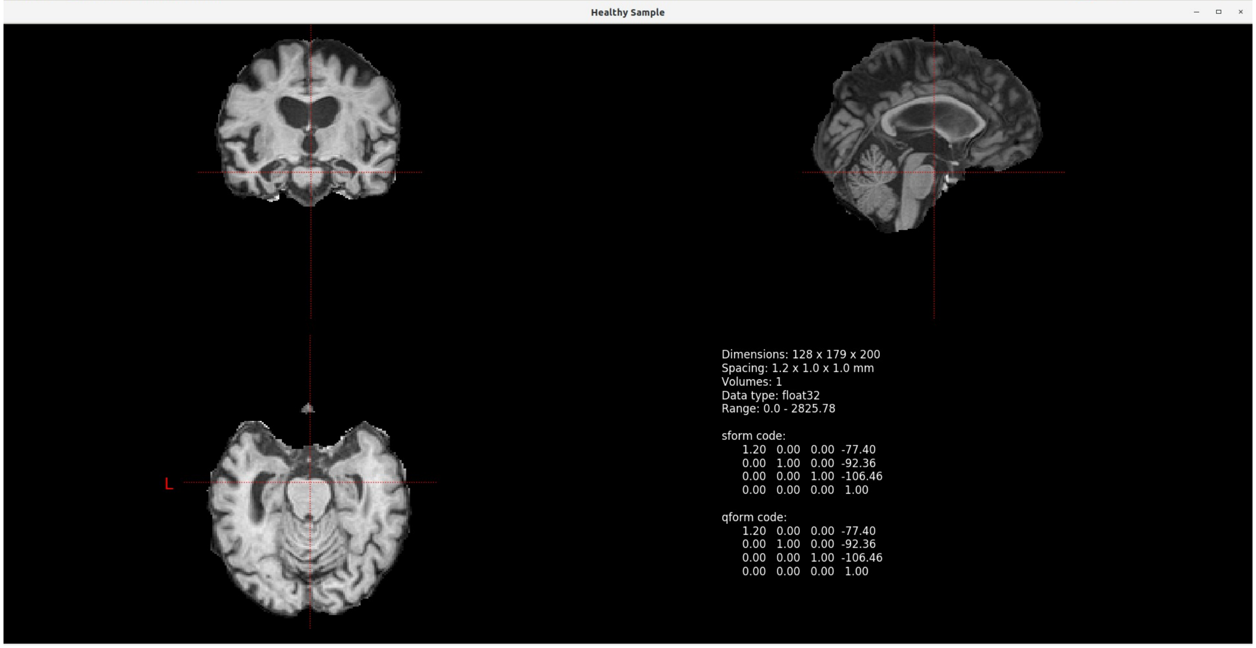
External file changes sync may be slow: The current inotify(7) watch limit is too low. More details. (today 12:01 PM)

5. MRI Image Visualization Healthy vs Patient

/Cnn_Mri_Classification/Data_preprocessing/show_image.py
In this python file, we have tried to show the 3D-image of both Healthy and Patient Dataset. We have randomly selected one image of both dataset given.

We also calculated some details of the Images as shown in screen shots like Dimensions, Spacing, Volume, Datatype, Range etc.

1. Health

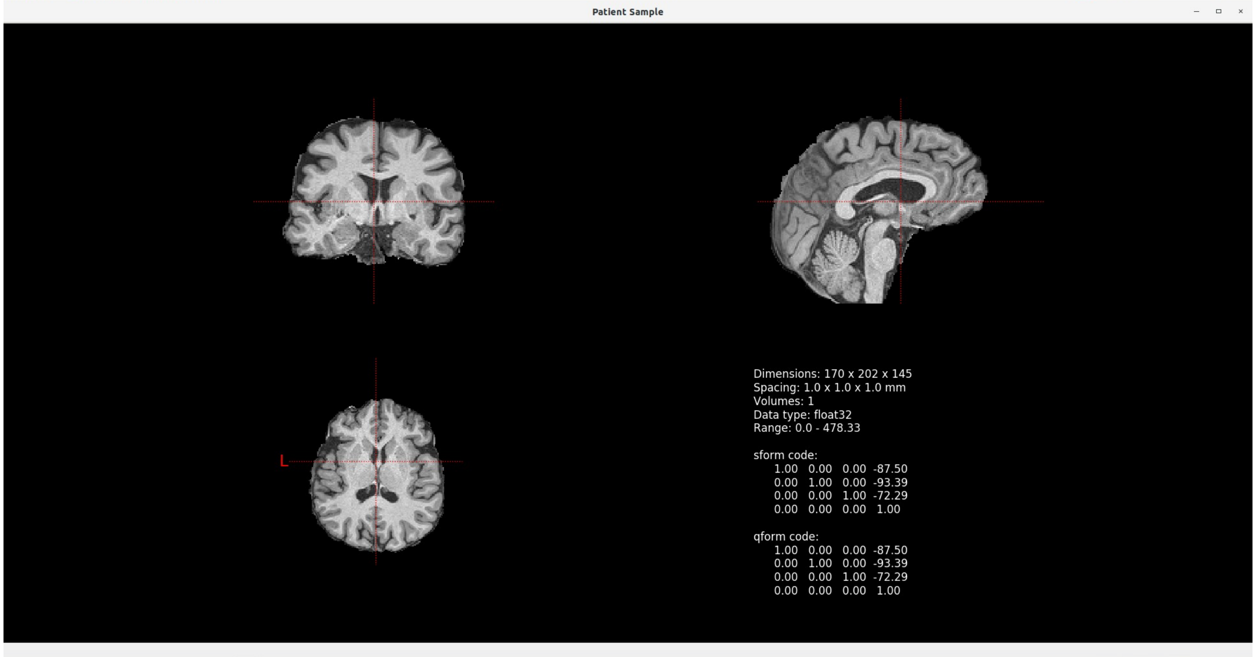


```
Dimensions: 128 x 179 x 200
Spacing: 1.2 x 1.0 x 1.0 mm
Volumes: 1
Data type: float32
Range: 0.0 - 2825.78

sform code:
     1.20   0.00   0.00  -77.40
     0.00   1.00   0.00  -92.36
     0.00   0.00   1.00  -106.46
     0.00   0.00   0.00   1.00

qform code:
     1.20   0.00   0.00  -77.40
     0.00   1.00   0.00  -92.36
     0.00   0.00   1.00  -106.46
     0.00   0.00   0.00   1.00
```

2. Patient



```
Dimensions: 170 x 202 x 145
Spacing: 1.0 x 1.0 x 1.0 mm
Volumes: 1
Data type: float32
Range: 0.0 - 478.33

sform code:
     1.00   0.00   0.00  -87.50
     0.00   1.00   0.00  -93.39
     0.00   0.00   1.00  -72.29
     0.00   0.00   0.00   1.00

qform code:
     1.00   0.00   0.00  -87.50
     0.00   1.00   0.00  -93.39
     0.00   0.00   1.00  -72.29
     0.00   0.00   0.00   1.00
```

We can see some differences in both Health and patient images.

6. Data Preprocessing

/Cnn_Mri_Classification/Data_preprocessing/preprocess_data.py
/Cnn_Mri_Classification/Data_preprocessing/preprocessing_testing.py

Below steps have been followed for both Training and Testing dataset but testing data kept separately to avoid over-fitting of the model.

Code has been implemented as below:
1.Unzip .nii in patient_raw, health_raw
nii raw MRI images will be saved in folder separately for Training and testing i.e.
(/Cnn_Mri_Classification/6389_project1/6389_project1/Training/health_raw,/
Cnn_Mri_Classification/6389_project1/6389_project1/Training/patient_raw)
and
(/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/health_raw,/
Cnn_Mri_Classification/6389_project1/6389_project1/Testing/patient_raw)

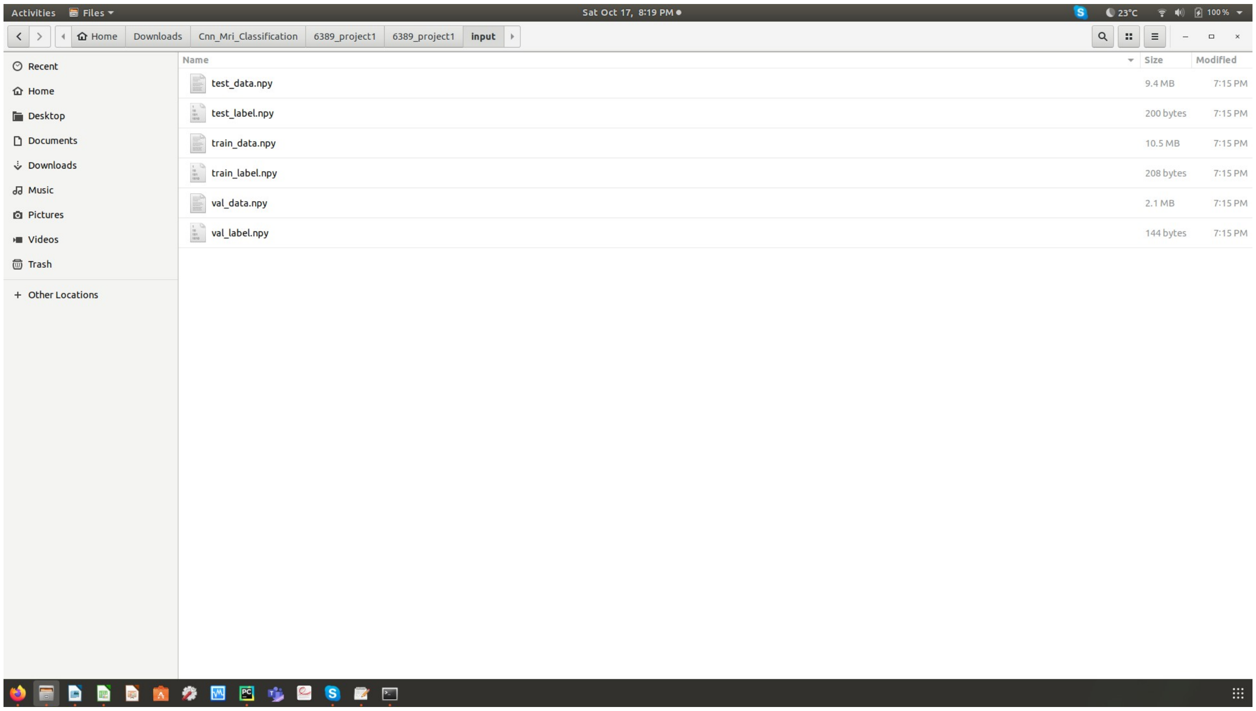
2.Filtered .nii to .npy in folder(patient_filtered, health_ filtered)
Images pre-processed differently also have a different head orientation, so they were removed. This is done by filtering the dimensions. By checking, it is found that if the last dimension s[2] is larger than the first dimension s[0], then the head orientation in this image is different to the majority, also the image was pre-processed differently can be seen from the name of the file.
(/Cnn_Mri_Classification/6389_project1/6389_project1/Training/health_filtered,/
Cnn_Mri_Classification/6389_project1/6389_project1/Training/patient_filtered)
and
(/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/health_filtered,/
Cnn_Mri_Classification/6389_project1/6389_project1/Testing/patient_filtered)

3.Processed all healthy/patient to processed folder
Pre-processing of original images. Pre-processing include: skull-stripping, brain cropping, resizing and intensity-normalization.
MRI Image has been resized to 64*64*64 due to limited GPU and CPU.
(/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed)
and
(/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed)

4.Final step is after pre-processing, data has been split into 2 parts :
Training and Validation. All the data has been converted into single files .npy. And Testing dataset as well. All files have been created in Input folder as data and their corresponding labels

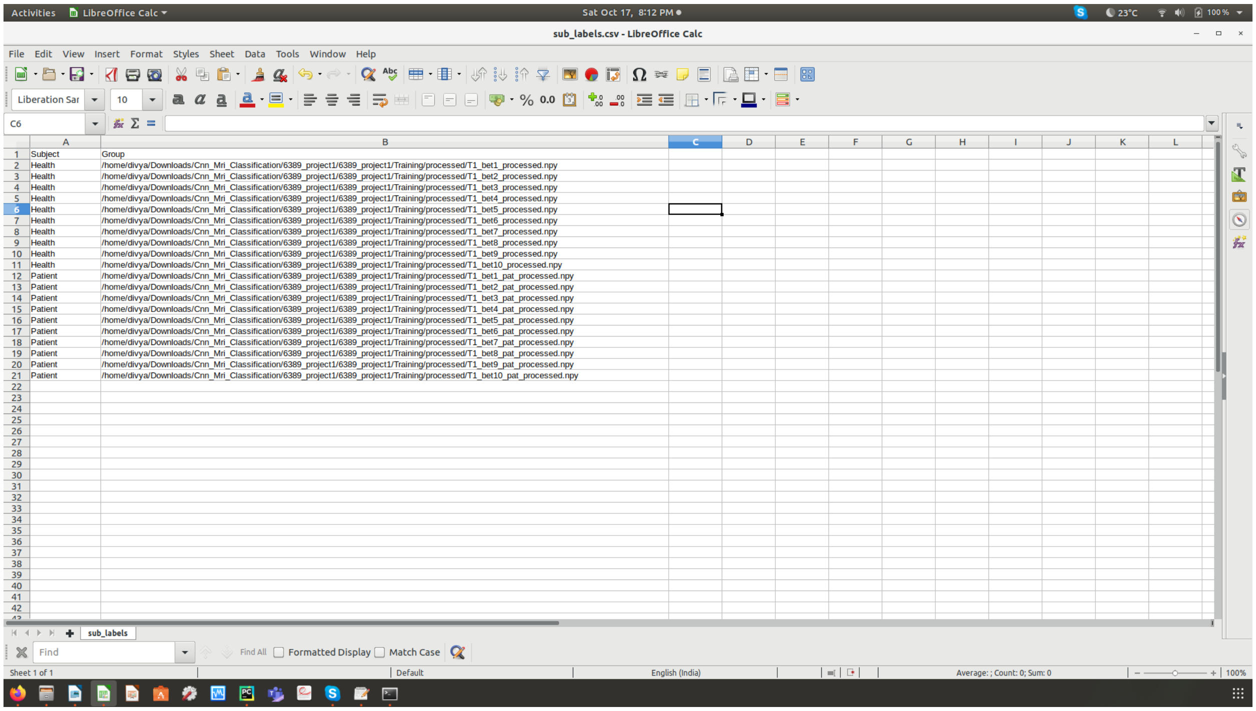
1) test_data.npy - Array of Input images for Testing dataset
2) test_label.npy - Array of Labels of Input images for Testing dataset
3) train_data.npy - Array of Input images for Training dataset
4) train_label.npy - Array of Labels of Input images for Training dataset
5) val_data.npy - Array of Input images for Vaidation dataset

6) val_label.npy - Array of Labels of Input images for Validation dataset



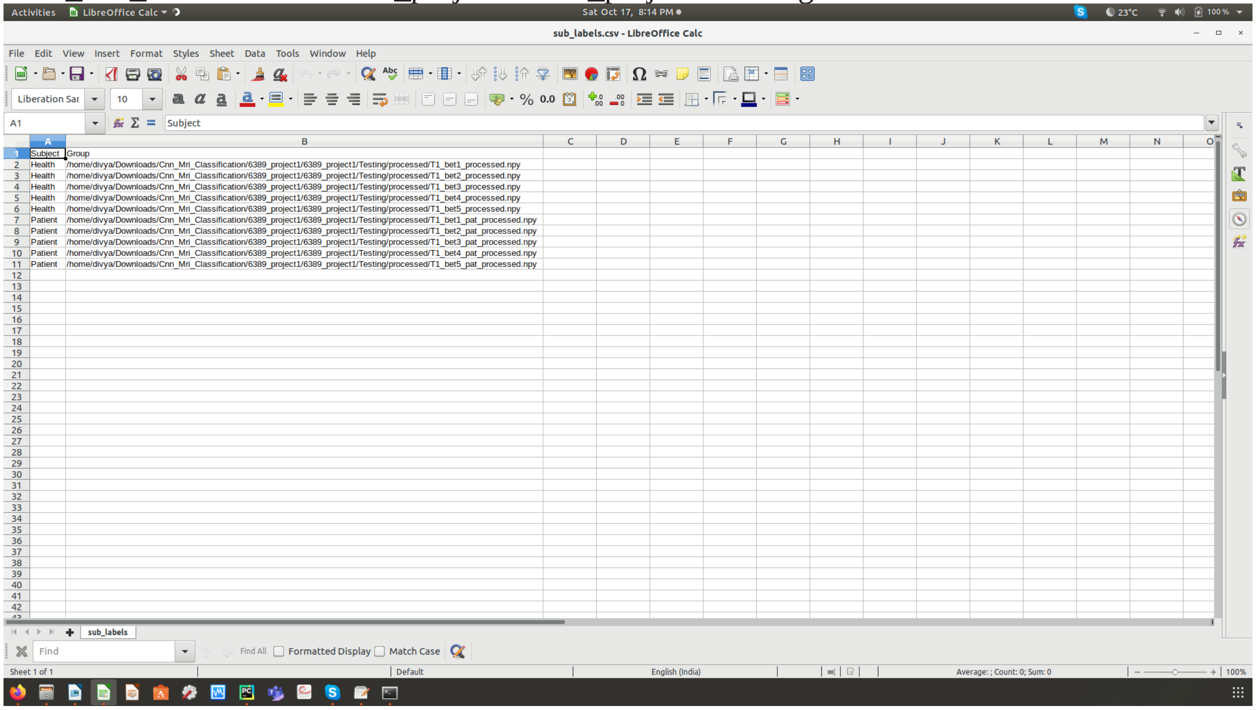
Name	Size	Modified
test_data.npy	9.4 MB	7:15 PM
test_label.npy	200 bytes	7:15 PM
train_data.npy	10.5 MB	7:15 PM
train_label.npy	208 bytes	7:15 PM
val_data.npy	2.1 MB	7:15 PM
val_label.npy	144 bytes	7:15 PM

5.Some manual steps required for Training the model
Created labels for Training/Validation in sub_label.csv as Subject and Group as the heading at location /Cnn_Mri_Classification/6389_project1/6389_project1/Training


sub_labels.csv - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
C6
sub_labels
Sheet 1 of 1
Default
English (India)

	A	B
1	Subject	Group
2	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet1_processed.npy
3	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet2_processed.npy
4	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet3_processed.npy
5	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet4_processed.npy
6	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet5_processed.npy
7	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet6_processed.npy
8	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet7_processed.npy
9	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet8_processed.npy
10	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet9_processed.npy
11	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet10_processed.npy
12	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet1_pat_processed.npy
13	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet2_pat_processed.npy
14	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet3_pat_processed.npy
15	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet4_pat_processed.npy
16	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet5_pat_processed.npy
17	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet6_pat_processed.npy
18	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet7_pat_processed.npy
19	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet8_pat_processed.npy
20	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet9_pat_processed.npy
21	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed/T1_bet10_pat_processed.npy

Created labels for Testing in sub_label.csv as Subject and Group as the heading at location /Cnn_Mri_Classification/6389_project1/6389_project1/Testing


sub_labels.csv - LibreOffice Calc
File Edit View Insert Format Styles Sheet Data Tools Window Help
A1
Subject
sub_labels
Sheet 1 of 1
Default
English (India)

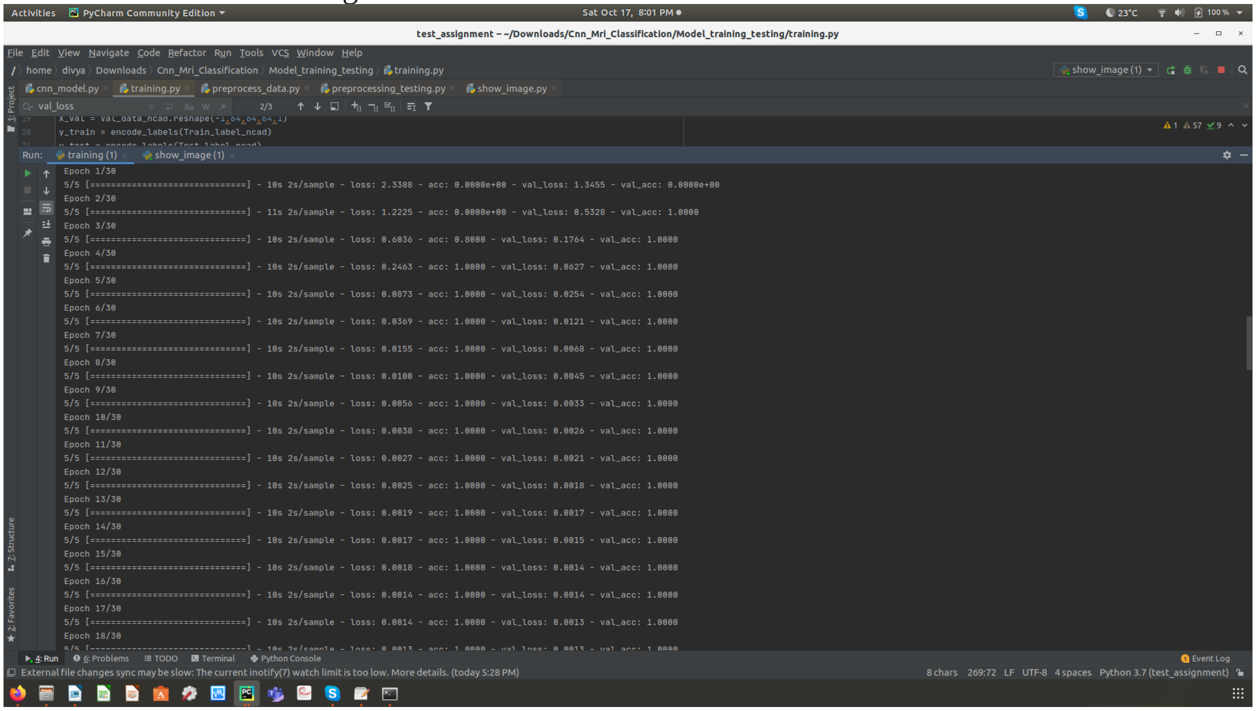
	A	B
1	Subject	Group
2	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet1_processed.npy
3	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet2_processed.npy
4	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet3_processed.npy
5	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet4_processed.npy
6	Health	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet5_processed.npy
7	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet1_pat_processed.npy
8	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet2_pat_processed.npy
9	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet3_pat_processed.npy
10	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet4_pat_processed.npy
11	Patient	/home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/processed/T1_bet5_pat_processed.npy

7. Training CNN Model

/Cnn_Mri_Classification/Model_training_testing/cnn_model.py

/Cnn_Mri_Classification/Model_training_testing/training.py

Model has been trained for 30 number of epochs as attached below using Training and validation dataset. At Epoch 00024: ReduceLROnPlateau reduced learning rate to 4.999999873689376e-06. Testing dataset has been kept aside for evaluating the model. It has been used as Independent dataset to avoid over-fitting.

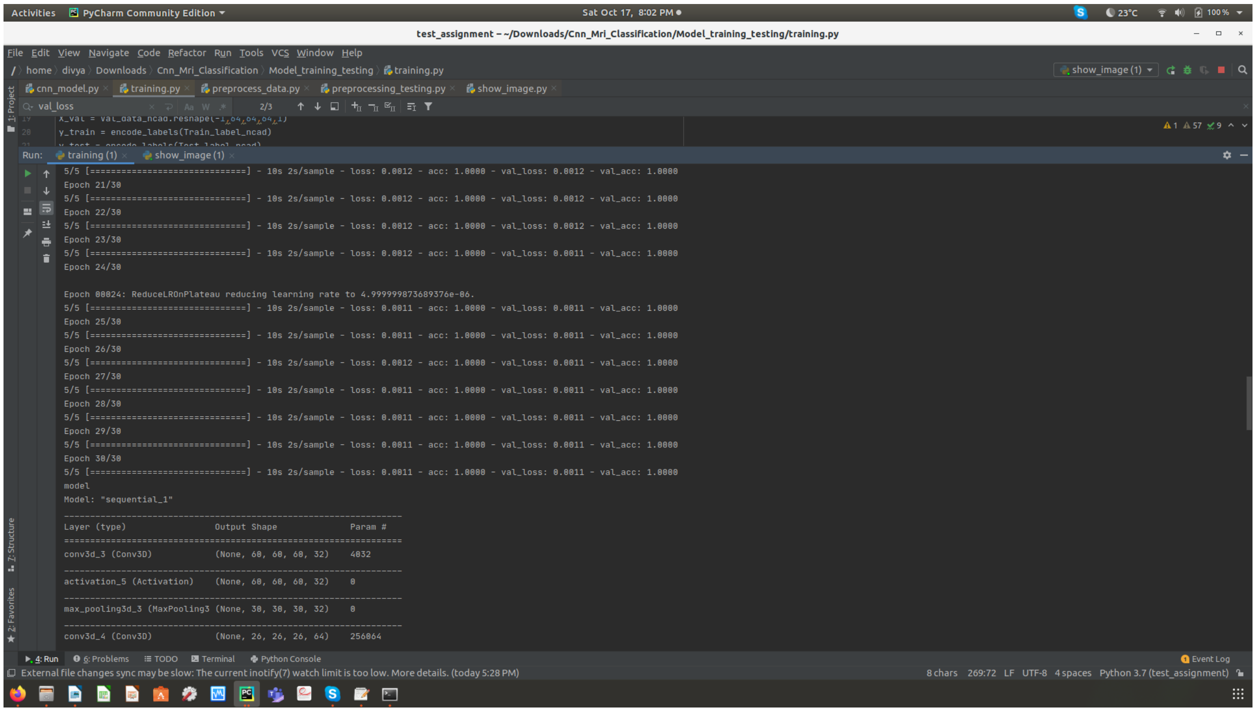

Activities
PyCharm Community Edition
Sat Oct 17, 8:01 PM
23°C
100 %
test_assignment – ~/Downloads/Cnn_Mri_Classification/Model_training_testing/training.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
/ home divya Downloads Cnn_Mri_Classification Model_training_testing training.py
show_image (1)
cnn_model.py training.py preprocess_data.py preprocessing_testing.py show_image.py
1: Project
val_loss
2/3
Run: training (1) show_image (1)
7: Structure
2: Favorites
4: Run 6: Problems TODO Terminal Python Console
Event Log
8 chars 269:72 LF UTF-8 4 spaces Python 3.7 (test_assignment)

```
x_val = val_data_ncad.reshape(-1,64,64,64,1)
y_train = encode_labels(Train_label_ncad)
y_test = encode_labels(Test_label_ncad)
```

```
Epoch 1/30
5/5 [==============================] - 10s 2s/sample - loss: 2.3388 - acc: 0.0000e+00 - val_loss: 1.3455 - val_acc: 0.0000e+00
Epoch 2/30
5/5 [==============================] - 11s 2s/sample - loss: 1.2225 - acc: 0.0000e+00 - val_loss: 0.5328 - val_acc: 1.0000
Epoch 3/30
5/5 [==============================] - 10s 2s/sample - loss: 0.6036 - acc: 0.8000 - val_loss: 0.1764 - val_acc: 1.0000
Epoch 4/30
5/5 [==============================] - 10s 2s/sample - loss: 0.2463 - acc: 1.0000 - val_loss: 0.0627 - val_acc: 1.0000
Epoch 5/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0873 - acc: 1.0000 - val_loss: 0.0254 - val_acc: 1.0000
Epoch 6/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0369 - acc: 1.0000 - val_loss: 0.0121 - val_acc: 1.0000
Epoch 7/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0155 - acc: 1.0000 - val_loss: 0.0068 - val_acc: 1.0000
Epoch 8/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0100 - acc: 1.0000 - val_loss: 0.0045 - val_acc: 1.0000
Epoch 9/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0056 - acc: 1.0000 - val_loss: 0.0033 - val_acc: 1.0000
Epoch 10/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0038 - acc: 1.0000 - val_loss: 0.0026 - val_acc: 1.0000
Epoch 11/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0027 - acc: 1.0000 - val_loss: 0.0021 - val_acc: 1.0000
Epoch 12/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0025 - acc: 1.0000 - val_loss: 0.0018 - val_acc: 1.0000
Epoch 13/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0019 - acc: 1.0000 - val_loss: 0.0017 - val_acc: 1.0000
Epoch 14/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0017 - acc: 1.0000 - val_loss: 0.0015 - val_acc: 1.0000
Epoch 15/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0018 - acc: 1.0000 - val_loss: 0.0014 - val_acc: 1.0000
Epoch 16/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0014 - acc: 1.0000 - val_loss: 0.0014 - val_acc: 1.0000
Epoch 17/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0014 - acc: 1.0000 - val_loss: 0.0013 - val_acc: 1.0000
Epoch 18/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0013 - acc: 1.0000 - val_loss: 0.0013 - val_acc: 1.0000
```

External file changes sync may be slow: The current inotify(7) watch limit is too low. More details. (today 5:28 PM)

As we can see, model has been over fitted so ReduceLROnPlateau class has been tried. It can be done by reduce learning rate when a metric has stopped improving. Models often benefit from reducing the learning rate by a factor of 2-10 once learning stagnates. This callback monitors a quantity and if no improvement is seen for a 'patience' number of epochs, the learning rate is reduced. Please refer below screenshot as below:


Activities
PyCharm Community Edition
Sat Oct 17, 8:02 PM
test_assignment – ~/Downloads/Cnn_Mri_Classification/Model_training_testing/training.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
/ home divya Downloads Cnn_Mri_Classification Model_training_testing training.py
cnn_model.py training.py preprocess_data.py preprocessing_testing.py show_image.py
val_loss
2/3
Run: training (1) show_image (1)
4: Run 6: Problems TODO Terminal Python Console
8 chars 269:72 LF UTF-8 4 spaces Python 3.7 (test_assignment)
Event Log

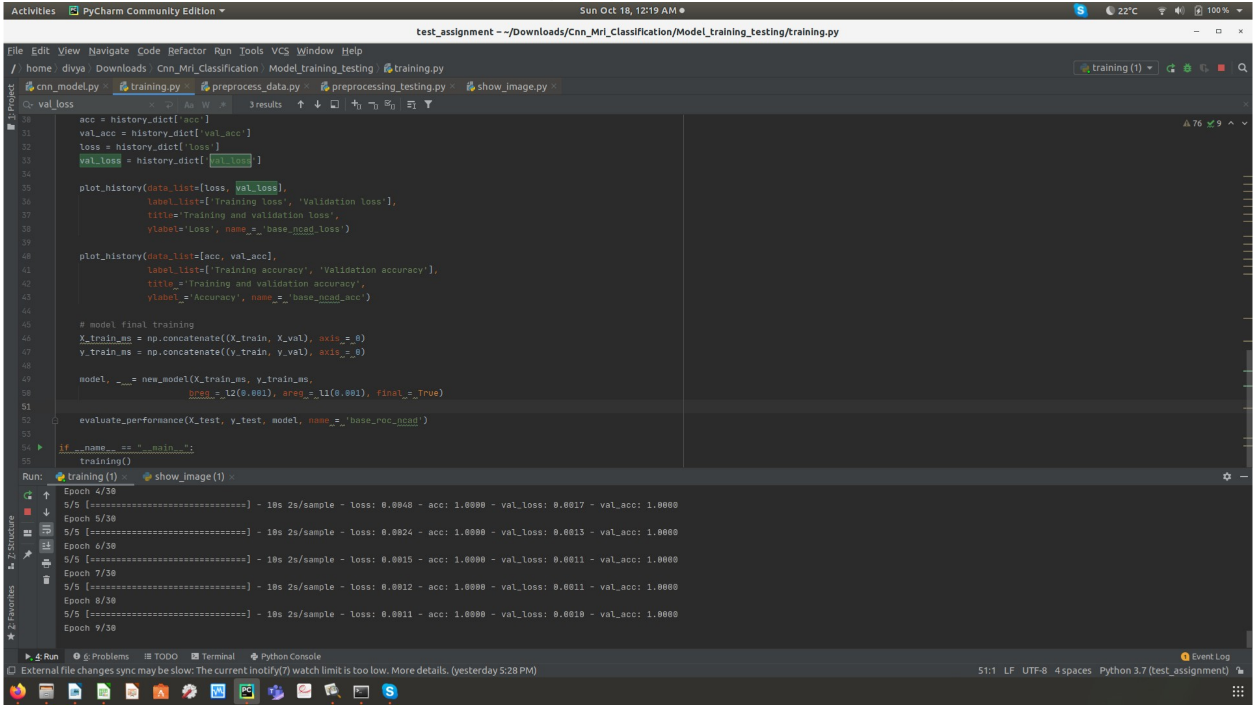
```python
X_val = val_data_ncad.reshape(-1,64,64,64,1)
y_train = encode_labels(Train_label_ncad)
y_test = encode_labels(Test_label_ncad)
```

```
5/5 [==============================] - 10s 2s/sample - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0012 - val_acc: 1.0000
Epoch 21/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0012 - val_acc: 1.0000
Epoch 22/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0012 - val_acc: 1.0000
Epoch 23/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 24/30

Epoch 00024: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-06.
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 25/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 26/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 27/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 28/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 29/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 30/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
model
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv3d_3 (Conv3D)            (None, 60, 60, 60, 32)    4032
_________________________________________________________________
activation_5 (Activation)    (None, 60, 60, 60, 32)    0
_________________________________________________________________
max_pooling3d_3 (MaxPooling3 (None, 30, 30, 30, 32)    0
_________________________________________________________________
conv3d_4 (Conv3D)            (None, 26, 26, 26, 64)    256064
```

External file changes sync may be slow: The current inotify(7) watch limit is too low. More details. (today 5:28 PM)

Testing dataset has been used independently for evaluating performance


Activities
PyCharm Community Edition
Sun Oct 18, 12:19 AM
test_assignment – ~/Downloads/Cnn_Mri_Classification/Model_training_testing/training.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help
/ home divya Downloads Cnn_Mri_Classification Model_training_testing training.py
cnn_model.py training.py preprocess_data.py preprocessing_testing.py show_image.py
val_loss
3 results
Run: training (1) show_image (1)
4: Run 6: Problems TODO Terminal Python Console
51:1 LF UTF-8 4 spaces Python 3.7 (test_assignment)
Event Log

```python
    acc = history_dict['acc']
    val_acc = history_dict['val_acc']
    loss = history_dict['loss']
    val_loss = history_dict['val_loss']

    plot_history(data_list=[loss, val_loss],
                 label_list=['Training loss', 'Validation loss'],
                 title='Training and validation loss',
                 ylabel='Loss', name = 'base_ncad_loss')

    plot_history(data_list=[acc, val_acc],
                 label_list=['Training accuracy', 'Validation accuracy'],
                 title ='Training and validation accuracy',
                 ylabel ='Accuracy', name = 'base_ncad_acc')

    # model final training
    X_train_ms = np.concatenate((X_train, X_val), axis = 0)
    y_train_ms = np.concatenate((y_train, y_val), axis = 0)

    model, _ = new_model(X_train_ms, y_train_ms,
                         breg = l2(0.001), areg = l1(0.001), final = True)

    evaluate_performance(X_test, y_test, model, name = 'base_roc_ncad')

if __name__ == "__main__":
    training()
```

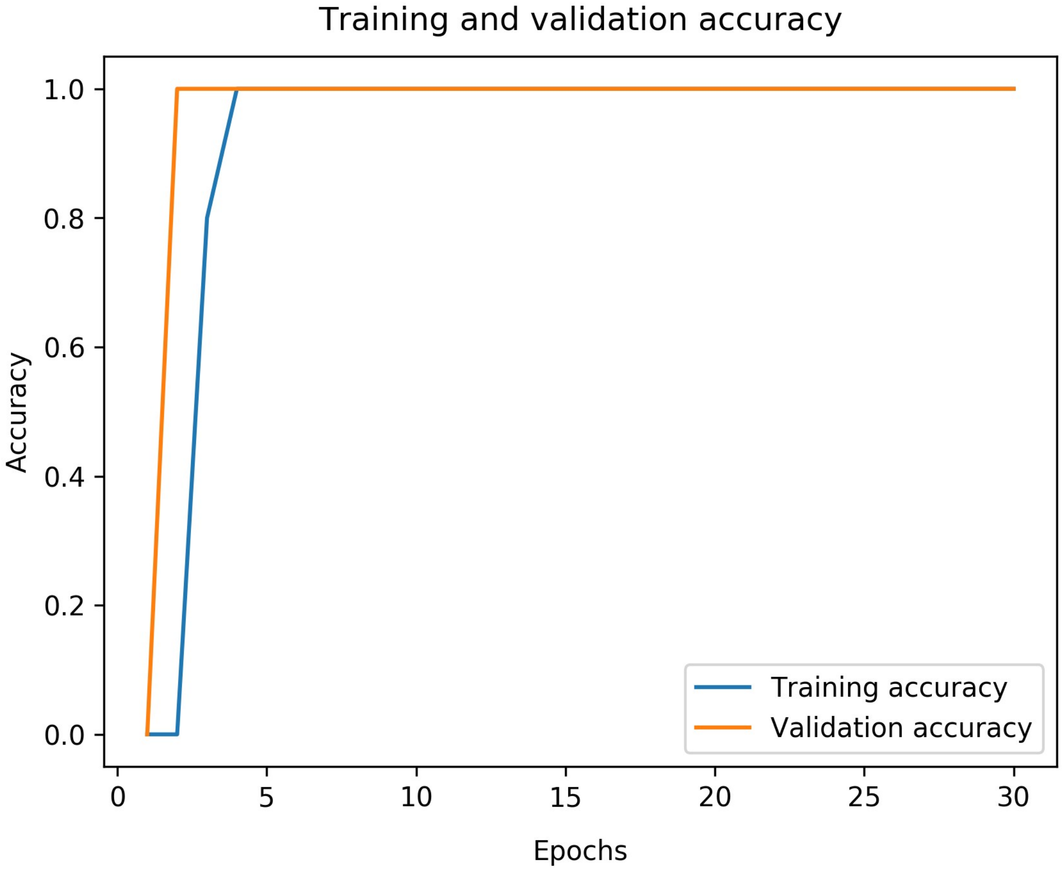
```
Epoch 4/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0048 - acc: 1.0000 - val_loss: 0.0017 - val_acc: 1.0000
Epoch 5/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0024 - acc: 1.0000 - val_loss: 0.0013 - val_acc: 1.0000
Epoch 6/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0015 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 7/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0012 - acc: 1.0000 - val_loss: 0.0011 - val_acc: 1.0000
Epoch 8/30
5/5 [==============================] - 10s 2s/sample - loss: 0.0011 - acc: 1.0000 - val_loss: 0.0010 - val_acc: 1.0000
Epoch 9/30
```

External file changes sync may be slow: The current inotify(7) watch limit is too low. More details. (yesterday 5:28 PM)

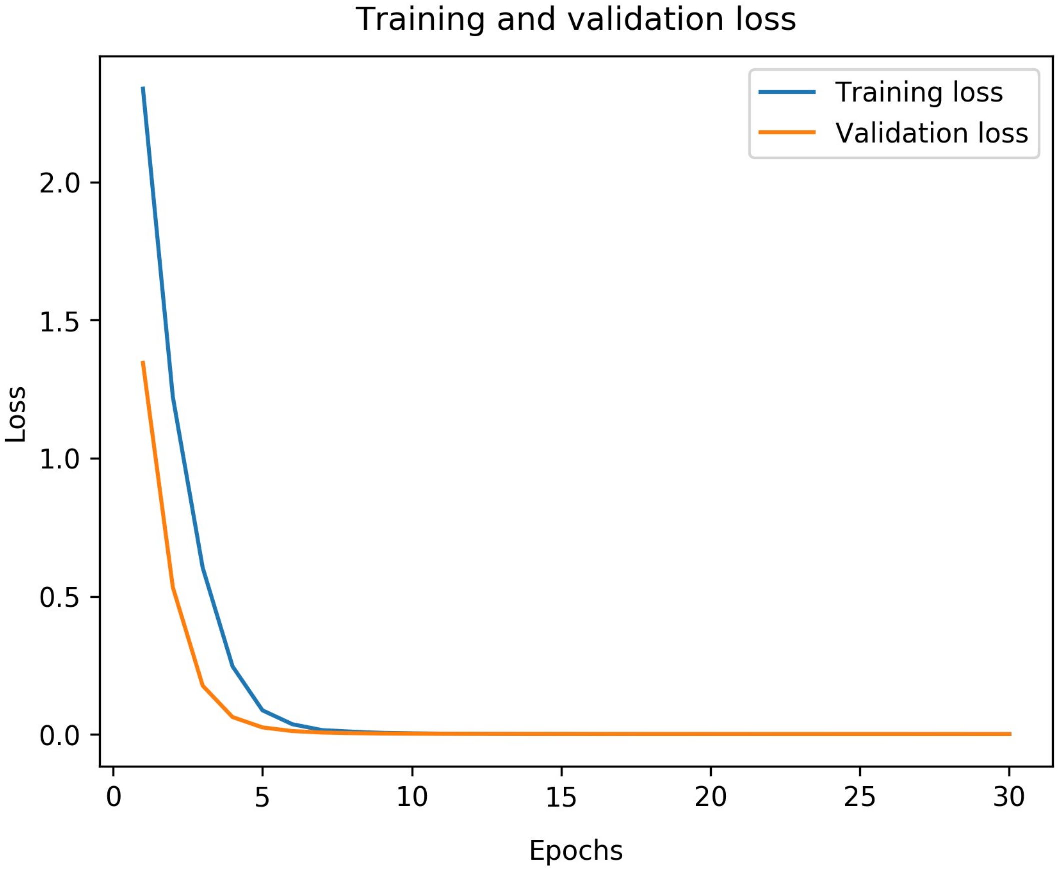
8. Graphs/Accuracy/Configuration Matrix

As we can see that due to limited dataset, the model has been over-fitted.

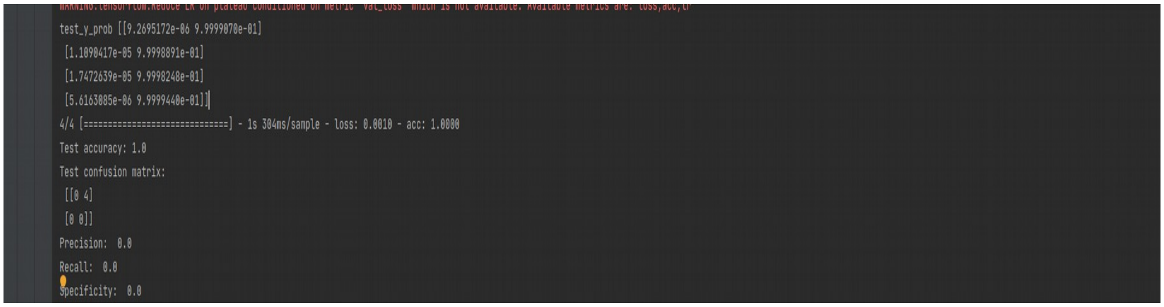
Attaching below graph for Training and validation accuracy.



Attaching below graph for Training and validation loss.



Attaching screenshot for the accuracy, confusion matrix and other parameters. Accuracy has been 1, it shows that the model has been overfitted. We have tried for 30 number of epochs for CNN model. At Epoch 00024: ReduceLROnPlateau reduced learning rate to 4.999999873689376e-06. For more details on ReduceLROnPlateau, you can refer here



```
WARNING:tensorflow:Reduce LR on plateau conditioned on metric `val_loss` which is not available. Available metrics are: loss,acc,lr
test_y_prob [[9.2695172e-06 9.9999070e-01]
 [1.1090417e-05 9.9998891e-01]
 [1.7472639e-05 9.9998248e-01]
 [5.6163085e-06 9.9999440e-01]]
4/4 [==============================] - 1s 304ms/sample - loss: 0.0010 - acc: 1.0000
Test accuracy: 1.0
Test confusion matrix:
 [[0 4]
 [0 0]]
Precision:  0.0
Recall:  0.0
Specificity:  0.0
```

9. Driver script

If we want to Pre-process data again, we have to empty our folders again using Terminal

```
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/health_filtered;
rm *.npy;
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/health_raw;
rm *.nii;cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/patient_filtered;
rm *.npy;
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/patient_raw;
rm *.nii;
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Training/processed;
rm *.npy;
#####Testing
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/health_filtered;
rm *.npy;
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/health_raw;
rm *.nii;
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/Testing/patient_filtered;
rm *.npy;
```

```
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/
Testing/patient_raw;
rm *.nii;
cd /home/divya/Downloads/Cnn_Mri_Classification/6389_project1/6389_project1/
Testing/processed;
rm *.npy;
```

10. References

https://rdrr.io/cran/kerasR/man/ReduceLROnPlateau.html
https://en.wikipedia.org/wiki/Convolutional_neural_network
https://www.vincentkoppelmans.com/neuroscience/quick-visualization-of-nifti-images/
https://www.kaggle.com/kmader/show-3d-nifti-images