Q1)

Import the nifti file and extract the fMRI parameters (such as voxel size, measurement unit, TE, and TR) from the header. Discuss what each parameter represents.

Hint: check the .json file to make sure you extracted the correct parameters from the header file.

In [ ]:
```
# Import the requred Packages
import os
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import nibabel as nib

fmri_file = '../../datasets/fMRI/HW4/sub-001_ses-001_task-eoec_bold.nii.gz' # Get the nifti file
img = nib.load(fmri_file) # Load in the nifti file
print(type(img)) # Print the type of the img variable (should be nifti)
print(img.shape) # Print the dimensions of the object
```

```
<class 'nibabel.nifti1.Nifti1Image'>
(64, 64, 35, 120)
```

From above we can tell that the voxel dimensions are 64x64x35 with 120 different time units.

In [2]:
```
# Extract the voxel size from MRI data.
hdr = img.header # Extract header information
print(hdr) # Print all header information
print('\n---\n') # text separator
print(hdr.get_zooms()) # Print header dimension values
print(hdr.get_xyzt_units()) # Print header dimension units (mm for example)
print(hdr)
```

```
<class 'nibabel.nifti1.Nifti1Header'> object, endian='<'
sizeof_hdr      : 348
data_type       : b''
db_name         : b''
extents         : 0
session_error   : 0
regular         : b'r'
dim_info        : 57
dim             : [  4  64  64  35 120   1   1   1]
intent_p1       : 0.0
intent_p2       : 0.0
intent_p3       : 0.0
intent_code     : none
datatype        : int16
bitpix          : 16
slice_start     : 0
pixdim          : [-1.   4.   4.   4.   2.   0.   0.   0.]
vox_offset      : 0.0
scl_slope       : nan
scl_inter       : nan
slice_end       : 0
slice_code      : unknown
xyzt_units      : 10
cal_max         : 0.0
cal_min         : 0.0
slice_duration  : 0.0
toffset         : 0.0
glmax           : 0
glmin           : 0
descrip         : b'TE=40;Time=185950.000'
aux_file        : b''
qform_code      : scanner
sform_code      : scanner
quatern_b       : 0.0
quatern_c       : 1.0
quatern_d       : 0.0
qoffset_x       : 126.781
qoffset_y       : -94.604996
qoffset_z       : -43.3223
srow_x          : [ -4.       0.      -0.     126.781]
srow_y          : [ -0.          4.         -0.        -94.604996]
srow_z          : [  0.        0.        4.      -43.3223]
intent_name     : b''
magic           : b'n+1'

---
```

```
(4.0, 4.0, 4.0, 2.0)
('mm', 'sec')
<class 'nibabel.nifti1.Nifti1Header'> object, endian='<'
sizeof_hdr      : 348
data_type       : b''
db_name         : b''
extents         : 0
session_error   : 0
regular         : b'r'
dim_info        : 57
dim             : [  4  64  64  35 120   1   1   1]
intent_p1       : 0.0
intent_p2       : 0.0
intent_p3       : 0.0
intent_code     : none
datatype        : int16
bitpix          : 16
slice_start     : 0
pixdim          : [-1.  4.  4.  4.  2.  0.  0.  0.]
vox_offset      : 0.0
scl_slope       : nan
scl_inter       : nan
slice_end       : 0
slice_code      : unknown
xyzt_units      : 10
cal_max         : 0.0
cal_min         : 0.0
slice_duration  : 0.0
toffset         : 0.0
glmax           : 0
glmin           : 0
descrip         : b'TE=40;Time=185950.000'
aux_file        : b''
qform_code      : scanner
sform_code      : scanner
quatern_b       : 0.0
quatern_c       : 1.0
quatern_d       : 0.0
qoffset_x       : 126.781
qoffset_y       : -94.604996
qoffset_z       : -43.3223
srow_x          : [ -4.      0.     -0.    126.781]
srow_y          : [ -0.         4.        -0.        -94.604996]
srow_z          : [  0.       0.       4.     -43.3223]
intent_name     : b''
magic           : b'n+1'
```

The dimensions of each voxel or the voxel size is (4mmx4mmx4mm) or 4mm^3 which means for every voxel in the dimensional space given (in this case being 64x64x35) is 4mm long in each of the 3 dimensions. And the time value is 2 seconds meaning for each value in the time given (in this case being 120) is 2 seconds long which becomes 240 seconds in total.

The Measurement units as previously mentioned where millimeters (mm) for the (x,y,z) component or voxels and seconds (sec) for the time component.

The Echo Time (TE, time between the delivery of the radiofrequency pulse) is 40 milliseconds (ms). We get that number by seeing the TE=40 in which TE is usually given in milliseconds. This is further varified by the json file giving a TE of 0.04 of which is in seconds and to convert seconds into milliseconds, 0.04s * 1000 = 40ms.

The Repetition Time (TR, the time between successive excitation pulses) is 2 seconds and is our time component which was discussed earlier inside the "get_zooms" method.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js