IntroToPython

September 10, 2024

1 Introduction to Medical Image Analysis in Python

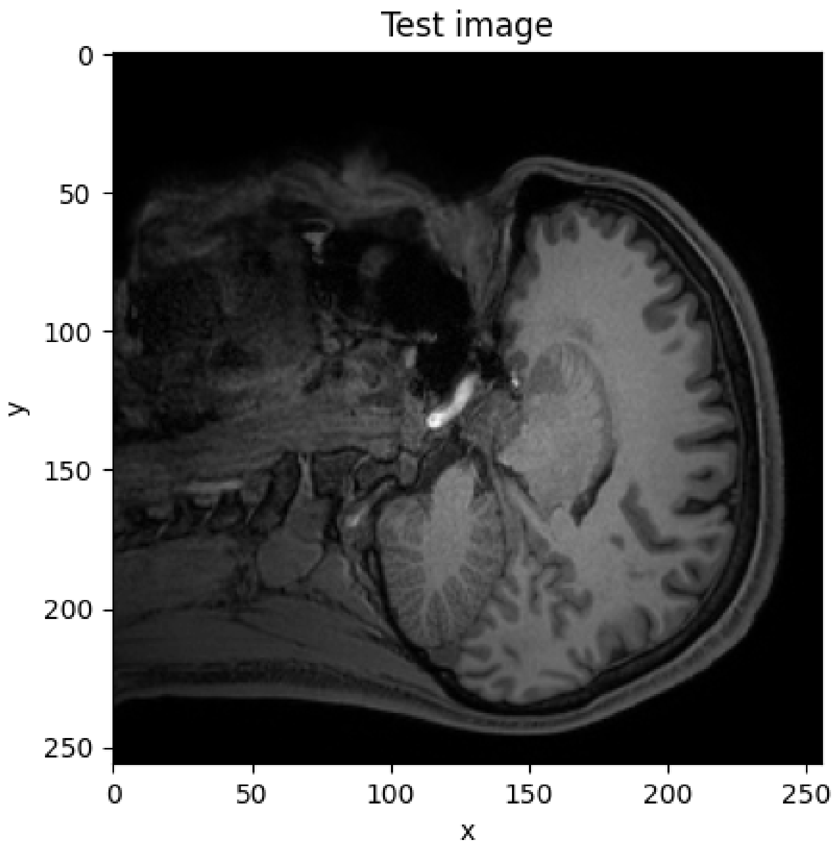
[1]:

```
## Import required modules
# Scientific computations: https://numpy.org/doc/stable/
import numpy as np

# Mathematical algorithm and convenience function: https://docs.scipy.org/doc/
  ↪scipy/
import scipy

# For plots and visualizations: https://matplotlib.org/stable/index.html
import matplotlib.pyplot as plt
%matplotlib inline

# Handling of medical/neuro image data: https://nipy.org/nibabel/gettingstarted.
  ↪html
import nibabel as nib

# Load the Nifti Image
T1 = nib.load('T1.nii')
```

1.1 Task 1: Show one slice of the T1 volume

[2]:

```
# Convert the image data to NumPy array
T1_data = T1.get_fdata()
T1_slice = T1_data[:,:,85]

# Display one slice using matplotlib
plt.figure()
plt.imshow(T1_slice, cmap='gray')
plt.title('Test image')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



1.2 Task 2: Get familiar with the interactive viewer

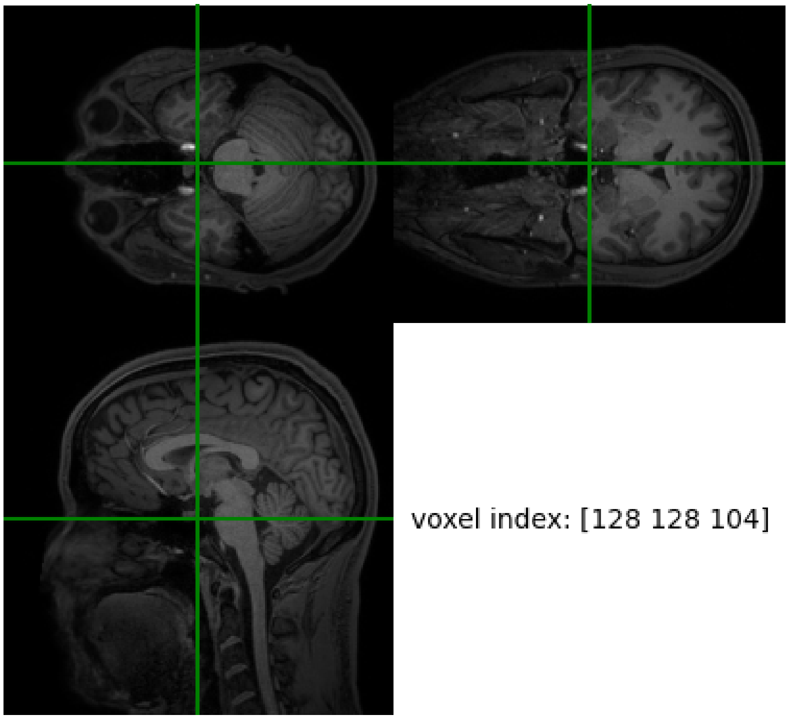
[3]:

```python
class Viewer:
    def __init__(self, data ):
        self.fig, self.ax = plt.subplots()
        self.data = data
        self.dims = self.data.shape
        self.position = np.round( np.array( self.dims ) / 2 ).astype( int )
        self.draw()
        self.fig.canvas.mpl_connect( 'button_press_event', self )
        self.fig.show()

    def __call__(self, event):
        print( 'button pressed' )
        if event.inaxes is None: return

        x, y = round( event.xdata ), round( event.ydata )

        #
```

```
      if ( x > (self.dims[0]-1) ) and ( y <= (self.dims[1]-1) ): return # lower-right quadrant

      #
      if x < self.dims[0]:
        self.position[ 0 ] = x
      else:
        self.position[ 1 ] = x - self.dims[0]

      if y < self.dims[1]:
        self.position[ 1 ] = y
      else:
        self.position[ 2 ] = y -self.dims[1]

      print( f"  voxel index: {self.position}" )
      print( f"  intensity: {self.data[ self.position[0], self.position[1], self.position[2] ]}" )

      self.draw()

  def draw( self ):
      #
      # Layout on screen is like this:
      #
      #     ^            ^
      #  Z  |         Z  |
      #     |            |
      #     ----->        ---->
      #       X             Y
      #     ^
      #  Y  |
      #     |
      #     ----->
      #       X
      #
      dims = self.dims
      position = self.position

      xySlice = self.data[ :, :, position[ 2 ] ]
      xzSlice = self.data[ :, position[ 1 ], : ]
      yzSlice = self.data[ position[ 0 ], :, : ]

      kwargs = dict( vmin=self.data.min(), vmax=self.data.max(),
                     origin='lower',
                     cmap='gray',
                     picker=True )
        self.ax.clear()

        self.ax.imshow( xySlice.T,
                        extent=( 0, dims[0]-1,
                                 0, dims[1]-1 ),
                        **kwargs )
        self.ax.imshow( xzSlice.T,
                        extent=( 0, dims[0]-1,
                                 dims[1], dims[1]+dims[2]-1 ),
                        **kwargs )
        self.ax.imshow( yzSlice.T, extent=( dims[0], dims[0]+dims[1]-1,
                                            dims[1], dims[1]+dims[2]-1 ),
                        **kwargs )

        color = 'g'
        self.ax.plot( (0, dims[0]-1), (position[1], position[1]), color )
        self.ax.plot( (0, dims[0]+dims[1]-1), (dims[1]+position[2],␣
 ↪dims[1]+position[2]), color )
        self.ax.plot( (position[0], position[0]), (0, dims[1]+dims[2]-1), color␣
 ↪)
        self.ax.plot( (dims[0]+position[1], dims[0]+position[1]), (dims[1]+1,␣
 ↪dims[1]+dims[2]-1), color )

        self.ax.set( xlim=(1, dims[0]+dims[1]), ylim=(0, dims[1]+dims[2]) )

        self.ax.text( dims[0] + dims[1]/2, dims[1]/2,
                      f"voxel index: {position}",
                      horizontalalignment='center', verticalalignment='center' )

        self.ax.axis( False )

        self.fig.canvas.draw()
```

[6]:
```
# %matplotlib tk
# %matplotlib notebook
# %matplotlib widget
%matplotlib inline

T1_viewer = Viewer( T1_data )
```

```
/var/folders/v6/q9xwr1gn2dvdngsx947dpj280000gn/T/ipykernel_47546/1359666277.py:9
: UserWarning: FigureCanvasAgg is non-interactive, and thus cannot be shown
  self.fig.show()
```



Text text ...

Text text ...