. MEDICAL IMAGE ANALYSIS.

Nikhil Gupta

Registration. No.: 11709733

Section: KM032 [51]

—

Course: INT247

—

Teacher: Pooja Rana

TOPIC: MEDICAL IMAGE ANALYSIS - DEEP LEARNING
(SOFT COMPUTING IN BIOMEDICAL)

DESCRIPTION: The project is an implementation support on the term paper topic "Soft computing Applications in Biomedical". I choose Medical image analysis as the topic for implementation as came across all the lots that comes under the category of biomedicals such as breast cancer detection, lung cancer detection, prostate cancer, ovaries cancer was already been assigned to the fellow classmates. The teacher asked everyone should work on distinctive projects so keeping that in account and the topic relating Biomedical and soft computing. I decided to go with a project of Medical Image analysis. Though that required me for a lot of self-seeking study for the topic. The prerequisites for working on the project is included in the report in the later sections.

The Project is for Showing the Applications of Soft Computing in Biomedical Sciences for which I've worked for the depiction with two examples.

DATA URL:

I. https://www.dicomlibrary.com/?requestType=WADO&studyUID=1.2.826.0.1.3680043.8.1055.1.20111102150758591.92402465.76095170&manage=1b9baeb16d2

This Dataset of DICOM images (.dcm format) is needed for the image analysis.
Thus, where best to find it from the dicomlibrary.com the database of .dcm images Medical images to be specific. We performed the analysis on the images of brain scanned as from MRI scanners to see and detect brain tumors or factual.

II. http://www.fil.ion.ucl.ac.uk/spm/download/data/MoAEpilot/MoAEpilot.zip

This dataset was taken from the database of Statistical Parametric mapping which has an account of live CT scanned images from the United Kingdom for statistical studies. The data here isn't of .dcm format but a proper medical image from CT scan it has all the attributes a 3D image of brain from which we've retrieved the brain pulse of the persons activity during the scan, just by the images of their brain.

- *IF it doesn't seem to be directly linked to soft computing that's because I haven't applied any regression or clustering as that is very difficult to do on medical images but sure I have made this project in accordance to the topic given to me for the term paper.*

INTRODUCTION:

Analyzing images and videos, and using them in various applications such as self-driven cars, drones etc. with underlying deep learning techniques has been the new research frontier. The recent research papers such as “A Neural Algorithm of Artistic Style”, show how a style can be transferred from an artist and applied to an image, to create a new image. Other papers such as “Generative Adversarial Networks” (GAN) and “Wasserstein GAN” have paved the path to develop models that can learn to create data that is similar to data that we give them. Thus opening up the world to semi-supervised learning and paving the path to a future of unsupervised learning.
While these research areas are still on the generic images, our goal is to use these researches into medical images to help healthcare. We need to start with some basics. In this article, I start with basics of image processing, basics of medical image format data and visualize some medical data. In the next article I will deep dive into some convolutional neural nets and use them with Keras for predicting lung cancer.

Medical Image Data Format

Medical images follow Digital Imaging and Communications (DICOM) as a standard solution for storing and exchanging medical image-data. The first version of this standard was released in 1985. Since then there are several changes made. This standard uses a file format and a communications protocol.

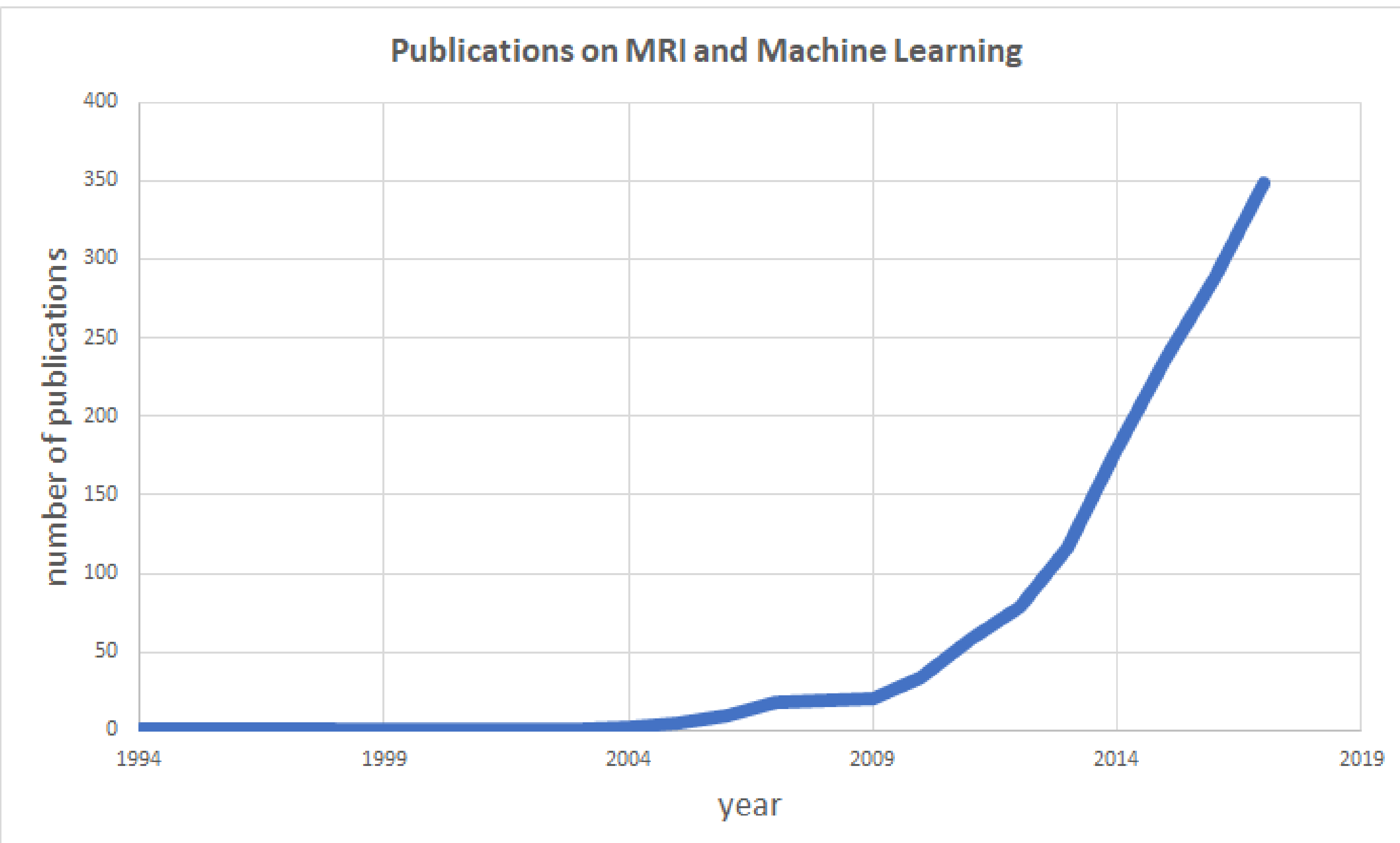
- **File Format** — All patient medical images are saved in the DICOM file format. This format has PHI (protected health information) about the patient such as — name, sex, age in addition to other image related data such as equipment used to capture the image and some context to the medical treatment. Medical Imaging Equipment create DICOM files. Doctors use DICOM Viewers, computer software applications that can display DICOM images, read and to diagnose the findings in the images.

- **Communications Protocol** — The DICOM communication protocol is used to search for imaging studies in the archive and restore imaging studies to the workstation in order to display it. All medical imaging applications that are connected to the hospital network use the DICOM protocol to exchange information, mainly DICOM images but also patient and procedure information. There are also more advanced network commands that are used to control and follow the treatment, schedule procedures, report statuses and share the workload between doctors and imaging devices.

MACHINE LEARNING ON MRI IMAGES

There is a growing interest in applying machine learning techniques on medical data. Brain scans from Magnetic Resonance Imaging experiments (MRI) have been a popular choice with the number of publications combining MRI and machine learning growing exponentially over the last years (see data from PubMed below). Therefore, in this first post we will cover some of the basics about structural and functional MRI (fMRI) data to give you an idea of how the data is generally structured. In the following post we will analyze the data by doing some correlation analysis and by building a general linear model (GLM) to identify active regions in the brain.

The focus of these posts will be on the structure and analysis of the data and not on the underlying principles of magnetic resonance imaging.



The DICOM images of brain pattern detection was taken from the database to study on the patterns and distinction.

Analyze DICOM Images

A very good python package used for analyzing DICOM images is pydicom. First section, we see how to render a DICOM image on a Jupyter notebook.

Download the dicom files and load them on your jupyter notebook.

```
In [14]: #Download dicom images
INPUT_FOLDER = '/Users/taposh/Downloads/dicom_Images/uncompressed/'
patients = os.listdir(INPUT_FOLDER)
patients.sort()
```

Now, load the DICOM images into a list.

```python
In [10]: #Collect all dicom images
         lstFilesDCM = []  # create an empty list
         def load_scan2(path):
             for dirName, subdirList, fileList in os.walk(path):
                 for filename in fileList:
                     if ".dcm" in filename.lower():  # check whether the file's DICO
                         lstFilesDCM.append(os.path.join(dirName,filename))
                         #print(lstFilesDCM)
             return lstFilesDCM

         first_patient = load_scan2(INPUT_FOLDER)
         #print(first_patient)
```

Step 1 : Basic View of DICOM Image in Jupyter

```python
In [15]: # Get ref file
         RefDs = pdicom.read_file(lstFilesDCM[0])

         # Load dimensions based on the number of rows, columns, and slices (along the Z axis)
         ConstPixelDims = (int(RefDs.Rows), int(RefDs.Columns), len(lstFilesDCM))

         # Load spacing values (in mm)
         ConstPixelSpacing = (float(RefDs.PixelSpacing[0]), float(RefDs.PixelSpacing[1]),
                              float(RefDs.SliceThickness))
```

In the first line we load the 1st DICOM file, which we're going to use as a reference named `RefDs`, to extract metadata and whose filename is first in the `lstFilesDCM` list.

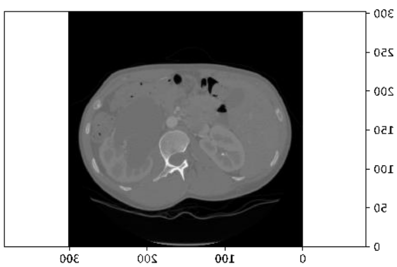
```python
In [7]: x = np.arange(0.0, (ConstPixelDims[0]+1)*ConstPixelSpacing[0], ConstPixelSpacing[0])
        y = np.arange(0.0, (ConstPixelDims[1]+1)*ConstPixelSpacing[1], ConstPixelSpacing[1])
        z = np.arange(0.0, (ConstPixelDims[2]+1)*ConstPixelSpacing[2], ConstPixelSpacing[2])
```

We then calculate the total dimensions of the 3D NumPy array which are equal to (Number of pixel rows in a slice) x (Number of pixel columns in a slice) x (Number of slices) along the x, y, and z cartesian axes. Lastly, we use the `PixelSpacing` and `SliceThickness` attributes to calculate the spacing between pixels in the three axes. We store the array dimensions in `ConstPixelDims` and the spacing in `ConstPixelSpacing [1]`.

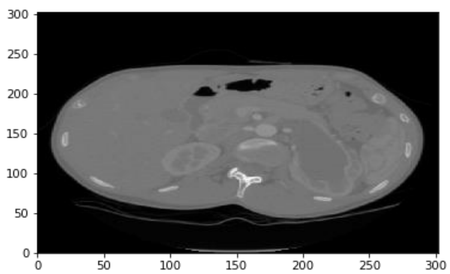
```python
In [8]: # The array is sized based on 'ConstPixelDims'

        ArrayDicom = np.zeros(ConstPixelDims, dtype=RefDs.pixel_array.dtype)

        # loop through all the DICOM files
        for filenameDCM in lstFilesDCM:
            # read the file
            ds = pdicom.read_file(filenameDCM)
            # store the raw image data
            ArrayDicom[:, :, lstFilesDCM.index(filenameDCM)] = ds.pixel_array
```

After this we plot the Image and analyse it in different forms.

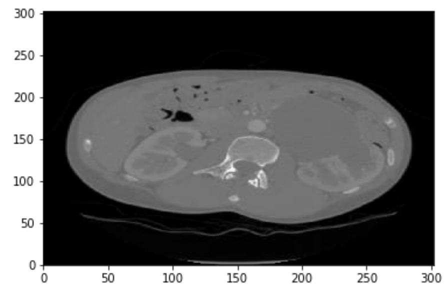
```
# finally plotting the dicom image with Quadmesh for 3D visual.
plt.figure(dpi=1600)
plt.axes().set_aspect('equal','datalim')
plt.set_cmap(plt.gray())
```



plt.pcolormesh(x, y, np.flipud(dicomArray[:,:,120]))



plt.pcolormesh(x, y, np.flipud(dicomArray[:,:,80]))



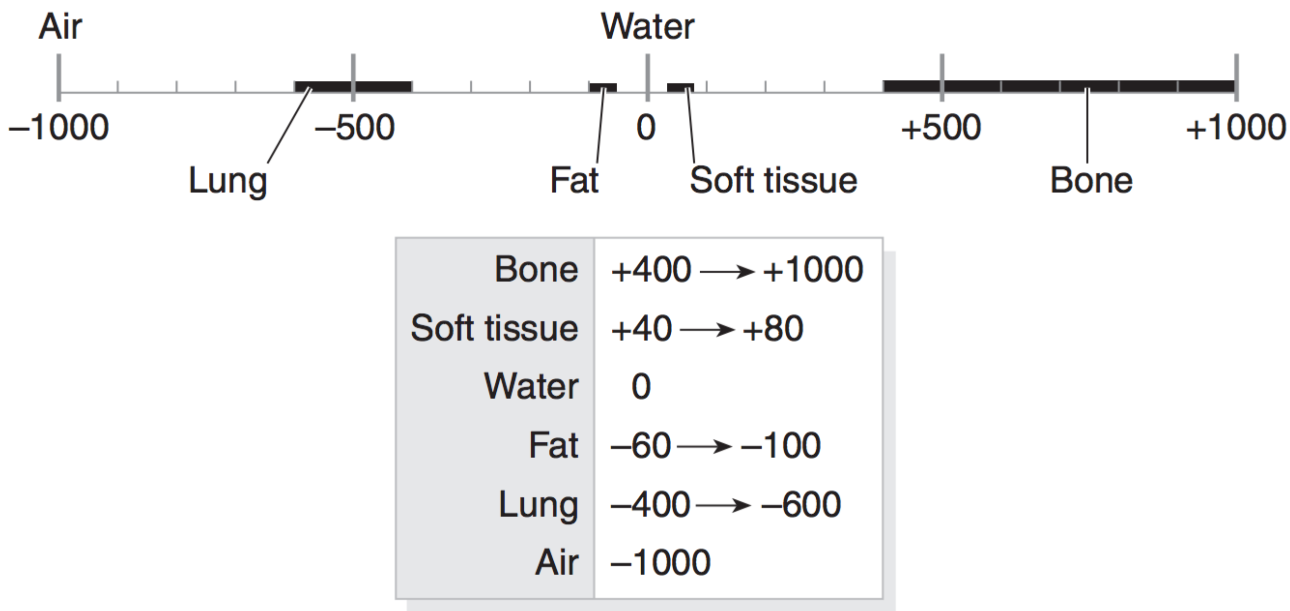
plt.pcolormesh(x, y, np.flipud(dicomArray[:,:,160]))

Step 2: Looking into details of DICOM format

The unit of measurement in CT scans is the **Hounsfield Unit (HU)**, which is a measure of radiodensity. CT scanners are carefully calibrated to accurately measure this. A detailed understanding on this can be found here.

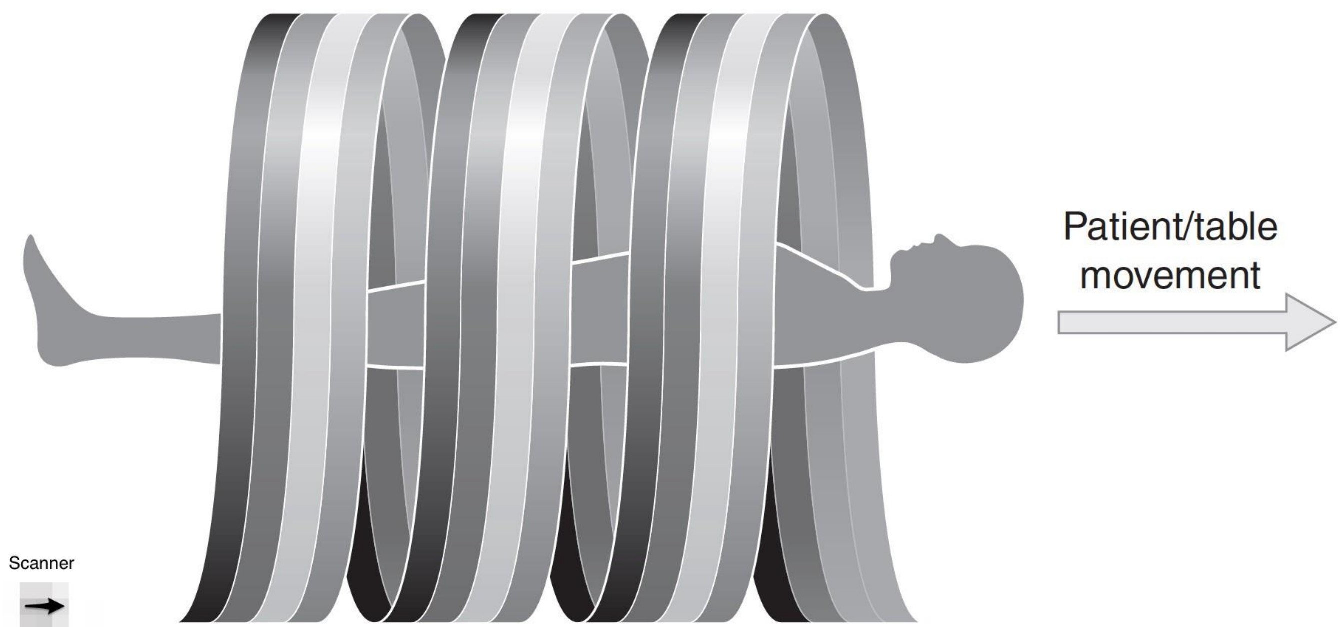
Each pixel is assigned a numerical value (CT number), which is the average of all the attenuation values contained within the corresponding voxel. This number is compared to the attenuation value of water and displayed on a scale of arbitrary units named Hounsfield units (HU) after Sir Godfrey Hounsfield.

This scale assigns water as an attenuation value (HU) of zero. The range of CT numbers is **2000 HU** wide although some modern scanners have a greater range of HU up to 4000. Each number represents a shade of grey with +1000 (white) and –1000 (black) at either end of the spectrum.



Hounsfield Scale [credits: [“Introduction to CT physics”](#) (PDF). elsevierhealth.com.]

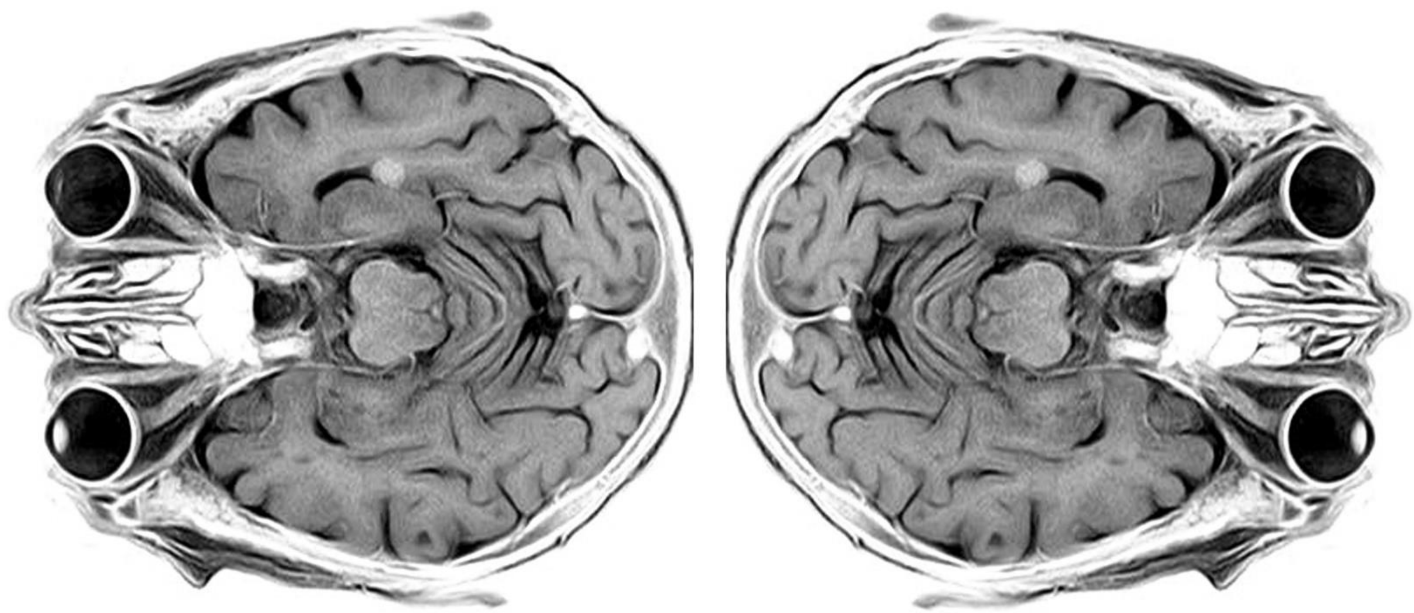
Some scanners have cylindrical scanning bounds, but the output image is square. The pixels that fall outside of these bounds get the fixed value **-2000**.



CT Scanner Image [credits : [“Introduction to CT physics”](#) (PDF). elsevierhealth.com.]

Structural MRI images

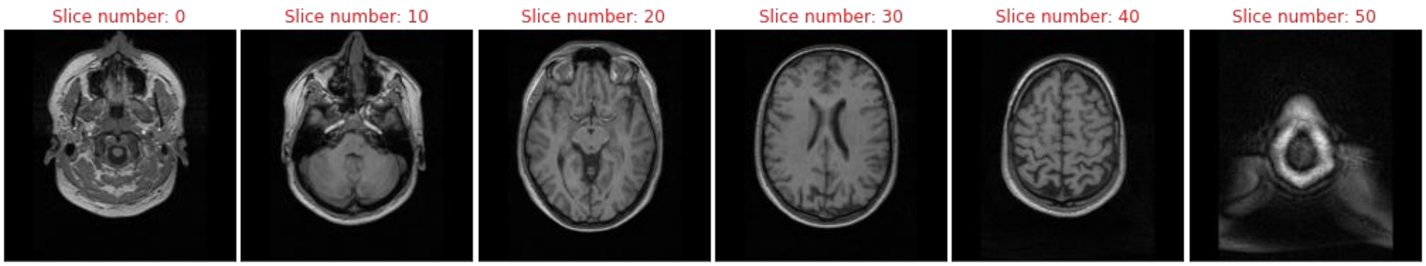
Structural MRI scans usually visualize the location of water in the human body. This means that soft tissues with high water and fat concentration such as the brain can be well resolved while more dense structures such as bones have a lower signal amplitude. Structural MRI scans allow clinicians to visualize and locate anatomical structures within the brain in great detail. This is why fMRI experiments which try to identify active regions in the brain during specific tasks are typically combined with structural MRI scans. Although structural MRI images are often shown as 2-D images they actually represent volume information. That is why the elements in each image are referred to as volumetric pixels, or voxels, instead of pixels as in standard 2-D images. Typically the brain is scanned in several planes or slices during a MRI session which underlines the volumetric nature of the method. We will see later when we come to functional MRI scans (fMRI) what this means in practice.



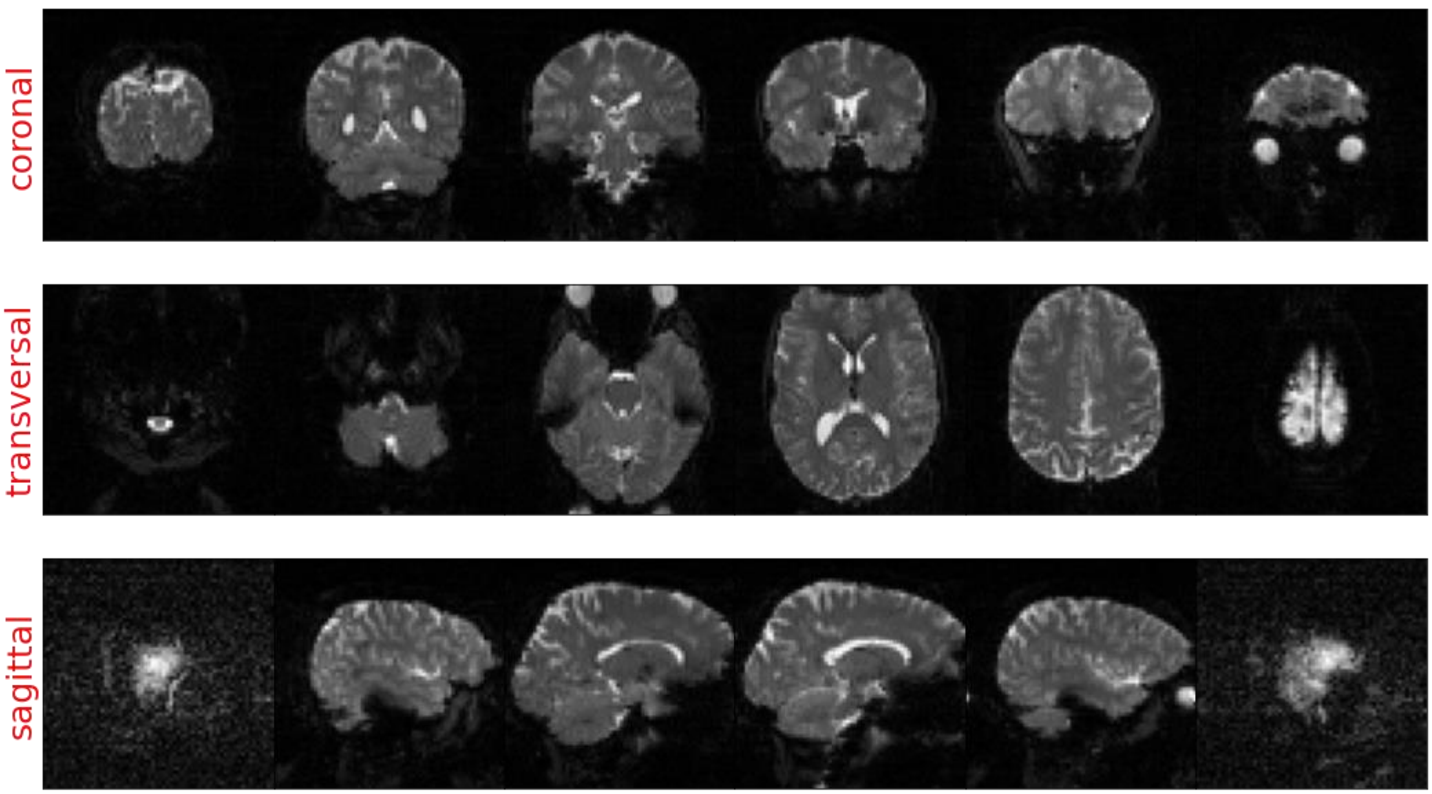
The data we are going to use here is from a human subject laying in a MRI machine while listening to “ bi-syllabic words” as the description reads on the SPM homepage. This auditory stimulation will later allow us to see which areas in the brain are involved in perceiving these words. But first we will have a look at the structural MRI scan.

Visualizing structural MRI data

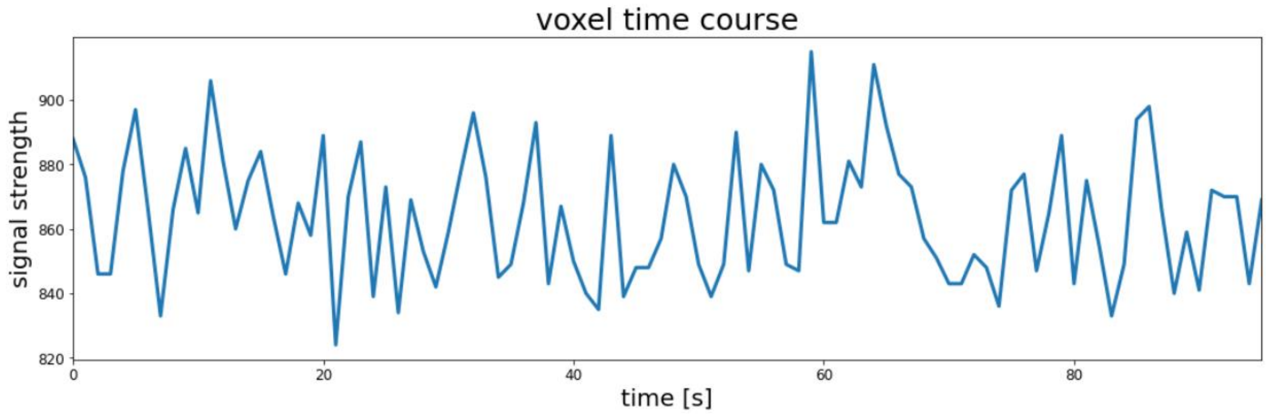
After loading in the data with the help of the NiBabel library we can see that the data actually has 4-Dimensions. The first two are the X- and Y-planes while the 3rd dimension represents the number of slice in the scan. The 4th dimension does not contain any information and can be discarded.



Visualizing the Data Image of functional MRI in 3Dimensions on a 2D plane we need to see it in 3 different views as follows.



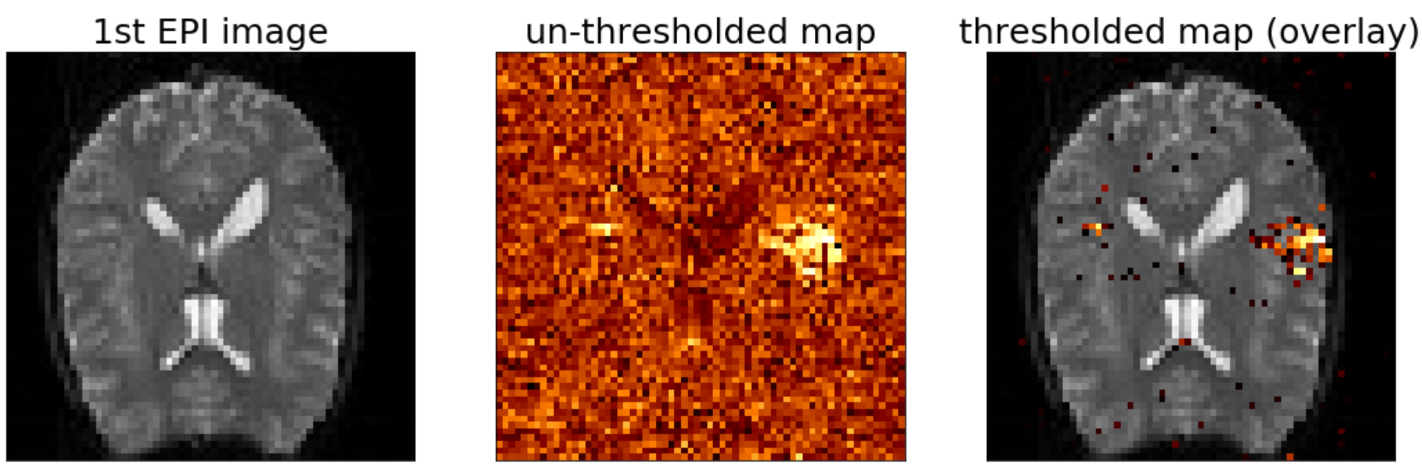
In the Last we see the Brain mapping pattern of the strengths in the traversal view.



RESULT and CONCLUSION:

Results shows how with the help of deep learning modules we can read and get insights from the MRI scanned images to detect patterns and predict brain signal strength,
These are just few examples in support of how Soft Computing is helping the Biomedical Industry.

Below is another example of detection of auditory signals which is an extended insight from the coxel signal strength we plotted. This below can be further visualized by plotting with cmap=gray cmap=amfhot and cmap=gray,smap=amfhot combined.



REFERENCES:

https://pyscience.wordpress.com/2014/09/08/dicom-in-python-importing-medical-image-data-into-numpy-with-pydicom-and-vtk/

https://nifti.nimh.nih.gov/

http://www.fil.ion.ucl.ac.uk/spm/

https://en.wikipedia.org/wiki/Anatomical_plane